

Designing and Implementing an Azure Data Solution

DP 203



Relational Data Stores



Agenda

01

Introduction to Relational
Data Stores

03

Azure SQL Security Capabilities

05

Azure Database for MySQL

07

What is PolyBase?

02

Azure SQL Database

04

High Availability and
Azure SQL Database

06

Azure Database
for MariaDB

08

What is Azure
Synapse Analytics ?

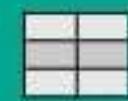
Introduction to Relational Data Stores

Relational Data Stores

Each table has its own columns, and every row in a table has the same set of columns



A relational database is a type of database that stores and provides access to data points that are related to one another

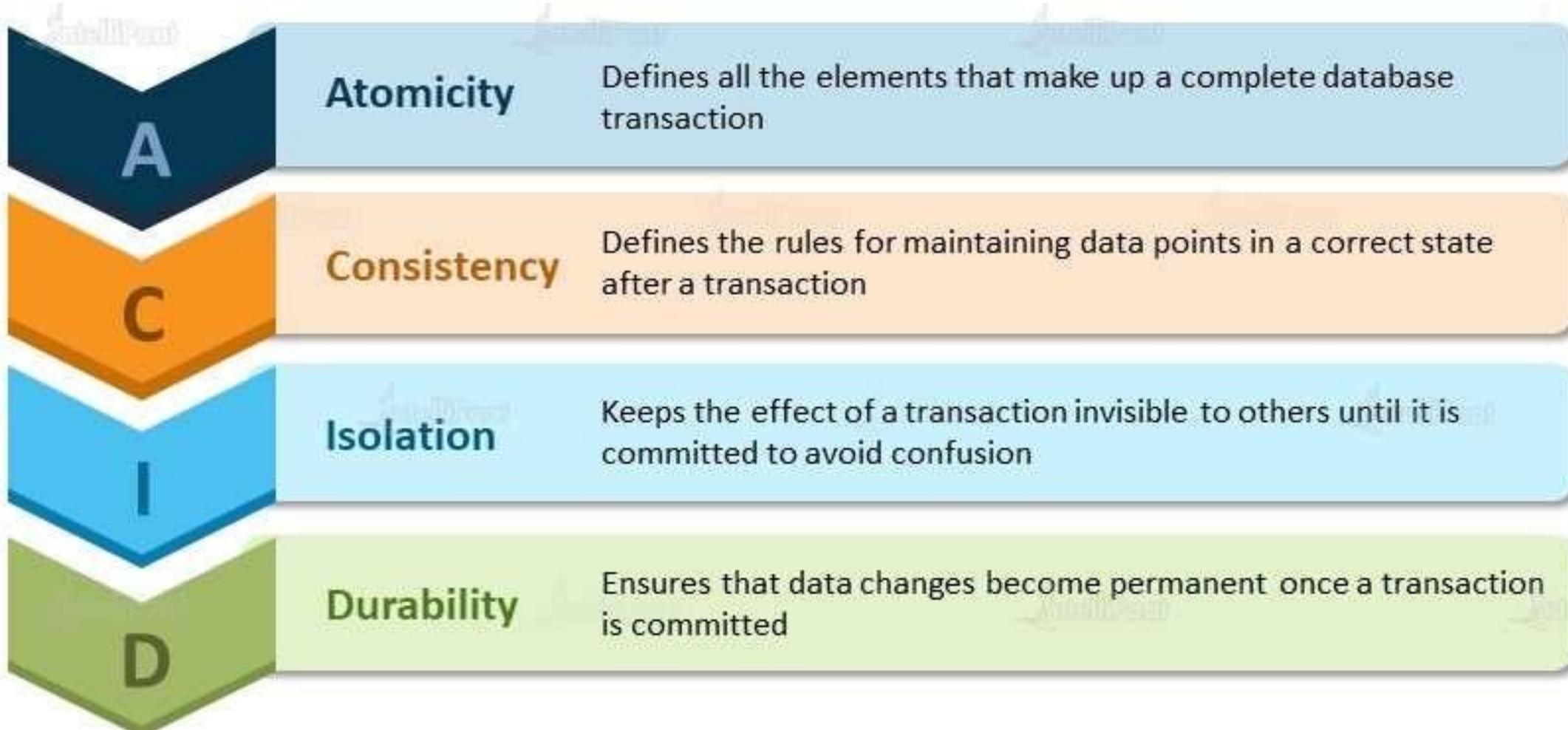


A relational database organizes data as a series of two-dimensional tables with rows and columns



These are the four crucial properties that define relational database transactions:





Relational Data Store

Relevant Azure Services

Azure
SQL Database



Azure Database
for MySQL



Azure Database
for PostgreSQL



Azure Database
for MariaDB



Azure SQL Database

Azure SQL Database

Azure SQL Database is a general-purpose relational database, provided as a managed service



It is the right choice for a variety of modern cloud applications as it enables us to process both relational data and non-relational structures

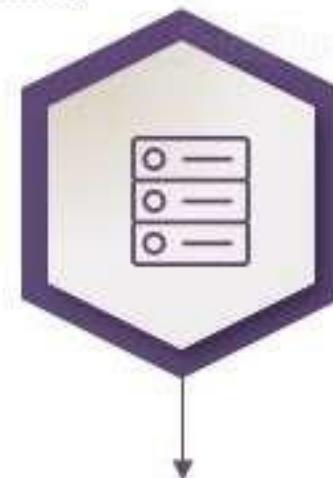


With it, we can create a high-available and high-performance data storage layer for the applications and solutions in Azure



Azure SQL Database

We can use advanced query processing features,
such as **high-performance in-memory
technologies** and **intelligent query processing**



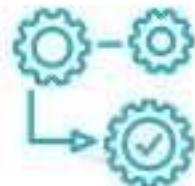
It is based on the latest stable
version of the **Microsoft SQL
Server database engine**

In fact, the newest capabilities of SQL
Server are released first to the SQL
Database and then to SQL Server



01

SQL Database enables us to easily define and scale performance within two different purchasing models: a **vCore-based purchasing model** and a **DTU-based purchasing model**

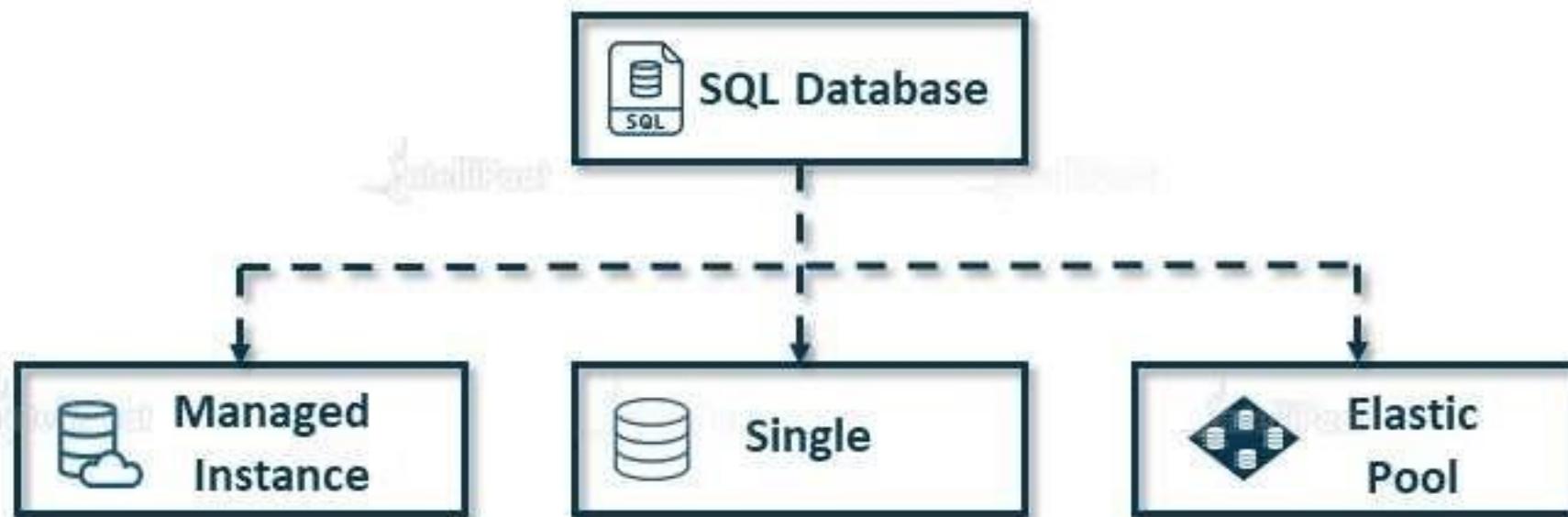


SQL Database is a fully managed service that has built-in high availability, backups, and other common maintenance operations

02

Deployment Models

Azure SQL Database provides the following deployment options



Deployment Models



Single Database



Managed Instance



Elastic Pool

Single database
represents a
fully managed,
isolated
database

A single
database is
similar to a
contained
database in
Microsoft SQL
Server Database
Engine

We use this if we
have modern
cloud
applications and
microservices
that need a
single reliable
data source

Deployment Models



Single Database



Managed Instance



Elastic Pool

Managed instance is a fully managed instance of the Microsoft SQL Server Database Engine

It contains a set of databases that can be used together

We use this option for easy migration of on-premises SQL Server databases to the Azure cloud

Deployment Models



Elastic pool is a collection of single databases with a shared set of resources, such as CPU or memory

Single databases can be moved into and out of an elastic pool

Azure SQL Database

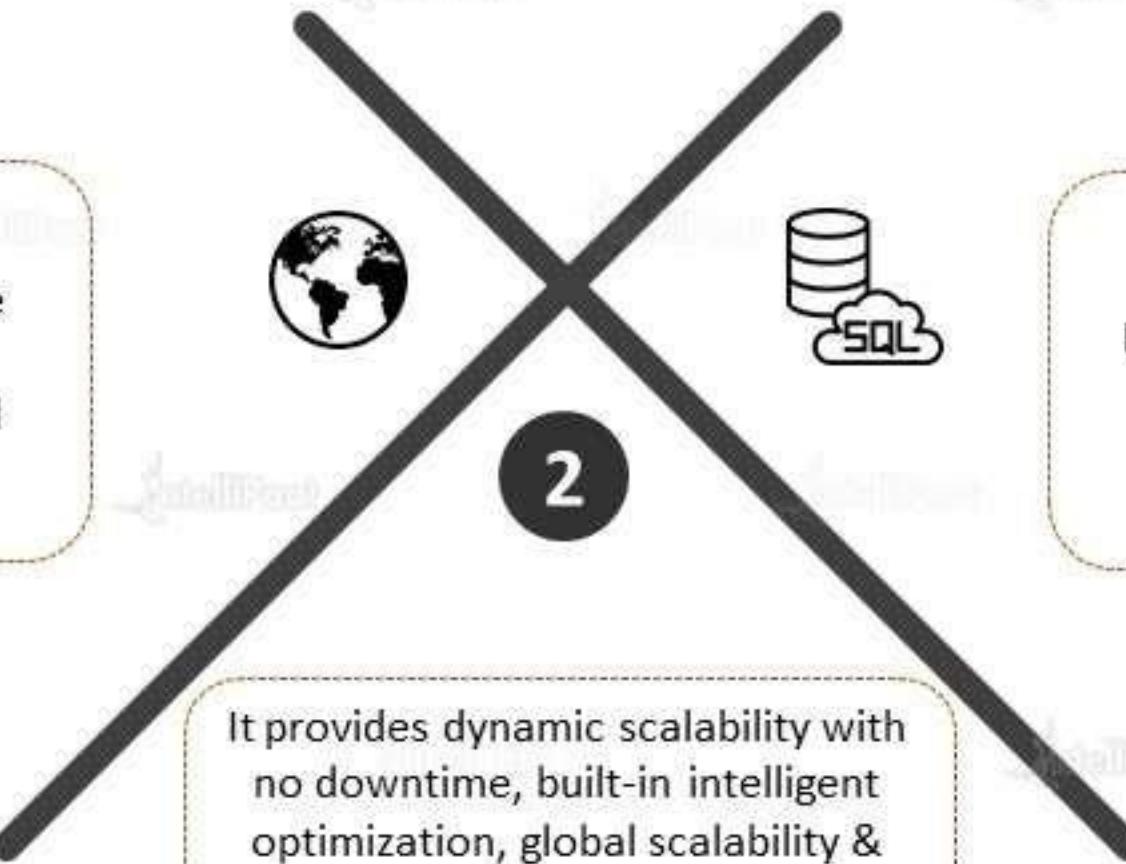
3

The SQL Database service is currently in 38 data centers around the world

1

It delivers predictable performance with multiple resource types, service tiers, and compute sizes

2



It provides dynamic scalability with no downtime, built-in intelligent optimization, global scalability & availability, and advanced security options

Service Tiers

Azure SQL Database offers three service tiers designed for different types of applications:

Hyperscale service tier designed for very large OLTP databases to leverage the ability to auto-scale storage and compute fluidly



General-purpose/Standard service tier is designed for common workloads. It offers budget-oriented balanced compute and storage options



Business-critical/Premium service tier is designed for OLTP applications with a high transaction rate and the lowest latency I/O. It offers the highest resilience to failures by using several isolated replicas

Hands-on: Creating a Single Database Using Azure Portal

Hands-on: Creating a Managed Instance

Hands-on: Creating an Elastic Pool

Why SQL Database Elastic Pool?

Why SQL Database Elastic Pool?

Pools are well suited for a large number of databases with specific utilization patterns



For a given database, this pattern is characterized by a low average utilization with relatively infrequent utilization spikes

The more databases we add to a pool the greater our savings become



Hands-on: Creating a SQL Virtual Machine

Hands-on: Configuring Active Geo-replication for Azure SQL Database in Azure Portal & Initiating a Failover

Why Azure SQL?

Why Azure SQL?

Frictionless Migration



- ★ Accelerates the on-premises SQL Server migrations without changing the application code with the Managed Instance
- ★ Always has the latest SQL Server capabilities in the cloud with an evergreen SQL Database that requires no patching or upgrading from our end

Why Azure SQL?

Built-in Machine Learning



- ★ Gets peak database performance and durability with safe, reliable, and proven AI technology
- ★ Maximizes the performance of our application with customized auto-tuning recommendations
- ★ Uses Intelligent Insights to monitor and detect disruptive events that can cause poor performance

Why Azure SQL?

Unmatched Scaling and High Availability



- ★ Scales our application on demand with up to 99.995 percent availability
- ★ Scales compute and storage resources independently for maximum flexibility and lowers the costs with discounted readable replicas
- ★ Its built-in high availability guarantees that the database will never be a single point of failure in our software architecture

Why Azure SQL?

Advanced Data Security

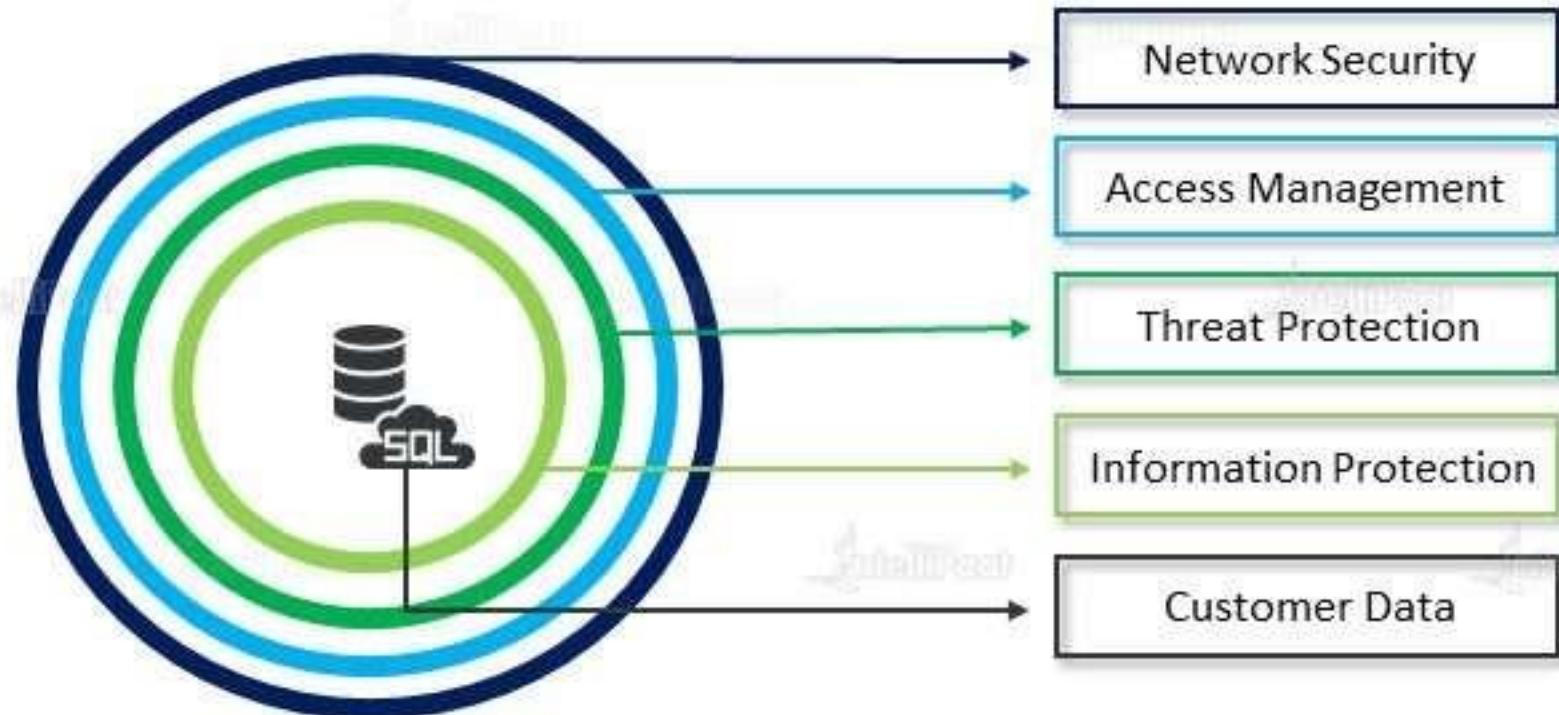


- ★ Protects our databases from malicious acts with fine-grained access controls, Always Encrypted technology, and advanced threat protection capabilities
- ★ Discovers, tracks, and remediates potential vulnerabilities from a single screen
- ★ Meets the most stringent compliance standards with built-in auditing and information protection technologies

Azure SQL Security Capabilities

Azure SQL Security Capabilities

The security strategy follows a layered defense-in-depth (DiD) approach that moves outside in, as shown in the picture below:



Azure SQL Security Capabilities

Network Security

Access Management

Threat Protection

Information Protection

Microsoft Azure SQL Database provides a relational database service for cloud and enterprise applications



To help protect customer data, firewalls prevent network access to the database server until the access is explicitly granted based on the IP address or Azure VNet traffic origin

Azure SQL Security Capabilities

Network Security

Access Management

Threat Protection

Information Protection

IP firewall rules grant access to databases based on the originating IP address of each request



Virtual network service endpoints extend our VNet connectivity over the Azure backbone

Virtual network service endpoints enable Azure SQL Database to identify the virtual network subnet that the traffic originates from

Azure SQL Security Capabilities

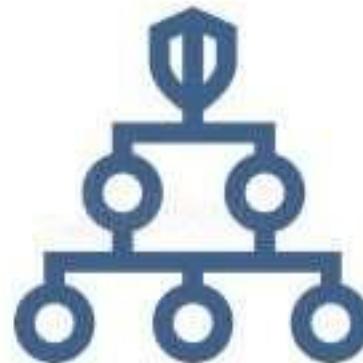
Network Security

Access Management

Threat Protection

Information Protection

To allow traffic to reach Azure SQL Database, we use the SQL service tags that allow the outbound traffic through Network Security Groups



Virtual network rules enable Azure SQL Database to accept communications that are sent only from select subnets inside a virtual network

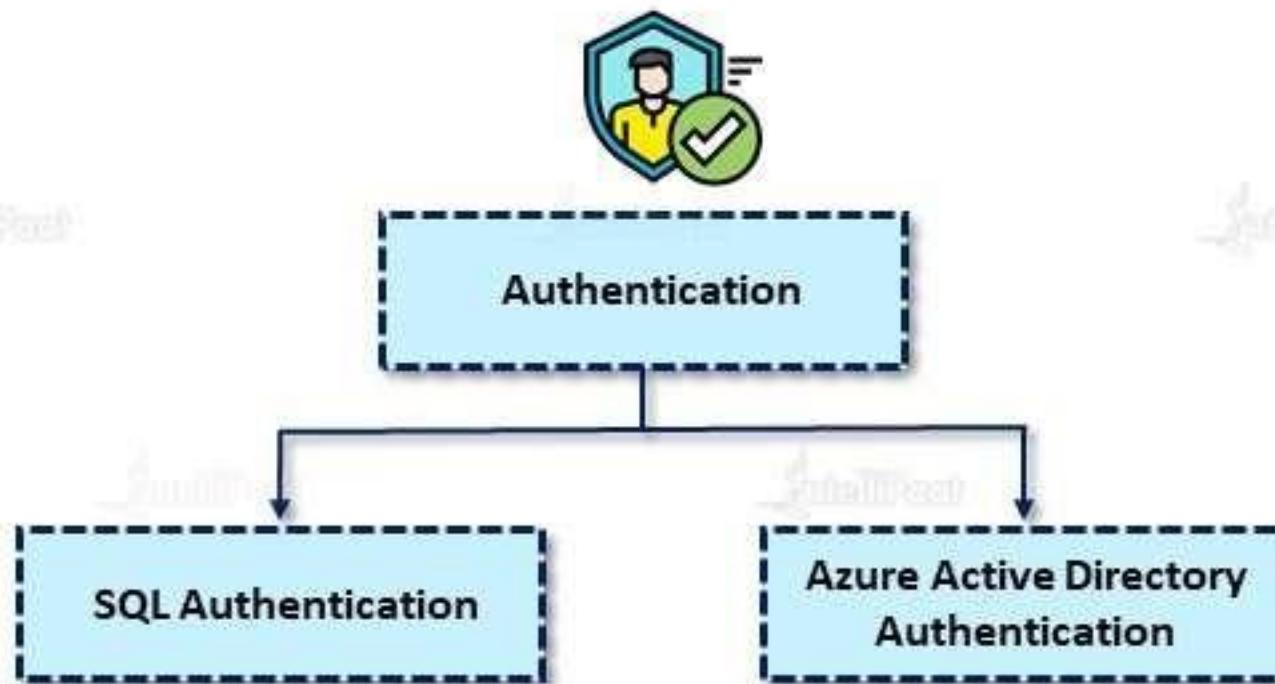
Network Security

Access Management

Threat Protection

Information Protection

- ★ Managing databases and database servers within Azure is controlled by the portal's role assignments of user accounts
- ★ **Authentication** is the process of proving that a user is who he/she claims to be



Azure SQL Security Capabilities

Network Security

Access Management

Threat Protection

Information Protection



SQL Authentication

SQL database authentication refers to the authentication of a user when connecting to Azure SQL Database using username and password

Azure SQL Security Capabilities

Network Security

Access Management

Threat Protection

Information Protection



Azure Active Directory
Authentication



- ★ Azure Active Directory authentication is a mechanism of connecting to Azure SQL Database and SQL Data Warehouse by using the identities in Azure Active Directory
- ★ A Server Admin called the **Active Directory Administrator** must be created to use Azure AD authentication with SQL Database

Network Security

Access Management

Threat Protection

Information Protection

- ★ **Authorization** refers to the permissions assigned to a user within Azure SQL Database, and it determines what the user is allowed to do
- ★ Permissions are controlled by adding user accounts to database roles and assigning database-level permissions to those roles or by granting a user certain object-level permissions



Azure SQL Security Capabilities

Network Security

Access Management

Threat Protection

Information Protection

Row-level Security

Row-level security enables customers to control access to rows in a database table based on the characteristics of the user executing a query

Example: Group membership or execution context



Azure SQL Security Capabilities



Network Security

Access Management

Threat Protection

Information Protection

- ★ SQL Database secures customer data by providing auditing and threat detection capabilities

SQL Auditing in Azure Monitor Logs and Event Hubs

SQL Database auditing tracks database activities and helps maintain compliance with security standards by recording database events to an audit log in a customer-owned Azure storage account



Azure SQL Security Capabilities



Network Security

Access Management

Threat Protection

Information Protection

Advanced Threat Protection

Advanced Threat Protection is analyzing our SQL Server logs to detect unusual behavior and potentially harmful attempts to access or exploit databases



Network Security

Access Management

Threat Protection

Information Protection



Transport Layer Security (Encryption-in-transit)

- ★ SQL Database secures customer data by encrypting it in motion with **Transport Layer Security (TLS)**
- ★ SQL Server enforces encryption (SSL/TLS) at all times for all connections
- ★ This ensures that all data is encrypted 'in transit' between the client and the server, irrespective of the setting of Encrypt or TrustServerCertificate in the connection string

Network Security

Access Management

Threat Protection

Information Protection



Transport Layer Security (Encryption-in-transit)

For example: When using the ADO.NET driver, this is accomplished via **Encrypt=True** and **TrustServerCertificate=False**. If we have obtained our connection string from Azure Portal, it will have the correct settings.

Network Security

Access Management

Threat Protection

Information Protection



Transparent Data Encryption (Encryption-at-rest)

- ★ Transparent Data Encryption (TDE) for Azure SQL Database adds a layer of security to help protect data at rest (the raw files or backup) from unauthorized or offline access
- ★ In Azure, all newly created SQL databases are encrypted by default, and the database encryption key is protected by a built-in server certificate

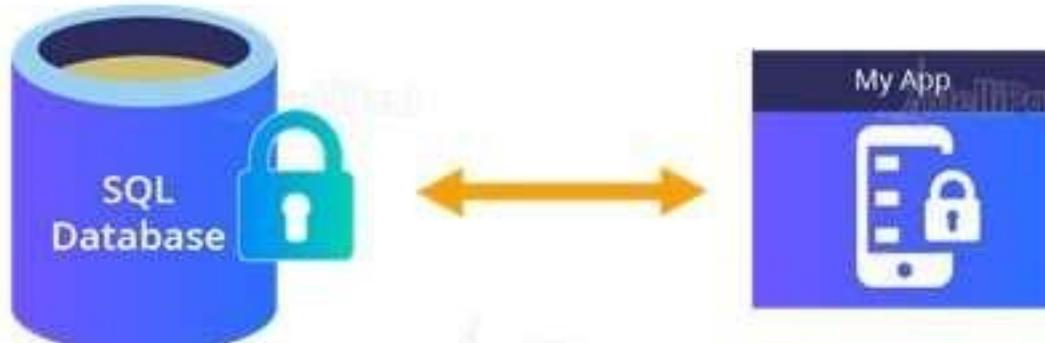


Key Management with Azure Key Vault

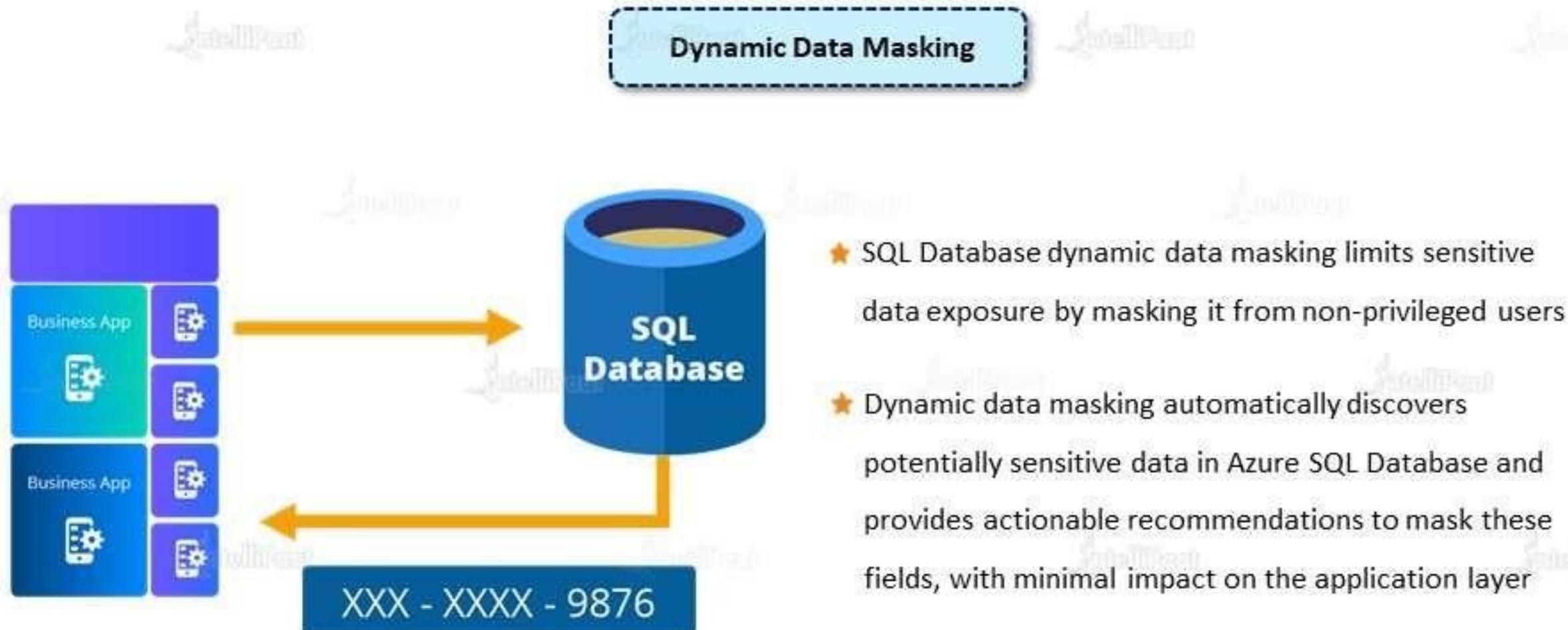
- ★ Bring Your Own Key (BYOK) support for Transparent Data Encryption (TDE) allows customers to take ownership of key management and rotation using Azure Key Vault, Azure's cloud-based external key management system
- ★ Azure Key Vault provides a central key management platform, leverages tightly monitored hardware security modules (HSMs), and enables the separation of duties between management of keys and data to help meet security compliance requirements

Always Encrypted (Encryption-in-use)

- ★ Always Encrypted is a feature designed to protect sensitive data stored in specific database columns from access
- ★ **Example:** credit card numbers, national identification numbers, etc.



Azure SQL Security Capabilities



High Availability and Azure SQL Database

- ★ The goal of the High Availability architecture in Azure SQL Database is to guarantee that our database is up and running 99.99 percent of time, without worrying about the impact of maintenance operations and outages
- ★ The high-availability solution is designed to ensure that the committed data is never lost due to failures, that maintenance operations do not affect our workload, and that the database will not be a single point of failure in our software architecture



There are two high-availability architectural models that are used in Azure SQL Database:

- ★ **Standard Availability Model**
- ★ **Premium Availability Model**

Standard Availability Model

Premium Availability Model

Standard Availability Model

- ★ This model is based on a separation of compute and storage
- ★ It relies on high availability and reliability of the remote storage tier
- ★ This architecture targets budget-oriented business applications that can tolerate some performance degradation during maintenance activities

Standard Availability Model

Premium Availability Model

Standard Availability Model

The standard availability model includes two layers:

- ★ Stateless Compute Layer
- ★ Stateful Data Layer

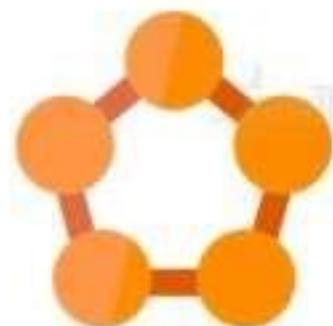


Standard Availability Model

Premium Availability Model

Stateless Compute Layer

- ★ Stateless compute layer runs the **sqlservr.exe** process and contains only transient and cached data, such as TempDB, model databases on the attached SSD, and plan cache, buffer pool, and columnstore pool in memory
- ★ This stateless node is operated by Azure Service Fabric that initializes **sqlservr.exe**, controls the health of the node, and performs a failover to another node if necessary



Standard Availability Model

Premium Availability Model

Stateful Data Layer

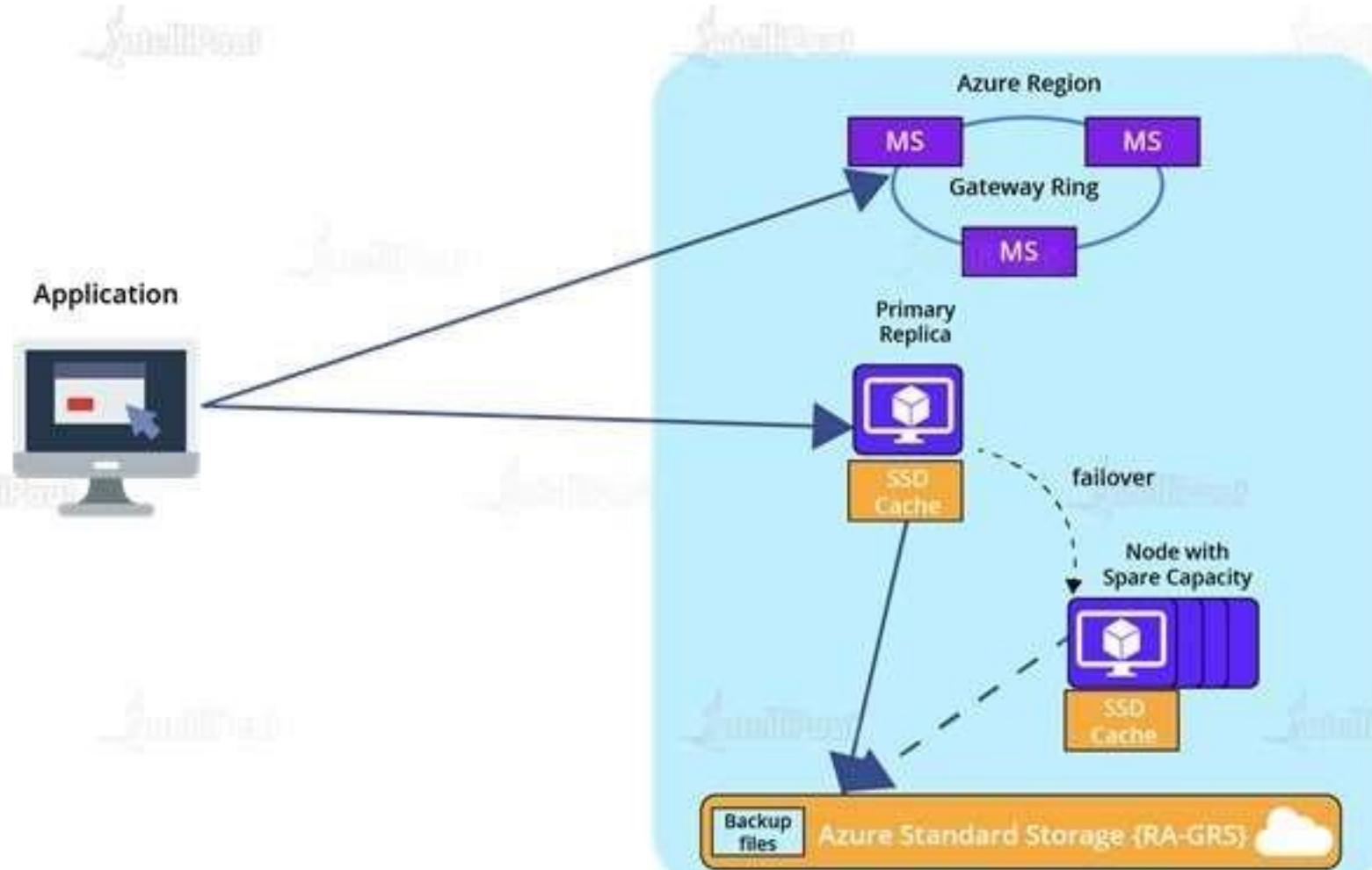
- ★ A stateful data layer, along with the database files (.mdf/.ldf), is stored in Azure Blob Storage
- ★ Azure Blob Storage has built-in data availability and redundancy features
- ★ It guarantees that every record in a log file or page in a data file will be preserved even if the SQL Server process crashes



High Availability and Azure SQL Database

Standard Availability Model

Premium Availability Model



Standard Availability Model

Premium Availability Model

Premium Availability Model

- ★ This model is based on a cluster of database engine processes
- ★ It relies on the fact that there is always a quorum of available database engine nodes
- ★ This architecture targets mission-critical applications with high I/O performance and a high transaction rate and guarantees minimal performance impact to our workload during maintenance activities

Standard Availability Model

Premium Availability Model

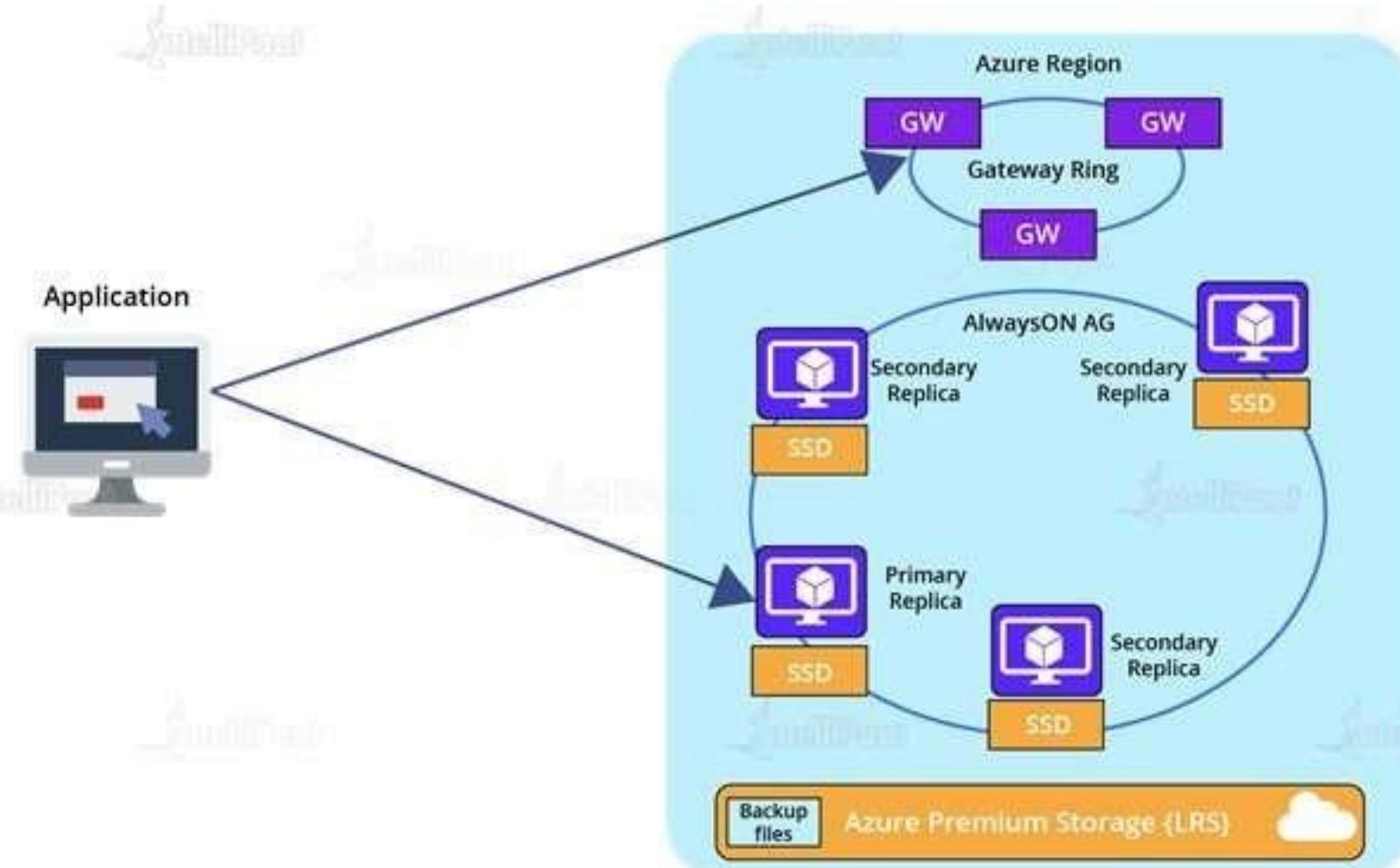
Premium Availability Model

- ★ Premium/Business-critical service tiers leverage the premium availability model, which integrates compute resources (the SQL Server Database Engine process) and storage (locally attached SSD) on a single node
- ★ High availability is achieved by replicating both compute and storage to additional nodes creating a 3-to-4-node cluster
- ★ The underlying database files (.mdf/.ldf) are placed on the attached SSD storage to provide very low latency I/O to our workload

High Availability and Azure SQL Database

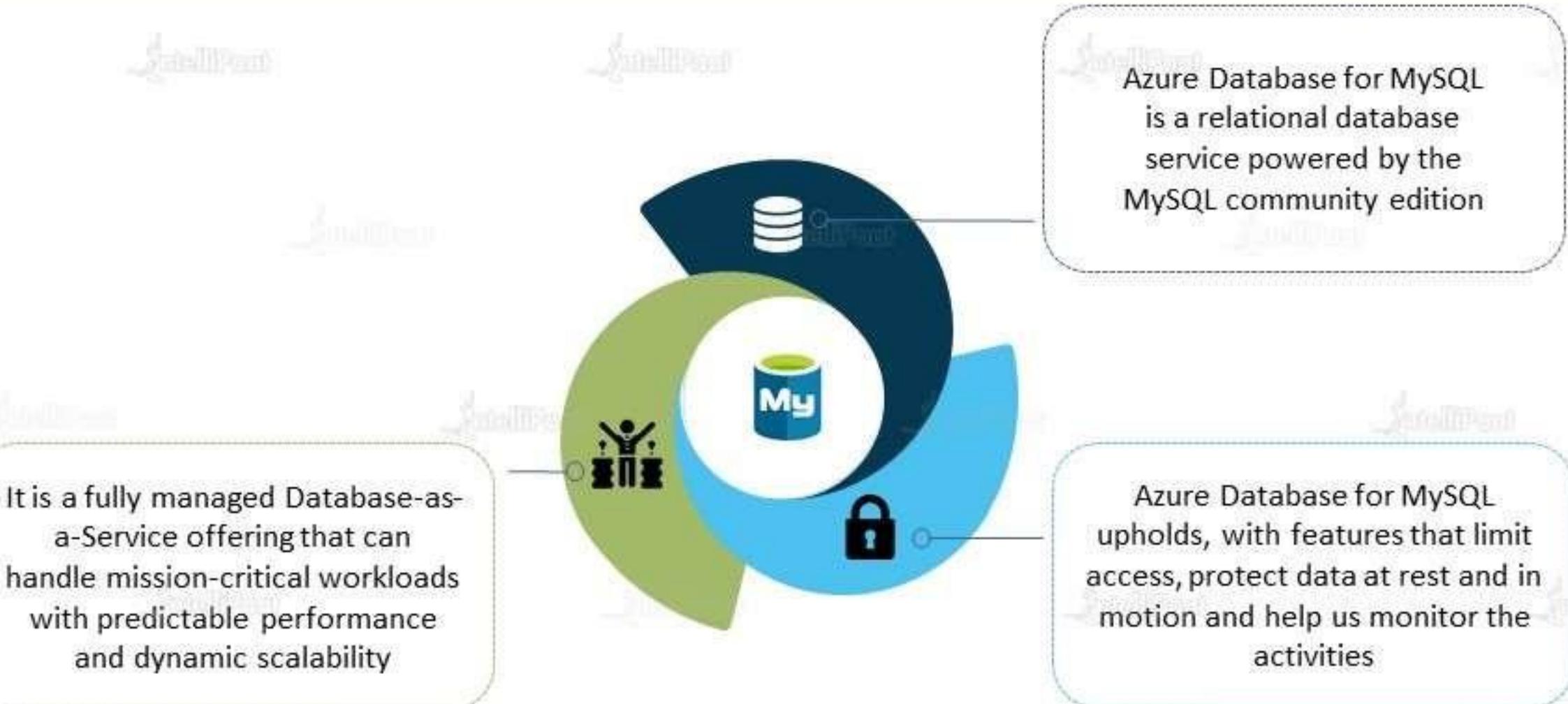
Standard Availability Model

Premium Availability Model



Azure Database for MySQL

Azure Database for MySQL



Azure Database for MySQL

Azure Database for MySQL delivers:

Enterprise-grade security and compliance

Automatic backups and point-in-time restore for up to 35 days

Secured protection of sensitive data at rest and in motion

Built-in high availability with no additional cost

Predictable performance, using the inclusive pay-as-you-go pricing

Scaling as needed within seconds



Which MySQL option to choose?



With Azure, our MySQL server workloads can run in a hosted virtual machine infrastructure as a service (IaaS) or as a hosted platform as a service (PaaS)



When we choose between IaaS and PaaS, we must decide if our aim is to manage our database, apply patches, and make backups

MySQL on Azure VMs



This option falls into the industry category of IaaS. With this service, we can run MySQL Server inside a fully managed virtual machine on the Azure cloud platform

01



All recent versions and editions of MySQL can be installed on an IaaS virtual machine

02



The most significant difference from Azure Database for MySQL is that MySQL on Azure VMs offers control over the database engine

03

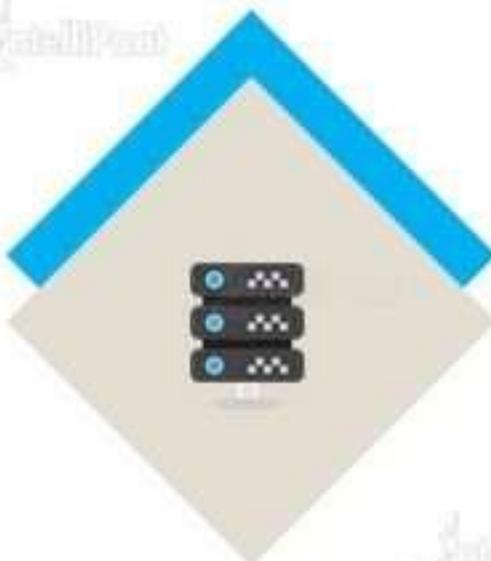
Hands-on: Designing an Azure Database for MySQL Using Azure Portal

Hands-on: Connecting Using MySQL

Workbench

Azure Database for PostgreSQL

Azure Database for PostgreSQL



Azure Database for PostgreSQL is a relational database service in the Microsoft cloud built for developers



It is based on the community version of open-source PostgreSQL database engine



It is available in two deployment options:
Single Server and
Hyperscale (Citus)

Single Server

The Single Server deployment option delivers:

Monitoring and alerting to assess our server

Automatic backups and point-in-time-restore for up to 35 days

Secured to protect sensitive data at-rest and in-motion

Built-in high availability with no additional cost (99.99% SLA)

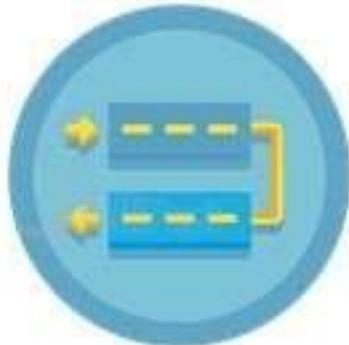
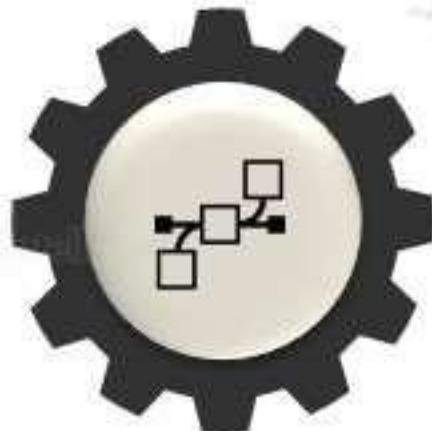
Predictable performance, using inclusive pay-as-you-go pricing

Vertical scale as needed within seconds



Hyperscale (Citus)

The **Hyperscale (Citus)** option horizontally scales queries across multiple machines using sharding

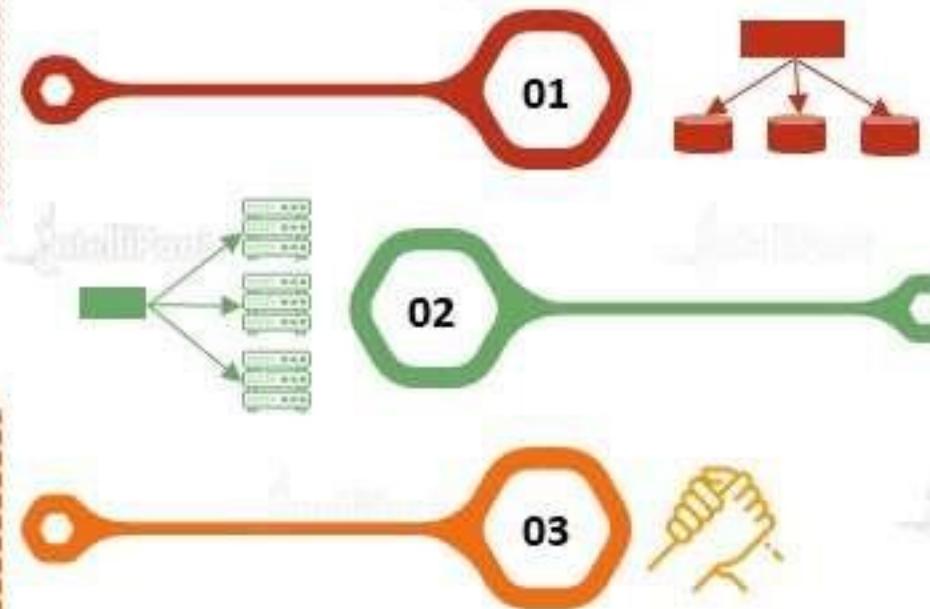


Its query engine parallelizes the incoming SQL queries across these servers for faster responses on large datasets

Hyperscale (Citus)

The Hyperscale deployment option delivers:

Horizontal scaling across multiple machines using sharding



Excellent support for multi-tenant applications, real-time operational analytics, and high-throughput transactional workloads

Query parallelization across multiple servers for faster responses on large datasets

Hands-on: Designing an Azure Database for PostgreSQL - Single Server

Azure Database for MariaDB

Azure Database for MariaDB is a relational database service in the Microsoft cloud



It is based on the MariaDB community edition database engine, version 10.2 and 10.3

Azure Database for MariaDB

Azure Database for MariaDB delivers:

Enterprise-grade security and compliance

Automatic backups and point-in-time restore for up to 35 days

Secured protection of sensitive data at rest and in motion

Built-in high availability with no additional cost

Predictable performance, using the inclusive pay-as-you-go pricing

Scaling as needed within seconds



Azure Database for MariaDB

Azure Database for MariaDB can help us rapidly develop our app and accelerate our Time to Market



We can also continue to develop our application by using the open-source tools and a platform of our choice

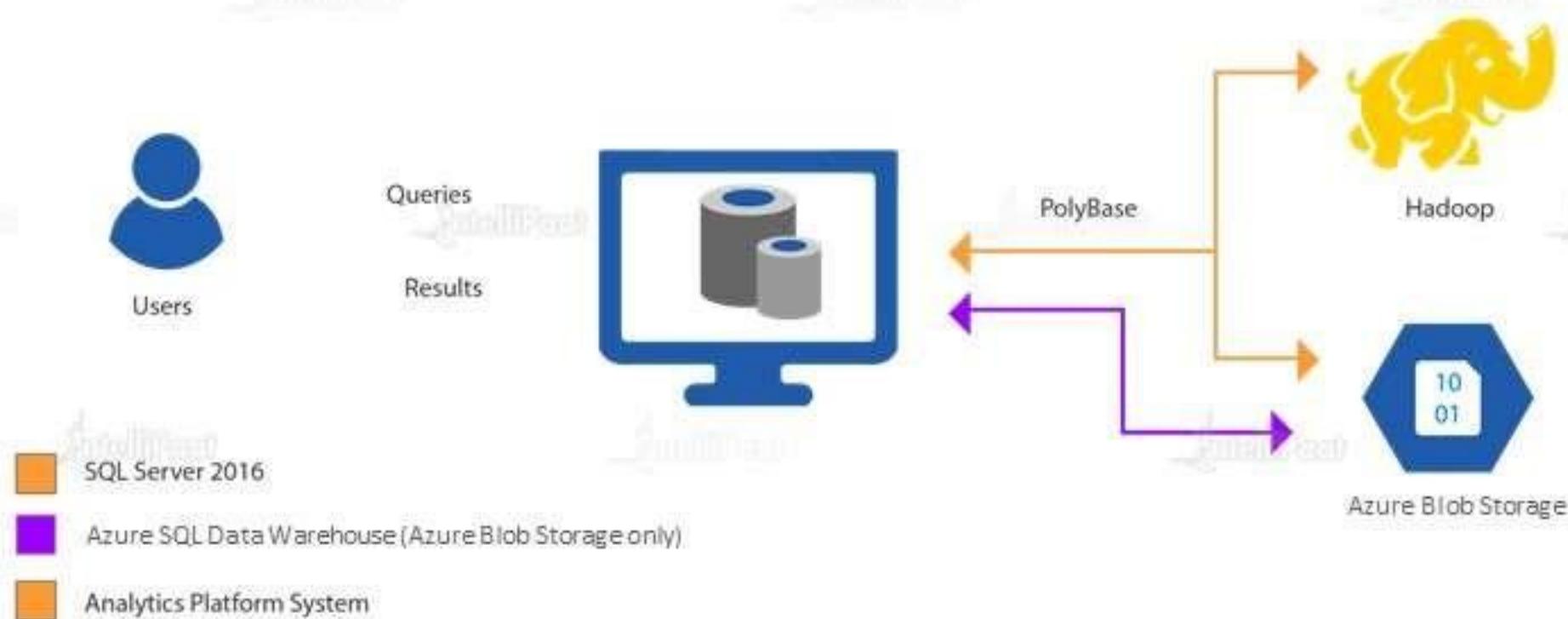
Azure Database for MariaDB delivers speed and efficiency for our business demands, all without learning new skills



Hands-on: Creating an Azure Database for MariaDB Server by Using Azure Portal

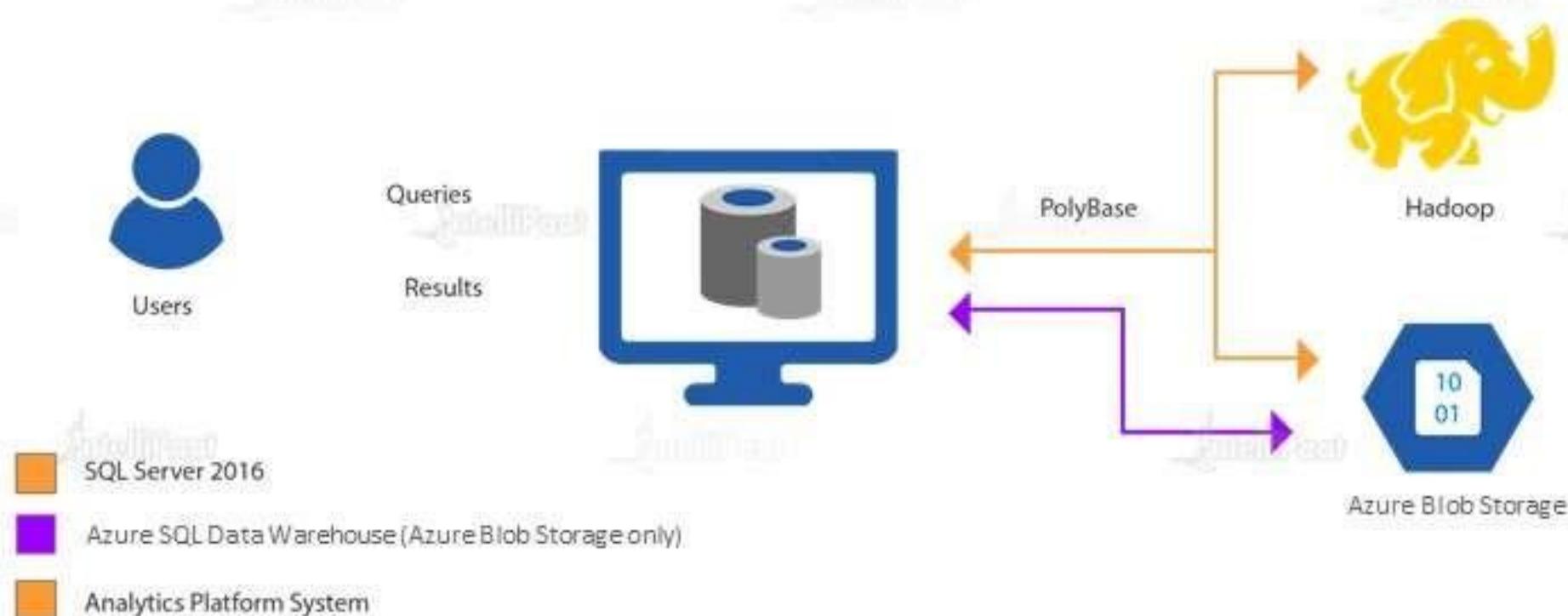
What is PolyBase?

- PolyBase enables our SQL Server 2016 instance to process Transact-SQL queries that read data from Hadoop
- The same query can also access the relational tables in our SQL Server



PolyBase

- In SQL Server, an **external table** or an **external data source** provides the connection to Hadoop
- PolyBase pushes some computations to the Hadoop node to optimize the overall query



Azure Integration



01



With the underlying help of PolyBase, T-SQL queries can also import and export data from Azure Blob Storage



02



PolyBase enables Azure SQL Data Warehouse to import and export data from Azure Data Lake Store and from Azure Blob Storage



Why PolyBase?

In the past, it was more difficult to join our SQL Server data with the external data. Then, we had only the following two unpleasant options:

- Transfer half of our data so that all our data is in one format or the other
- Query both sources of the data and then write custom query logic to join and integrate the data at the client level



PolyBase avoids these unpleasant options by using T-SQL to join the data

Why PolyBase?

PolyBase enables the following scenarios in SQL Server:

Integrating with BI tools

Querying the data stored in Hadoop from SQL Server or PDW

Exporting data to Hadoop, Azure Blob Storage, or Azure Data Lake Store

Querying the data stored in Azure Blob Storage



Importing data from Hadoop, Azure Blob Storage, or Azure Data Lake Store

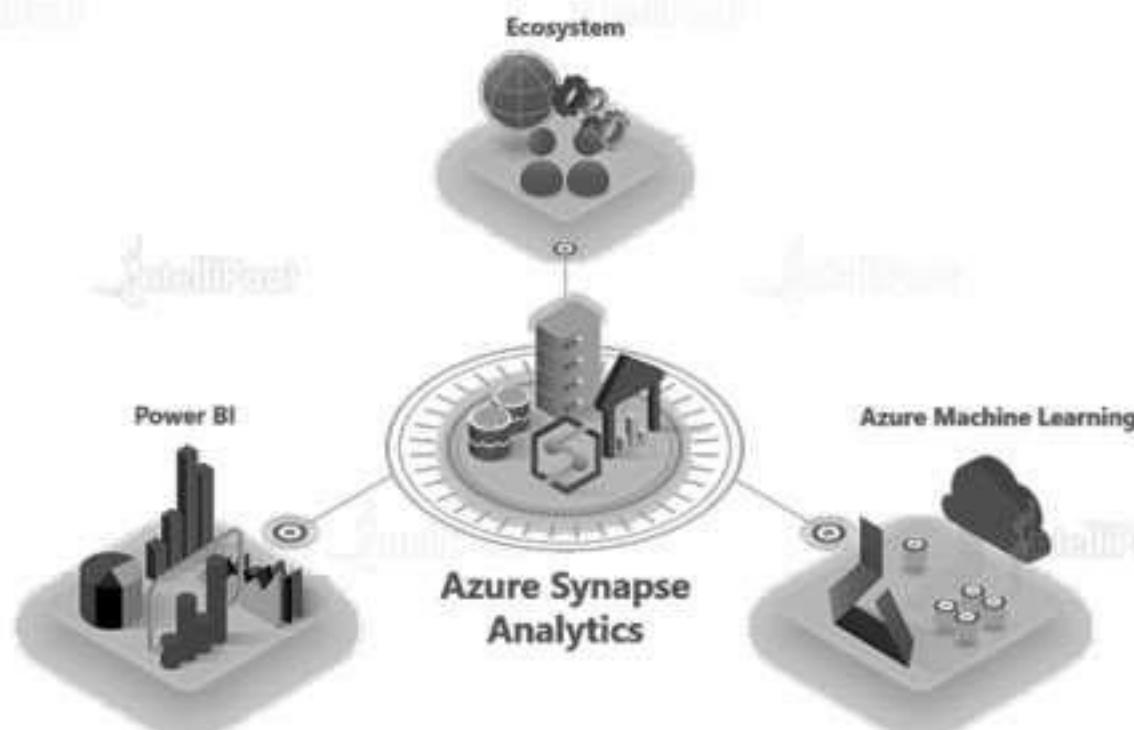
What is Azure Synapse Analytics (Formerly, SQL DW)?

01

Azure Synapse is a limitless analytics service that brings together enterprise data warehousing and Big Data Analytics



Azure Synapse brings these two worlds together with a unified experience to ingest, prepare, manage, and serve data for immediate **BI** and **Machine Learning** needs



02

It gives us the freedom to query data on our terms, using either serverless on-demand or provisioned resources at scale



Azure Synapse has four components:

Complete T-SQL-based analytics

- SQL pool
- SQL on-demand

SQL Analytics



Spark

Deeply integrated Apache Spark

Hybrid data integration

Data Integration

Studio

Unified user experience

01



SQL Analytics refers to the enterprise data warehousing features that are generally available in Azure Synapse

02



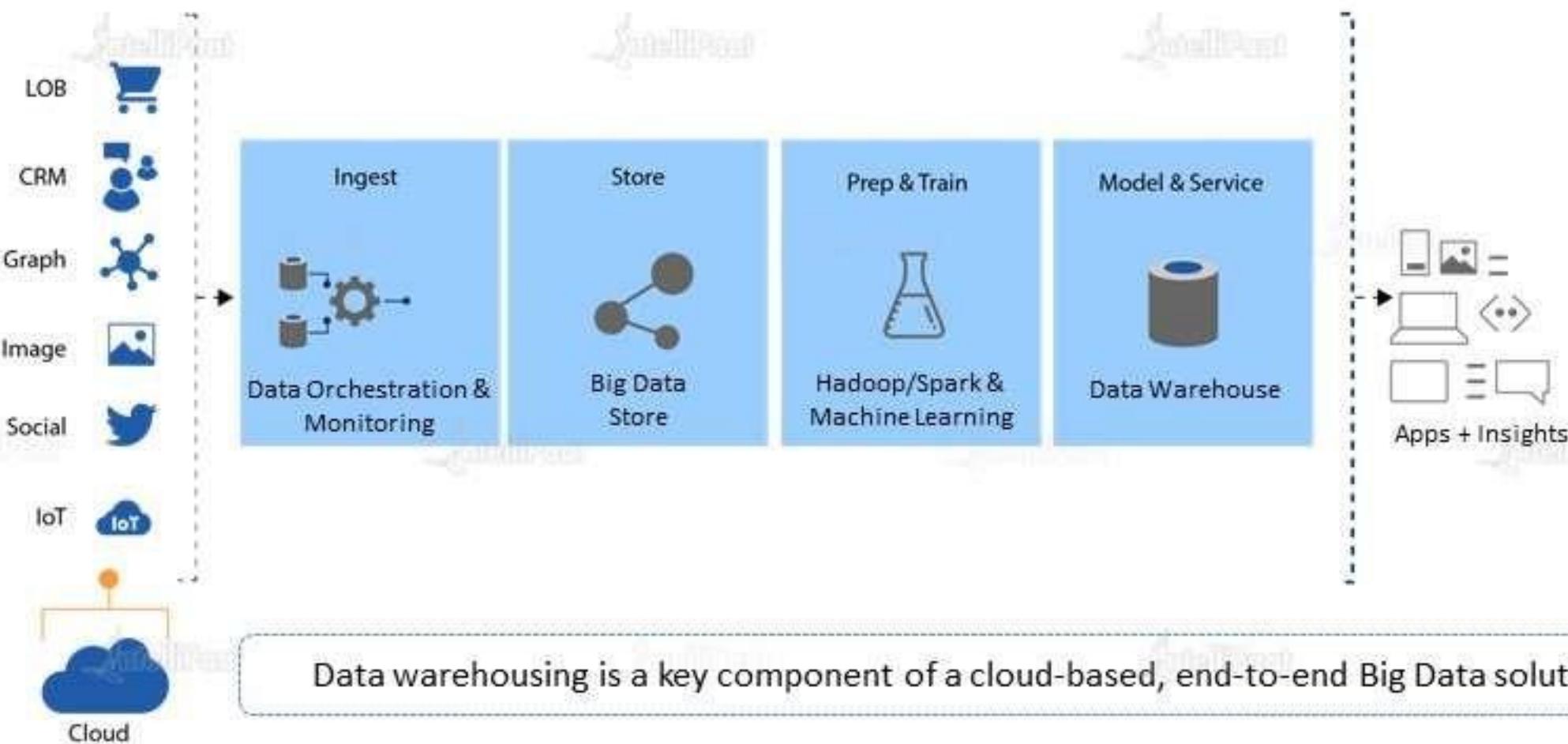
SQL pool represents a collection of analytic resources that are being provisioned when using SQL Analytics. The size of the SQL pool is determined by Data Warehousing Units (DWU)

03

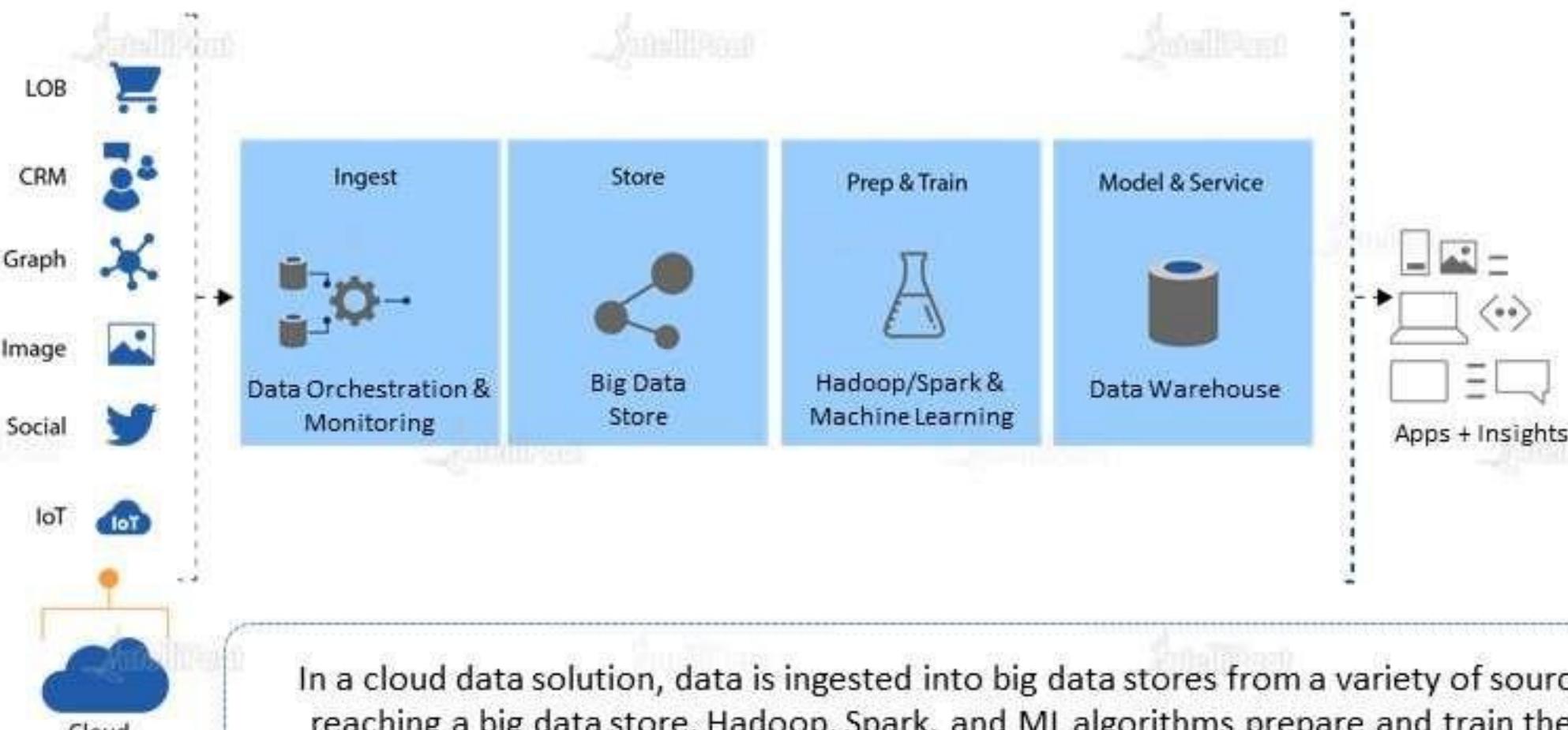


We can import big data with simple PolyBase T-SQL queries and then use the power of Massive Parallel Processing (MPP) to run high-performance analytics

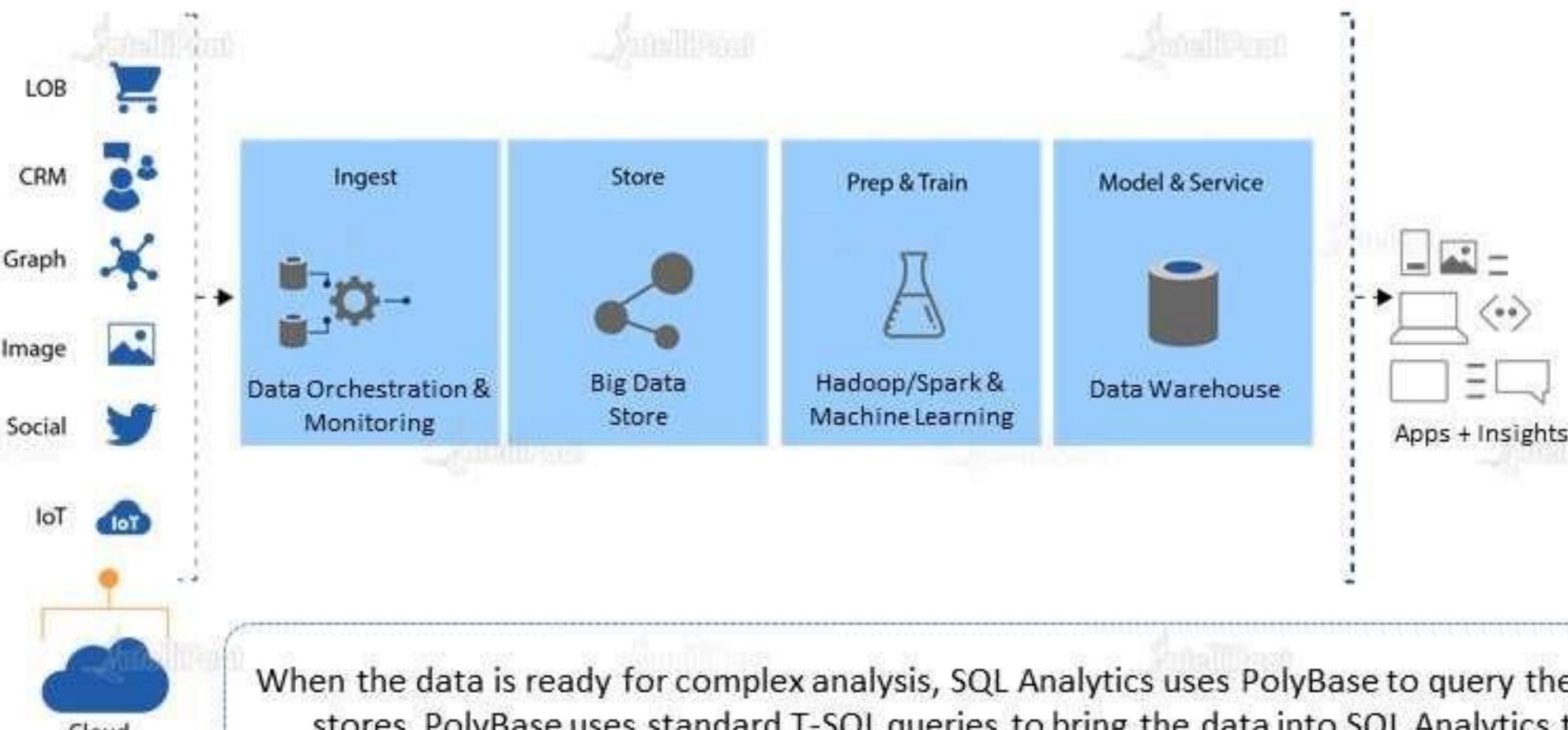
Key Components of a Big Data Solution



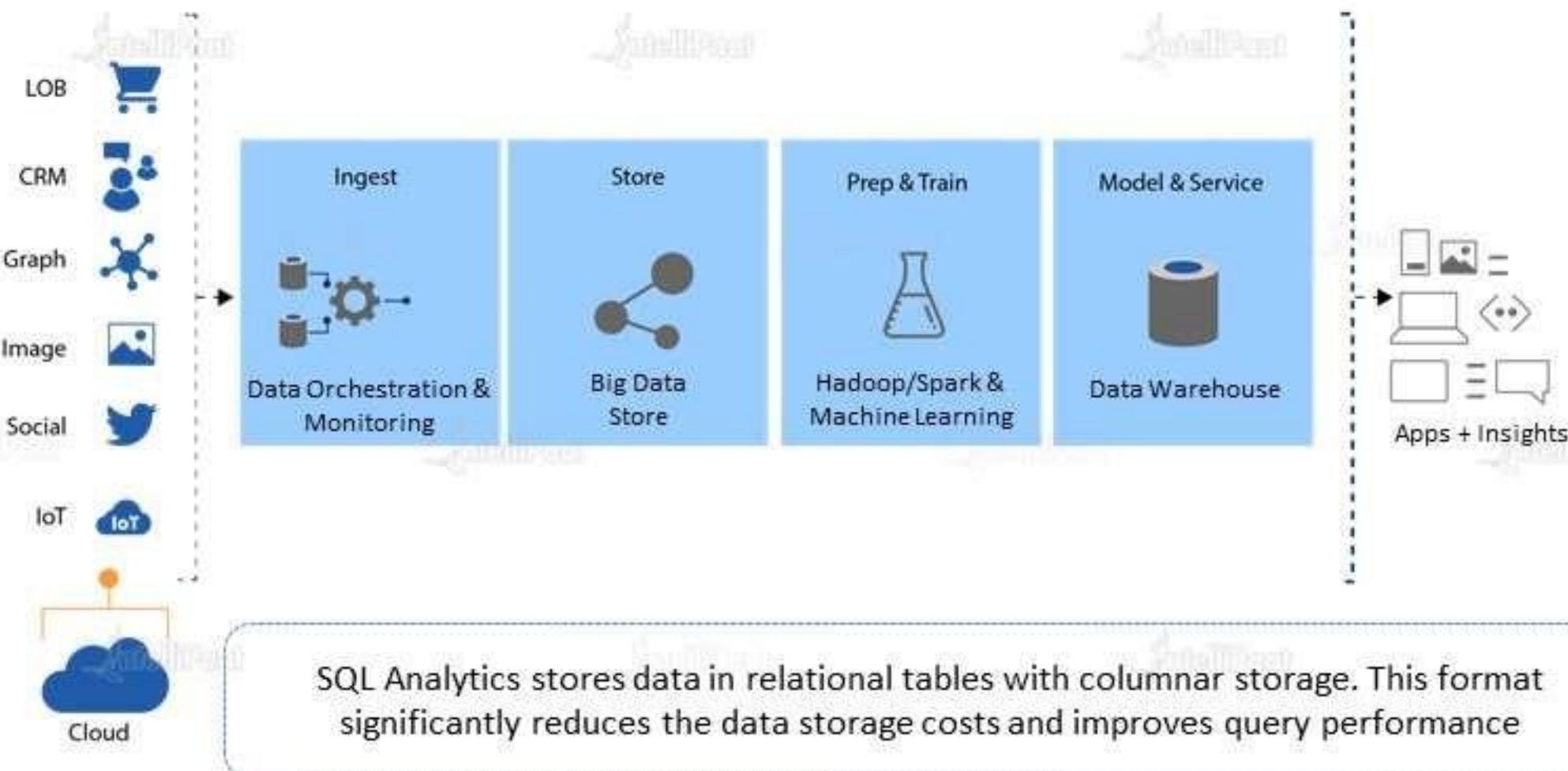
Key Components of a Big Data Solution



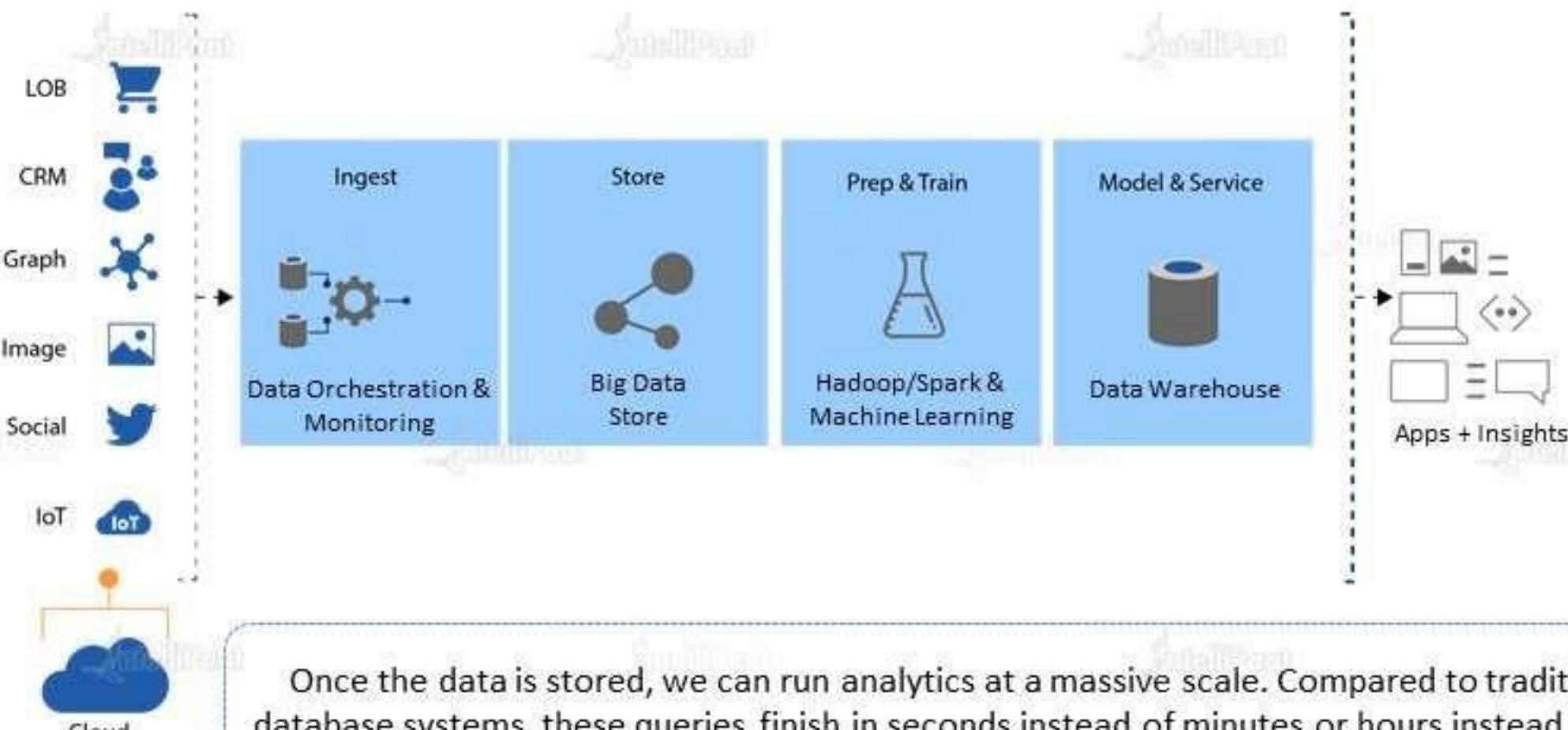
Key Components of a Big Data Solution



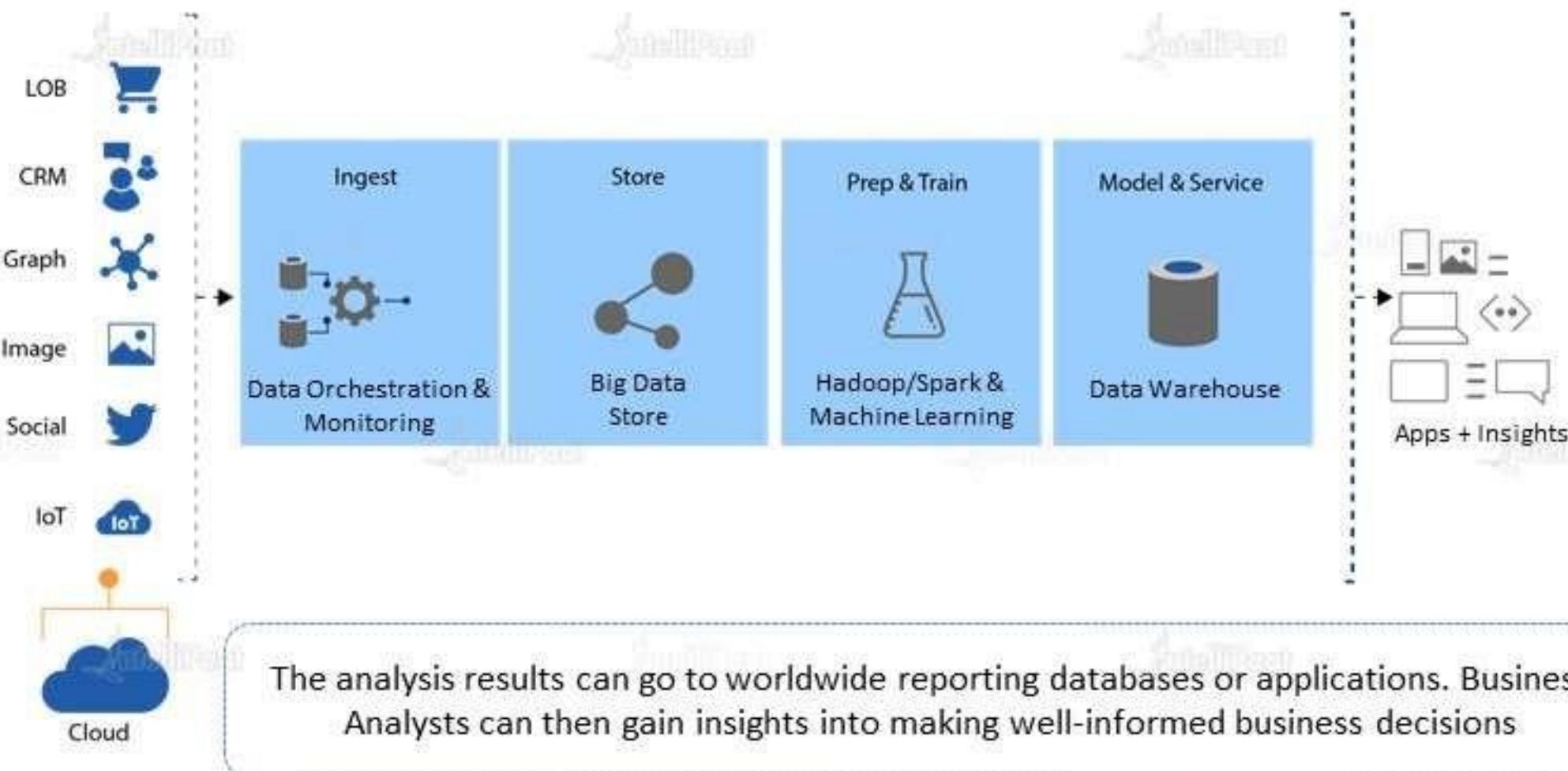
Key Components of a Big Data Solution



Key Components of a Big Data Solution

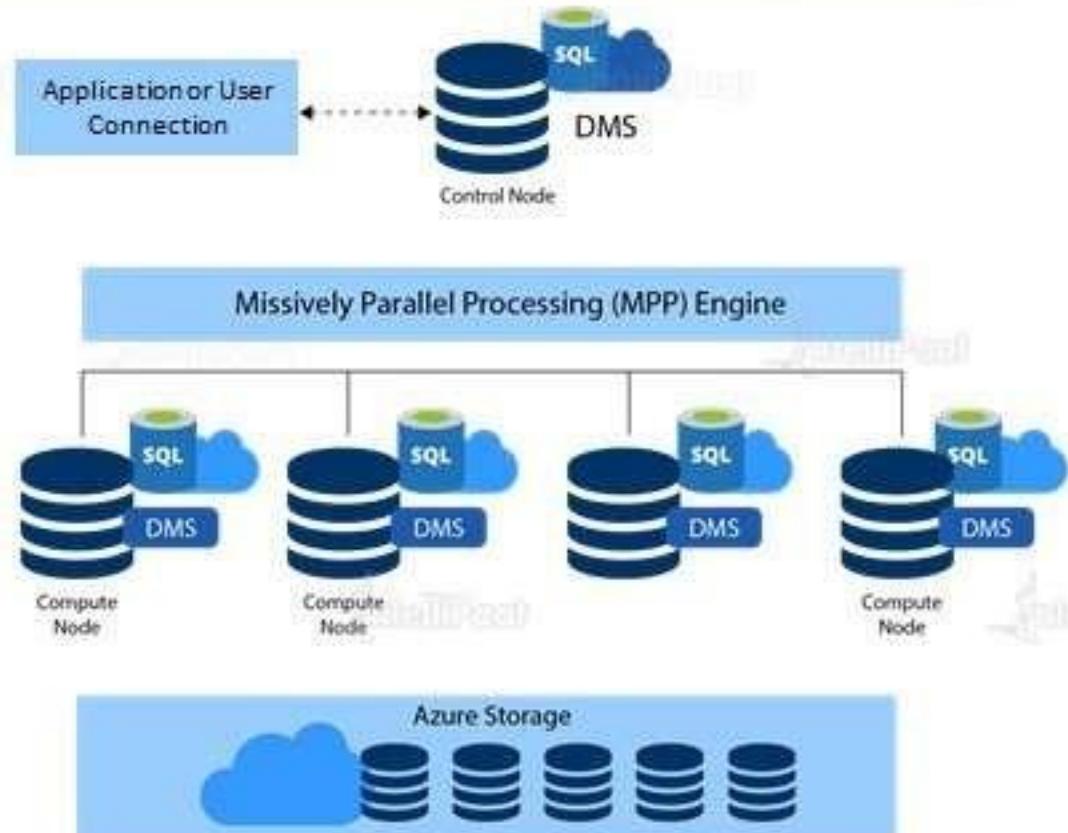


Key Components of a Big Data Solution



SQL Analytics MPP Architecture Components

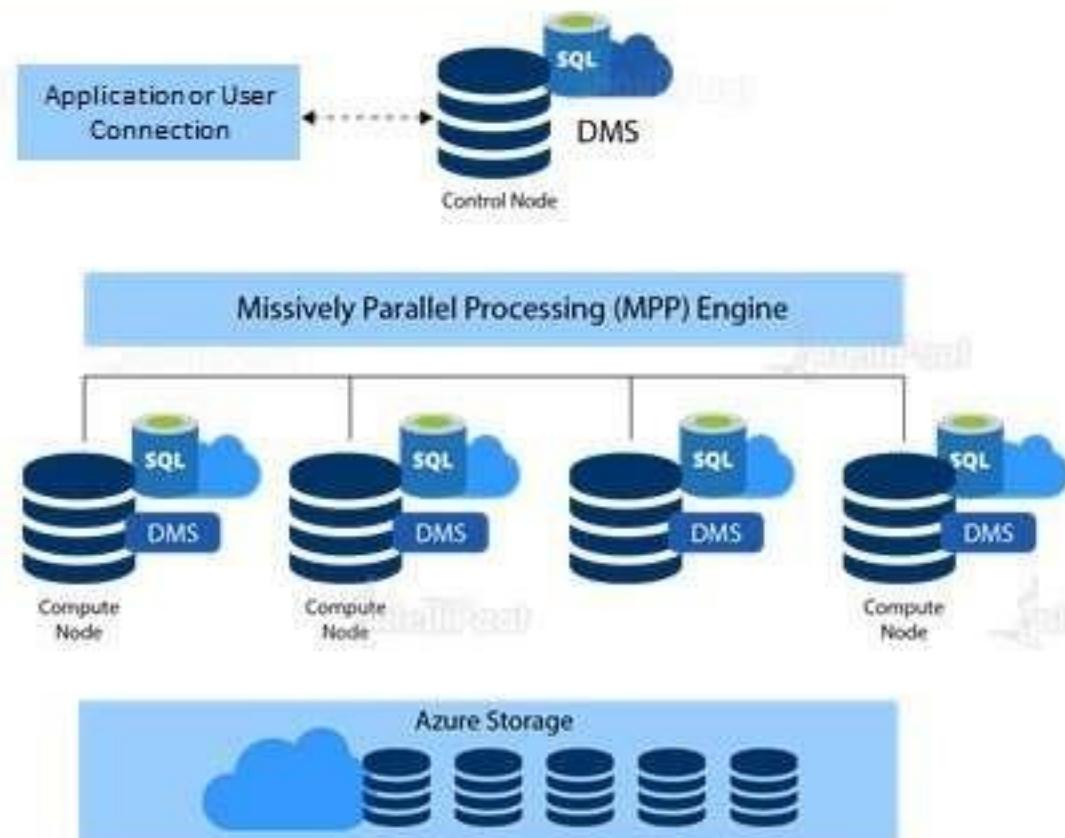
- SQL Analytics leverages a scale-out architecture to distribute computational processing of data across multiple nodes
- The unit of scale is an abstraction of compute power that is known as a data warehouse unit



Compute is separate from storage, which enables us to scale compute independently of the data in our system

SQL Analytics MPP Architecture Components

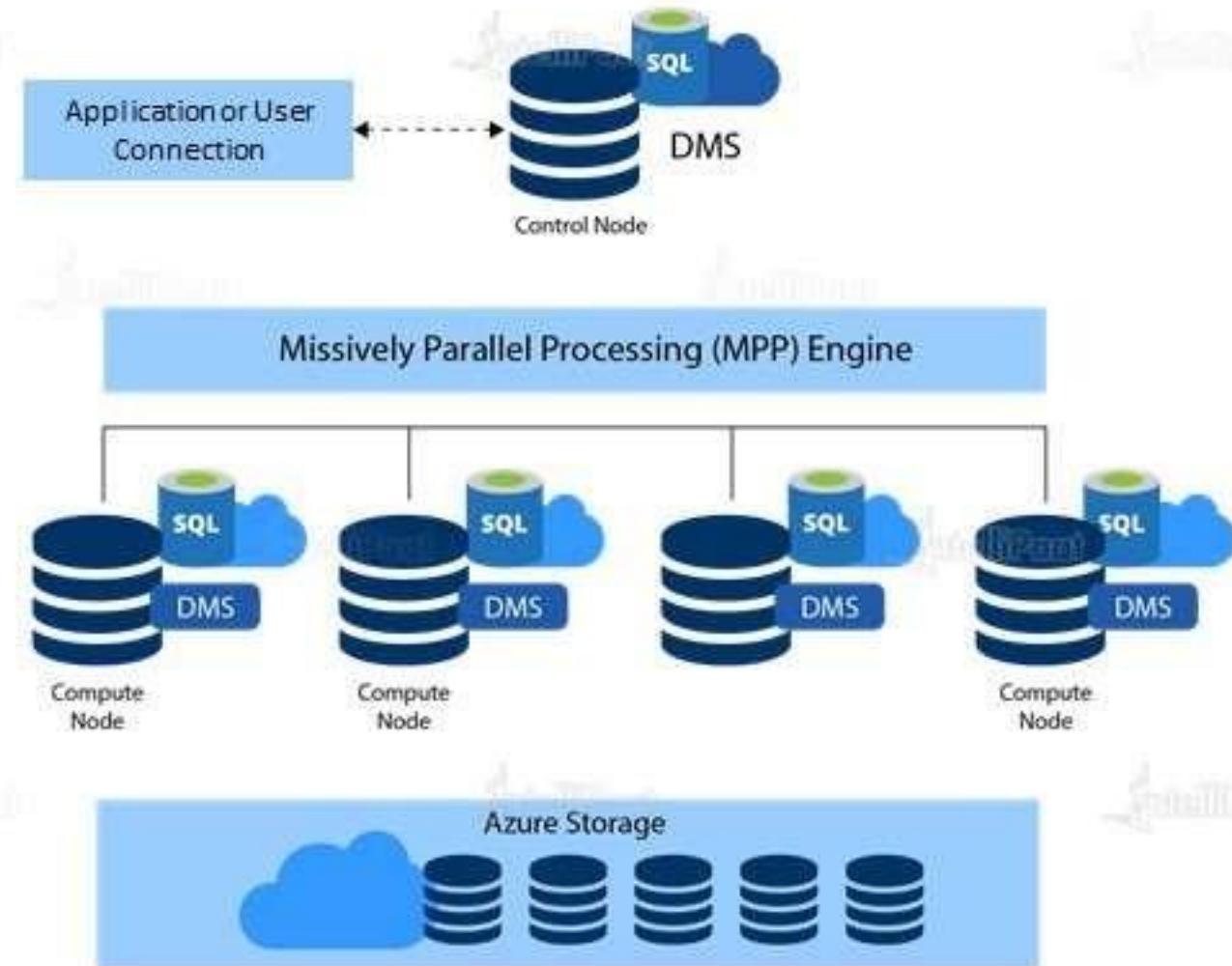
- SQL Analytics uses a node-based architecture
- Applications connect and issue T-SQL commands to a Control node, which is the single point of entry for SQL Analytics



The Control node runs the MPP engine, which optimizes queries for parallel processing, and then passes operations to the Compute nodes to do the work in parallel

SQL Analytics MPP Architecture Components

- The Compute nodes store all user data in Azure Storage and run the parallel queries
- The Data Movement Service (DMS) is a system-level internal service that moves data across the nodes as necessary to run queries in parallel and return accurate results



SQL Analytics MPP Architecture Components

With decoupled storage and compute, when using SQL Analytics, we can

01

02

03

04

Independently size the compute power irrespective of our storage needs

Grow or shrink the compute power, within a SQL pool (data warehouse), without moving data

Pause compute capacity while leaving the data intact so that we only pay for storage

Resume compute capacity during operational hours

Hands-on: Importing Data from Blob Storage to Azure Synapse Analytics Using PolyBase



India: +91-7847955955



sales@intellipaat.com



24/7 Chat with Our Course Advisor