Azure Database Services

- Azure has a number of fully managed databases.
- They areSQL , COSMOS DB MySQL, PostgreSQL, MariaDB, and Managed Instance databases, each with specific workloads in mind.

1. Azure SQL Database (Relational Database Service)

What is Azure SQL Database?

Azure SQL Database is a fully managed relational database service built on Microsoft SQL Server. It offers features such as high availability, automatic backups, and seamless scaling, and does not require users to manage the underlying infrastructure.

Key Features
✓ Fully Managed – Microsoft handles patching, backups, and maintenance.
✓ Built-in AI Optimization – Performance tuning using AI-based indexing.
✓ High Availability – Geo-replication and automated failover.
✓ Security – Transparent data encryption (TDE) and threat detection.

Practical Example: Creating an Azure SQL Database

Step 1: Create an Azure SQL Database

```
az sql server create \
    --name my-sql-server \
    --resource-group MyResourceGroup \
    --location eastus \
    --admin-user adminuser \
    --admin-password MySecurePass123!

az sql db create \
    --resource-group MyResourceGroup \
    --server my-sql-server \
    --name mydatabase \
    --edition GeneralPurpose \
--compute-model Serverless
```

Step 2: Create a Firewall Rule (To Permitted Outgoing Internet Access)

```
az sql server firewall-rule create \
    --resource-group MyResourceGroup \
    --server my-sql-server \
    --name AllowMyIP \
    --start-ip-address <YOUR_IP> \
    --end-ip-address <YOUR_IP>
```

Step 3: Use Azure Data Studio or SSMS to connect to SQL database

```
SELECT name, database_id, create_date FROM sys.databases;
```

✅Use Cases: Store transactional data, analytics, business applications

2. Azure Cosmos DB (NoSQL Database Service)

What is Azure Cosmos DB?

Azure Cosmos DB is a globally distributed, multiple-model NoSQL database service designed for applications that are very low-latency performance.

Key Features

- Multi-Model Support: Key-Value, Document, Graph, and Column-Family data models.
- Global Distribution: Data Replication across Multiple Regions for Low Latency.
- Multi-API Compatibility – Supports MongoDB, Cassandra, Gremlin, and Table Storage.
- Automatic Scaling – Dynamically scales performance according to demand.

Practical Example: Creating an Azure Cosmos DB Instance

Step 1: Create a Cosmos DB Account

```
az cosmosdb create \
    --name mycosmosdb \
    --resource-group MyResourceGroup \
    --kind MongoDB
```

Step 2: Create a Database and Collection

```
az cosmosdb sql database create\
--account-name mycosmosdb \
    --resource-group MyResourceGroup \
    --name mydatabase

az cosmosdb sql container create \
    --account-name mycosmosdb \
    --resource-group MyResourceGroup \
    --database-name mydatabase \
    --name mycollection \
    --partition-key-path \"/id
```

Step 3: Insert a JSON Document

```
az cosmosdb sql container throughput update \
    --account-name mycosmosdb \
--resource-group MyResourceGroup \
    --database-name mydatabase \
    --name mycollection \
    --throughput 400
```

Step 4: Query Data

```
SELECT * FROM mycollection WHERE mycollection.id = '123'
```

✅ Use Case: Real-time applications, IoT telemetry, e-commerce platforms, and globally distributed apps.

3.Azure Database for MySQL (Relational Database Service)

What is Azure Database for MySQL?

- A fully managed MySQL service with built-in high availability, scaling, and security.
- Supports automatic backups, scaling compute & storage, and performance tuning.

Practical Example: Deploying an Azure MySQL Database

Step 1: Create an Azure MySQL Server

```
az mysql flexible-server create \
    --resource-group MyResourceGroup \
    --name mymysqlserver \
    --location eastus \
    --admin-user myadmin \
    --admin-password MySecurePass123! \
    --sku-name Standard_D2s_v3
```

Step 2: Configure a Firewall Rule (To Allow Connections)

```
az mysql flexible-server firewall-rule create \
    --resource-group MyResourceGroup \
    --name mymysqlserver \
    --rule-name AllowMyIP \
    --start-ip-address <YOUR_IP> \
    --end-ip-address <YOUR_IP>
```

Step 3: Connect to MySQL from CLI

```
mysql -h mymysqlserver.mysql.database.azure.com -u myadmin -p
```

✅ Use Case: Web applications, e-commerce platforms, and SaaS applications.

4. PostgreSQL Azure Database (Relational Database Service)

What is PostgreSQL Azure Database?

- Open-source PostgreSQL with scalability and security features is supported by this fully managed PostgreSQL service.
- Hyperscale (Citis) for a distributed scale and a flexible server for optimal cost and performance support.

Implementing an Azure PostgreSQL Database as a Real-World Example

Step 1: Set up a PostgreSQL server in Azure

```
az postgres flexible-server create \
 --sku-name Standard_B2ms MyResourceGroup \
--name mypostgresqlserver \
--location eastus \
--admin-user myadmin \
--admin-password MySecurePass123!
```

Step 2: Configure a firewall

```
az postgres flexible-server firewall-rule create \
   --resource-group MyResourceGroup \
   --name mypostgresqlserver \
   --rule-name AllowMyIP \
   --start-ip-address <YOUR_IP> \
   --end-ip-address <YOUR_IP>
```

Step 3: Use the CLI

```
psql "host=mypostgresqlserver.postgres.database.azure.com
user=myadmin password=MySecurePass123! dbname=postgres"
```

✅ Use Case: Workloads related to analytics, financial systems, and geographic applications.

5. Azure Database for MariaDB (Relational Database Service)

What is Azure Database for MariaDB?

A managed MariaDB service built for high availability and scaling.
Suitable for web applications that require MariaDB compatibility
.
Practical Example: Creating an Azure MariaDB Database

Step 1: Create an Azure MariaDB Server

```
az mariadb server create \
    --resource-group MyResourceGroup \
    --name mymariadbserver
--location eastus \
    --admin-user myadmin \
    --admin-password MySecurePass123! \
    --sku-name GP_Gen5_2
```

Step 2: Configure a Firewall Rule

```
az mariadb server firewall-rule create \
    --resource-group MyResourceGroup \
    --server mymariadbserver \
    --name AllowMyIP \
    --start-ip-address <YOUR_IP> \
    --end-ip-address <YOUR_IP>
```

Step 3: Connect to MariaDB from CLI

```
mysql -h mymariadbserver.mariadb.database.azure.com -u myadmin -p
```

✅ use case: An efficient web hosting option for WordPress and Joomla.

6. Azure Managed Instance for SQL Server (Hybrid & Enterprise-Grade SQL)

What is Azure Managed Instance?

A hybrid SQL database solution that supports the migration of on-prem SQL Server workloads to Azure without change. Supports SQL Server features like SQL Agent, Cross-Database Queries, and Linked Servers.

Deploying an Azure Managed Instance is a real-world example.

Step 1: Establish a Managed Azure SQL Instance

```
az sql mi create \
    --name mymanagedinstance \
    --resource-group MyResourceGroup \
    --location eastus \
    --admin-user myadmin \
    --admin-password MySecurePass123! \
    --vnet-name MyVNet \
    --subnet MySubnet
```

Step 2: Connecting to the Managed Instance with SSMS

```
sqlcmd -S mymanagedinstance.public.abcdefg.database.windows.net -U myadmin -P MySecurePass123!
```

✅ Use Case: Enterprise databases, hybrid cloud solutions, and legacy SQL migrations.