Module 7: Comparing Database Options in Azure

Contents:

Module overview

This module compares the various relational and non-relational data storage options available in Azure. Options are explored as groups such as relational databases (Azure SQL Database, MySQL, and PostgreSQL on Azure), non-relational (Azure Cosmos DB, Storage Tables), streaming (Stream Analytics) and storage (Data Factory, Data Warehouse, Data Lake).

Objectives

After completing this module, students will be able to:

- Compare and contrast various database options on Azure.
- Identify data streaming options for large-scale data ingest.
- Identify longer-term data storage options.

Lesson 1: Relational Databases

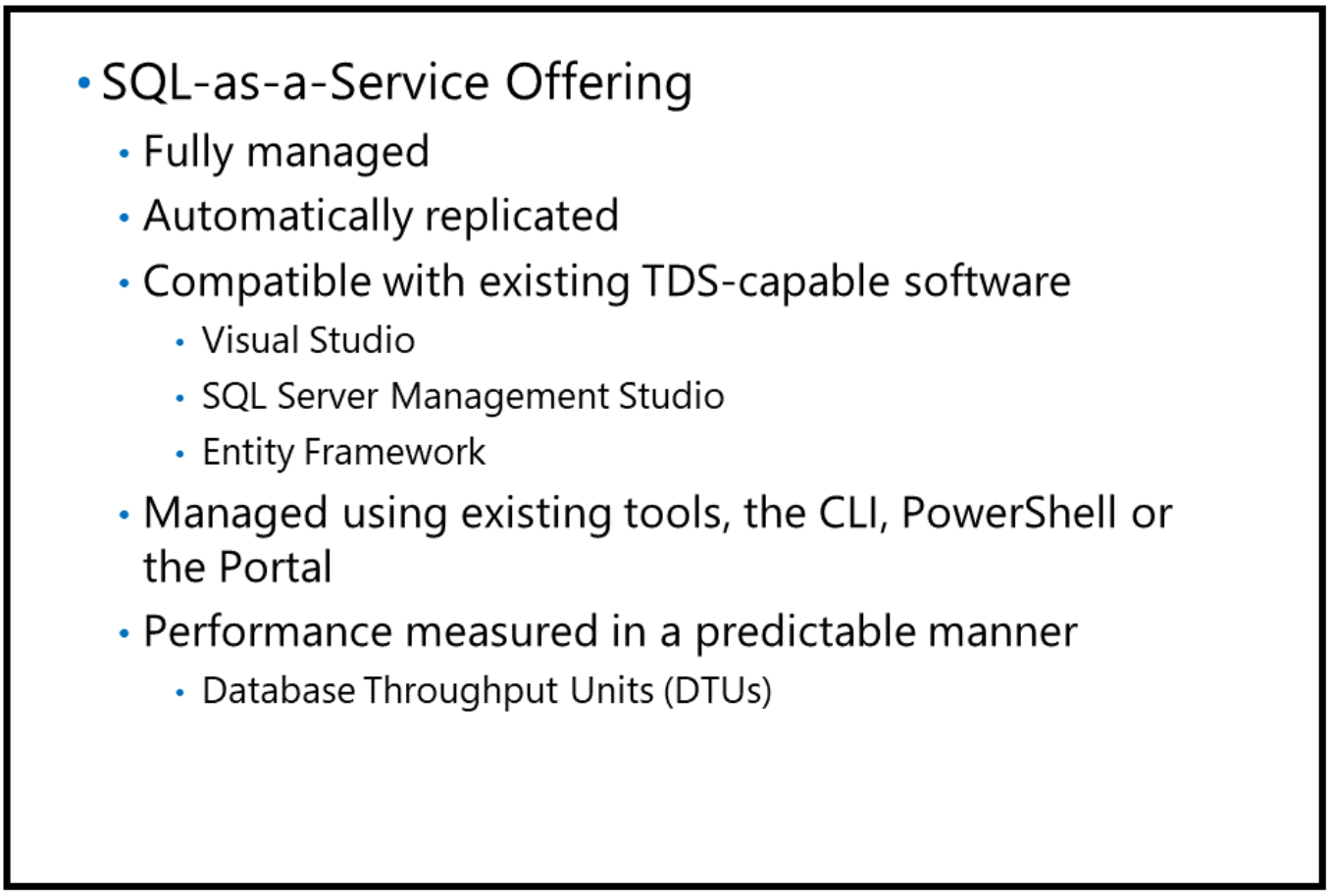
This lesson discusses the three managed database services in Azure; Azure SQL Database, Azure Database for MySQL, and Azure Database for PostgreSQL.

Lesson objectives

After completing this lesson, you will be able to:

- Describe the difference between the three services for SQL, MySQL and PostgreSQL.
- Determine when to use advanced features of Azure SQL Database such as Elastic Database and Stretch Database.

Azure SQL Database



Azure SQL Database

SQL Database is a relational database as-a-service offering that provides predictable performance and a high degree of compatibility with existing management tools.

Predictable Performance

By using a consistent unit of measurement, such as Database Throughput Units, you can compare the expected service level for each performance tier that is offered in the SQL Database service. Consistent and predictable performance allows you to select a tier that very closely matches your application's real-world utilization.

High Compatibility

A Tabular Data Stream (TDS) endpoint is provided for each logical server that is created in the SQL Database service. You can use existing SQL client applications and tools with SQL Database by using the TDS protocol.

Simple Management

Additional tools are available in Azure to manage databases that are created by SQL Database. A portal for managing database objects is available in the Azure Management Portal, which you can access by clicking the Manage button. You also can manage SQL Database instances by using the portals, REST API, Windows PowerShell, or the cross-platform command-line interface (Xplat CLI).

Database Throughput Unit (DTU)

DTUs are used to describe the capacity for a specific tier and performance level. DTUs are designed to be relative so that you can directly compare the tiers and performance levels. For example, the Basic tier has a single performance level (B) that is rated at 5 DTU. The S2 performance level in the Standard tier is rated at 50 DTU. This means that you can expect ten times the power for a database at the S2 performance level than a database at the B performance level in the Basic tier.

Azure SQL Database Tiers

The SQL Database service is offered in several tiers. You can select a tier that closely matches your application's intended or actual resource needs. The following is a list of SQL Database service tiers with the associated performance characteristics:

- **Basic:** Ideal for simple databases that requires only a single connection performing a single operation at a time.
- **Standard:** The most common option and is used for databases that require multiple concurrent connections and operations.
- **Premium:** Designed for applications that require large quantities of transactions at volume. These databases support a large quantity of concurrent connections and parallel operations.

These tiers are further separated into performance levels. Performance levels are very specific categories within a service tier that provides a specific level of service. For example, the P1 performance level in the Premium tier offers a maximum database size of 500 gigabyte (GB) and a benchmarked transaction rate of 105 transactions per second.

Every tier has one or more performance levels. In general, the performance levels in the Premium tier are rated higher than the performance levels in the Standard tier, which are again rated higher than the Basic tier. The following chart illustrates this distinction.

Azure SQL Database Elastic Scale

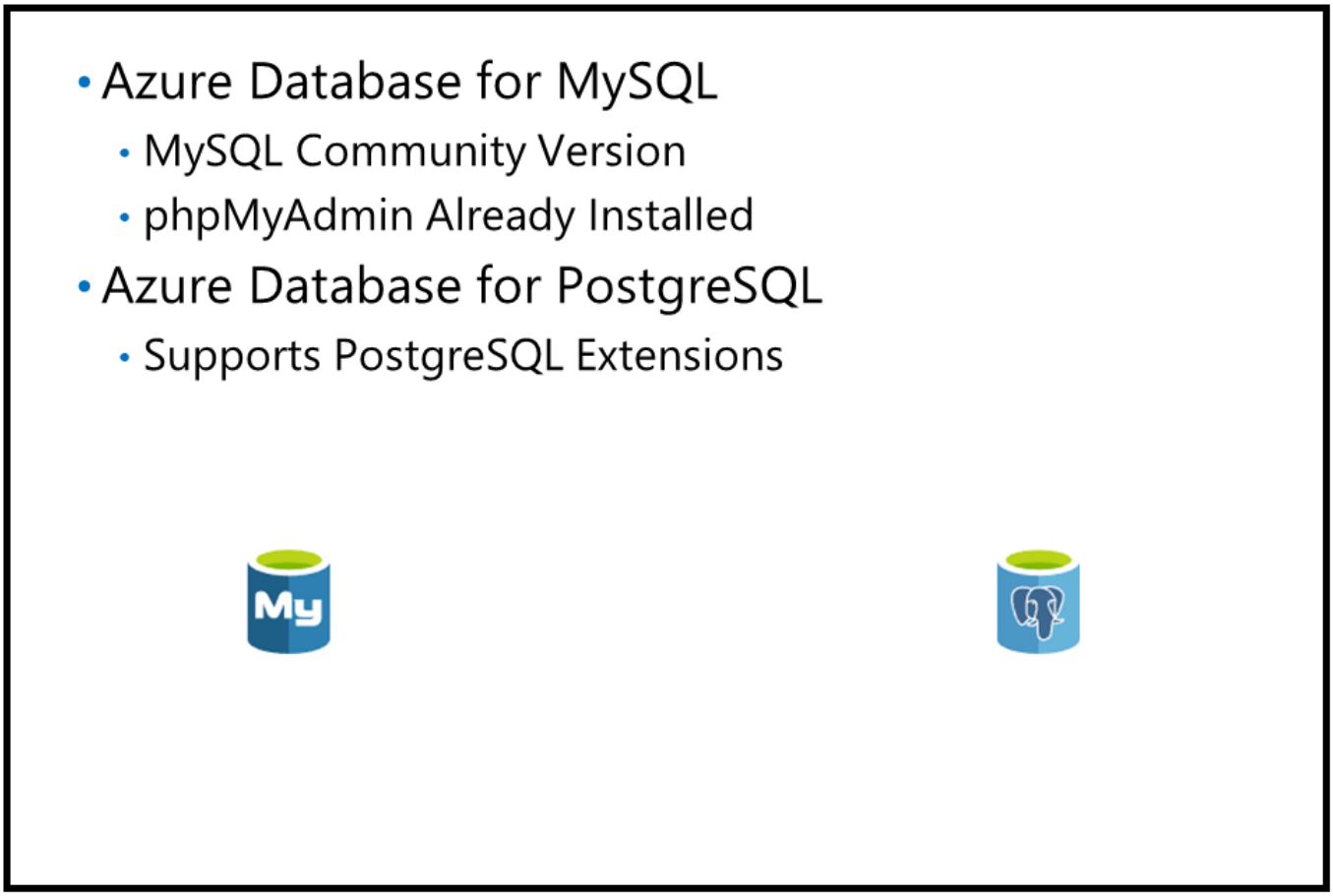
The new Elastic Scale capabilities simplify the process of scaling out (and in) a cloud application's data tier by streamlining development and management. Elastic Scale is composed of two main parts:

- An Elastic Scale library for client applications to configure shards and access shards.
- The Elastic Scale features in Azure SQL Database that implements the any changes requested by your application.

Elastic Scale implements the database scaling strategy known as sharding. As a developer, you can establish a "contract" that defines a shard key and how shards should be partitioned across a collection of databases. The

application, using the SDK, can then automatically direct transactions to the appropriate database (shard), perform queries across multiple shards or modify the service tier for existing shards. Elastic Scale also enables coordinated data movement between shards to split or merge ranges of data among different databases and satisfy common scenarios such as pulling a busy tenant into its own shard. The Split-Merge service is provided through a downloadable package that customers can deploy as an Azure cloud service into their own subscription.

Third-Party Databases in Azure



Azure provides two additional managed database options for applications running on Azure.

- **Azure Database for MySQL**: This is a managed MySQL instance running the MYSQL community version. The instance comes with tools preinstalled like **mysql.exe** and **phpMyAdmin**. You can run one or more databases with this instance.
- **Azure Database for PostgreSQL**: This is a managed PostgreSQL instance that can run one or more databases.

Both of the managed database services in Azure share a common set of features:

- Built-in high availability with no additional cost.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Scale on the fly within seconds.
- Secured to protect sensitive data at-rest and in-motion.

- Automatic backups and point-in-time-restore for up to 35 days.
- Enterprise-grade security and compliance.

Other Options

Using Windows or Linux virtual machines, you can always install and run MySQL in the Azure environment. ClearDB also provides a managed MySQL instance that you can create from the Azure Marketplace.

Lesson 2: NoSQL Services

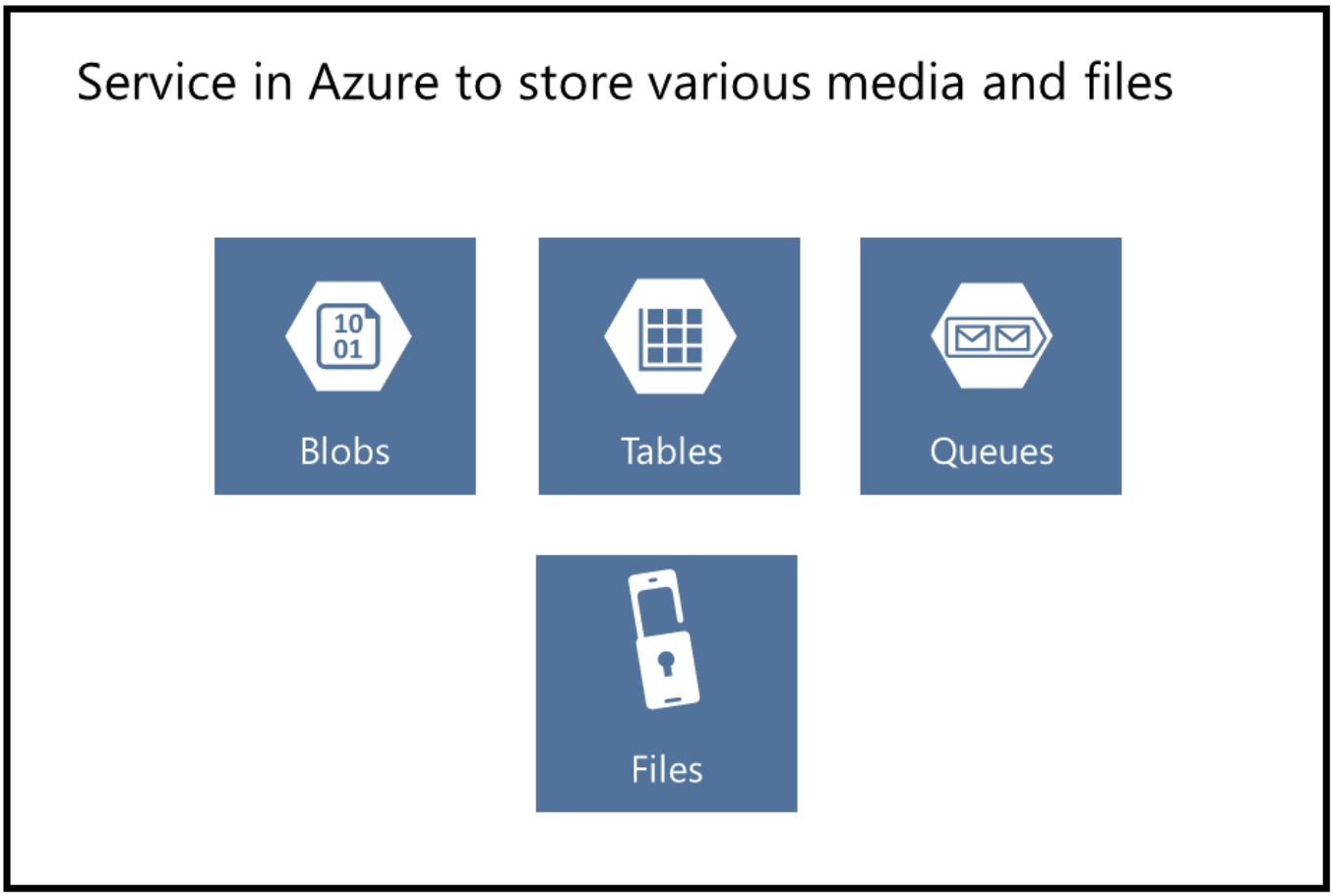
This lesson talks about some of the NoSQL services available in Azure to store and perform analysis on massive data sets.

Lesson objectives

After completing this lesson, you will be able to:

- Identify when to use Azure Storage Tables in a solution.
- Integrate Azure Search with existing data solutions.

Azure Storage



Azure Storage is massively scalable, so you can store and process hundreds of terabytes of data to support the big data scenarios required by scientific, financial analysis, and media applications. You can also store the small amounts of data required for a small business website as billing is calculated by usage, not capacity. Storage uses an auto-partitioning system that automatically load-balances your data based on traffic. Since storage is elastic and decoupled from your application, you can focus on your workload while relying on Storage's elastic capabilities to scale to meet demand for your applications.

All Storage services can be accessed using a REST API. Client libraries are also available for popular languages and platforms such as:

- .NET
- Java/Android
- Node.js
- PHP
- Ruby
- Python
- PowerShell

Finally, all resources in Storage can be protected from anonymous access and can be used in the Valet-Key pattern configuration discussed in previous modules.

Replication

The data in your Microsoft Azure storage account is always replicated to ensure durability and high availability. At a minimum, your data is stored in triplicate. You may also choose extended replication options for scenarios where you require your data to be replicated across geography.

- **Locally redundant storage (LRS).** Locally redundant storage maintains three copies of your data. LRS is replicated three times within a single facility in a single region. LRS protects your data from normal hardware failures, but not from the failure of a single facility. LRS is the minimum amount of replication.
- **Zone-redundant storage (ZRS)**. Zone-redundant storage maintains three copies of your data. ZRS is replicated three times across two to three facilities, either within a single region or across two regions, providing higher durability than LRS. ZRS ensures that your data is durable within a single region.
- **Geo-redundant storage (GRS)**. Geo-redundant storage is enabled for your storage account by default when you create it. GRS maintains six copies of your data. With GRS, your data is replicated three times within the primary region, and is also replicated three times in a secondary region hundreds of miles away from the primary region, providing the highest level

of durability. In the event of a failure at the primary region, Azure Storage will failover to the secondary region. GRS ensures that your data is durable in two separate regions.

- **Read access geo-redundant storage (RA-GRS)**. Read access geo-redundant storage replicates your data to a secondary geographic location, and also provides read access to your data in the secondary location. Read-access geo-redundant storage allows you to access your data from either the primary or the secondary location, in the event that one location becomes unavailable. RA-GRS is also used in scenarios where reporting and other read-only functions can easily be distributed to the replica instead of the primary therefore spreading application load across multiple instances.

> **Note:** Geographically distributed replicas receive any replication asynchronously. This means that your replica is eventually consistent and could possibly have older data if you access the replica before the replication operation from the primary is complete.

Architecture

The **Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency** whitepaper that was released at the 2011 Association for Computing Machinery (ACM) Symposium on Operating Systems Principles.

> **Reference Links:** http://sigops.org/sosp/sosp11/current/2011-Cascais/printable/11-calder.pdf

Azure Storage Tables

The Azure Table storage service stores large amounts of structured data. The service is a NoSQL datastore which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data. Common uses of the Table service include:

- Storing TBs of structured data capable of serving web scale applications
- Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
- Quickly querying data using a clustered index
- Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries

You can use the Table service to store and query huge sets of structured, non-relational data, and your tables will scale as demand increases.

Storage Table Components

The Table service contains the following components:

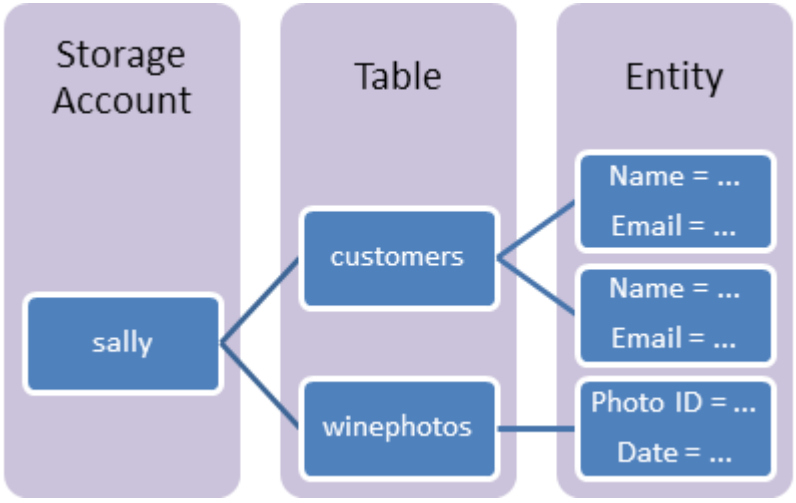


FIGURE 7.1: STORAGE TABLE COMPONENTS

- **URL format**: Code addresses tables in an account using this address format:

  http://<storage account>.table.core.windows.net/<table>

  You can address Azure tables directly using this address with the OData protocol.

- **Storage Account**: All access to Azure Storage is done through a storage account.

- **Table**: A table is a collection of entities. Tables don't enforce a schema on entities, which means a single table can contain entities that have different sets of properties. The number of tables that a storage account can contain is limited only by the storage account capacity limit.

- **Entity**: An entity is a set of properties, similar to a database row. An entity can be up to 1MB in size.

- **Properties**: A property is a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has 3 system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition.

Table Partitioning

Partitions represent a collection of entities with the same PartitionKey values. Partitions are always served from one partition server and each partition server can serve one or more partitions. A partition server has a rate limit of the number of entities it can serve from one partition over time. Specifically, a partition has a scalability target of 500 entities per second. This throughput may be higher during minimal load on the storage node, but it will be throttled down when the node becomes hot or very active.

To better illustrate the concept of partitioning, the following figure illustrates a table that contains a small subset of data for users. It presents a conceptual view of partitioning where the PartitionKey contains different values:

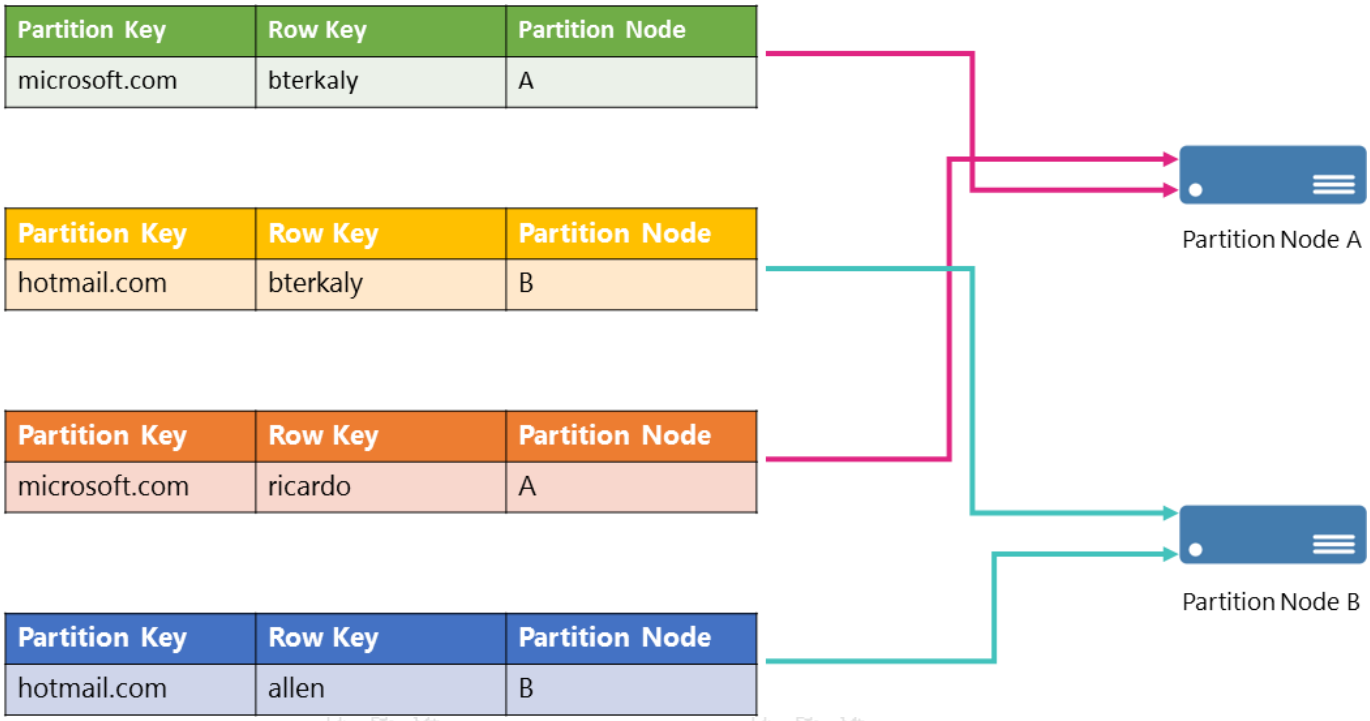


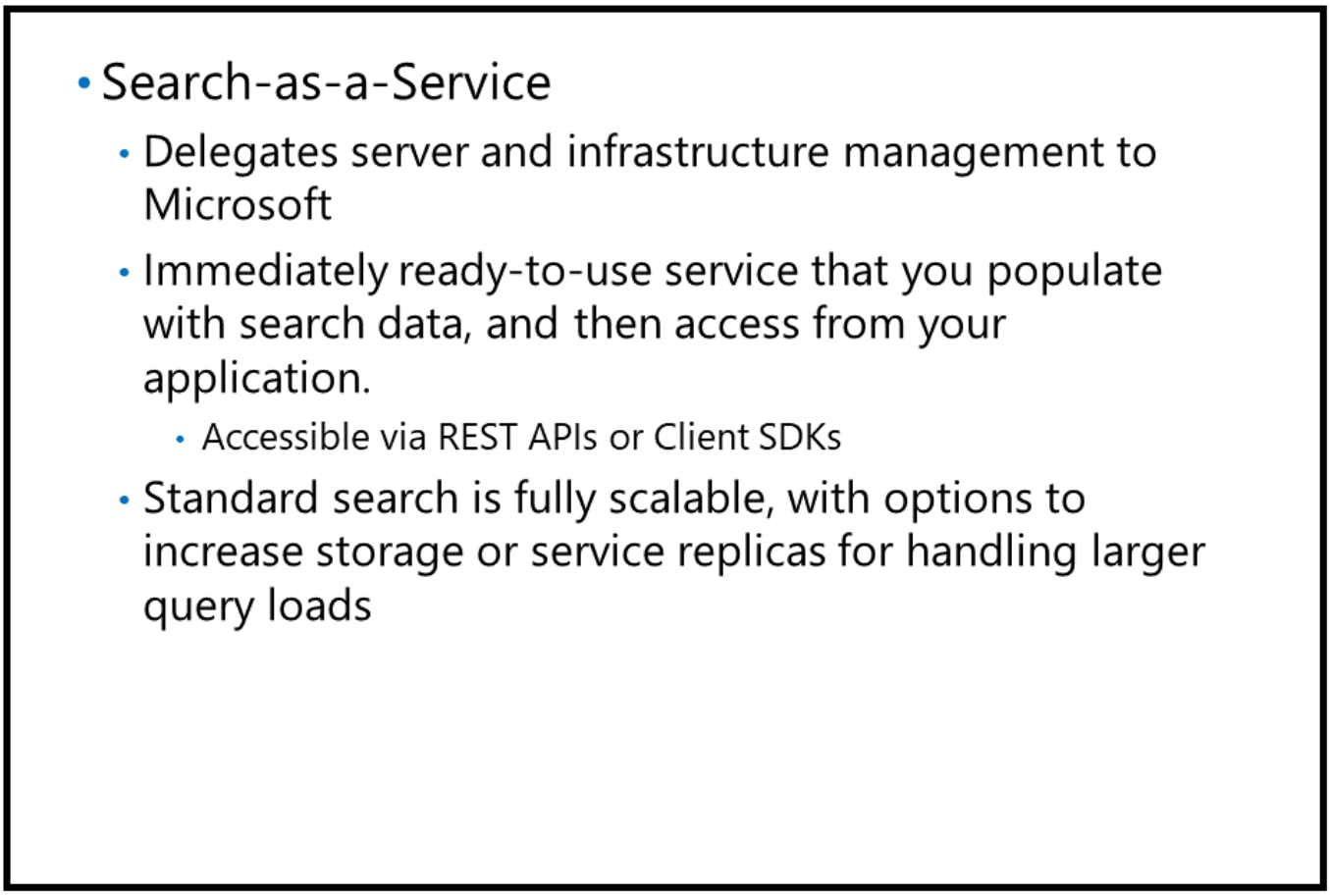
FIGURE 7.2: STORAGE TABLE PARTITIONING

The primary key for an Azure entity consists of the combined PartitionKey and RowKey properties, forming a single clustered index within the table. The PartitionKey and RowKey properties can store up to 1 KB of string values. Empty strings are also permitted; however, null values are not. The clustered index sorts by the PartitionKey in ascending order and then by RowKey in ascending order. The sort order is observed in all query responses.

Because a partition is always served from a single partition server and each partition server can serve one or more partitions, the efficiency of serving entities is correlated with the health of the server. Servers that encounter high traffic for their partitions may not be able to sustain a high throughput. For example, if there are many requests for Partition B, server B may become too hot. To increase the throughput of the server, the storage system load-balances the partitions to other servers. The result is that the traffic is distributed across many other servers. For optimal load balancing of traffic, you should use more partitions, so that the Azure Table service can distribute the partitions to more partition servers.

The PartitionKey values you choose will dictate how a table will be partitioned and the type of queries that can be used. Storage operations, in particular inserts, can also affect your choice of PartitionKey values. The PartitionKey values can range from single values to unique values and also can be composed from multiple values. Entity properties can be composed to form the PartitionKey value. Additionally, the application can compute the value.

Azure Search



Azure Search is a fully managed cloud service that allows developers to build rich search applications using a .NET SDK or REST APIs. It includes full-text search scoped over your content, plus advanced search behaviors similar to those found in commercial web search engines, such as type-ahead query suggestions based on a partial term input, hit-highlighting, and faceted navigation. Natural language support is built-in, using the linguistic rules that are appropriate to the specified language.

You can scale your service based on the need for increased search or storage capacity. For example, retailers can increase capacity to meet the extra volumes associated with holiday shopping or promotional events. Azure Search is also an API-based service for developers and system integrators who know how to work with web services and HTTP. You can use existing platforms and frameworks since search only requires HTTP requests.

Azure Search is a PaaS service that delegates server and infrastructure management to Microsoft, leaving you with a ready-to-use service that you populate with search data, and then access from your application. Depending on how you configure the service, you'll use either the free service that is shared with other Azure Search subscribers, or the Standard pricing tier that offers dedicated resources used only by your service. Standard search is scalable, with options to meet increased demands for storage or query loads. Azure Search stores your data in an index that can be searched through full text queries. The schema of these indexes can either be created in the Azure Portal, or programmatically using the client library or REST APIs. The schema can also be auto-generated from an existing data source such as SQL Database or Document DB. Once the schema is defined, you can then upload your data to the Azure Search service where it is subsequently indexed.

Lesson 3: Azure Cosmos DB

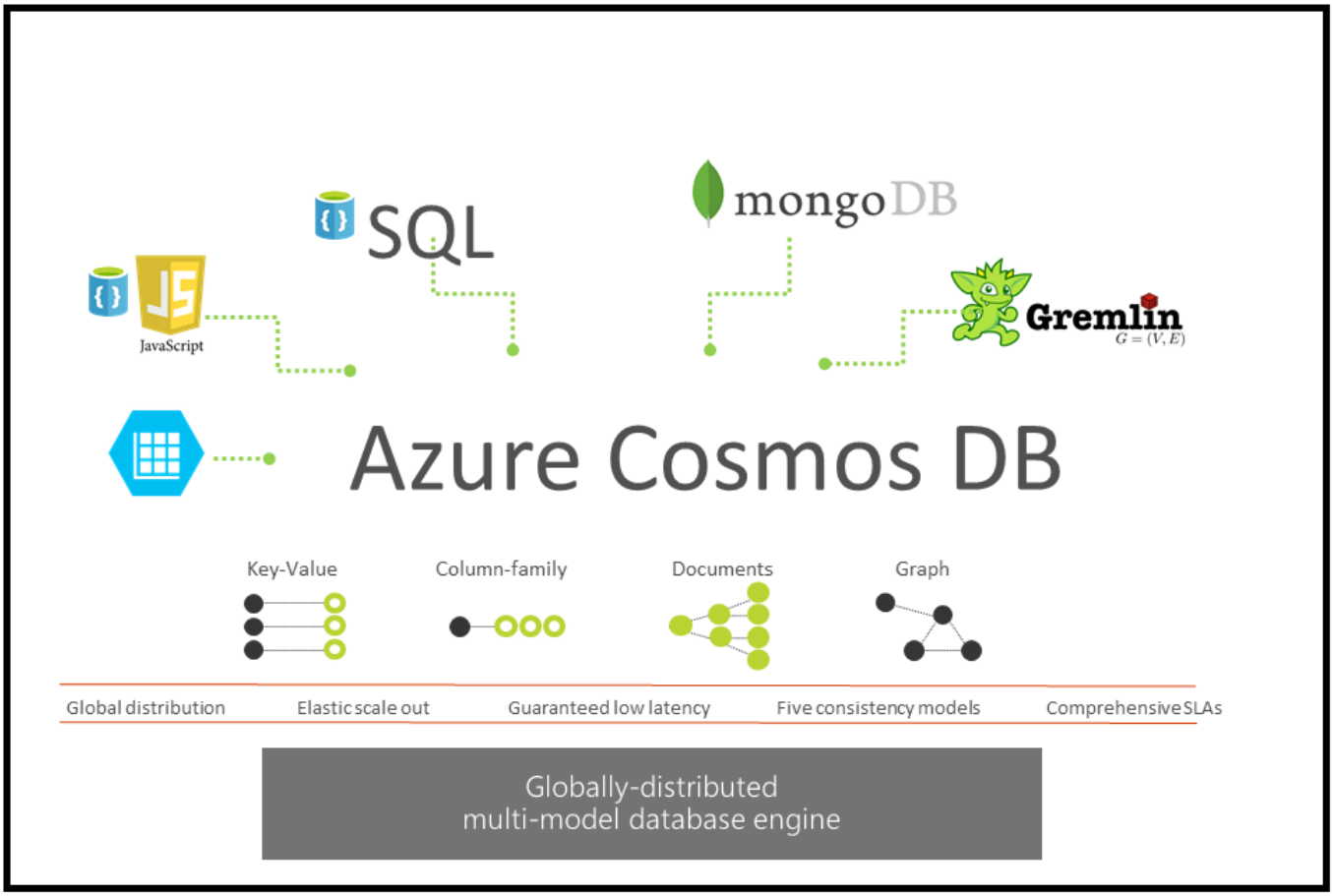
This lesson briefly introduces the Azure Cosmos DB NoSQL database service.

Lesson objectives

After completing this lesson, you will be able to:

- Describe the general features available in the Azure Cosmos DB service.
- List the specific APIs and models available for Azure Cosmos DB client applications.

Cosmos DB



Azure Cosmos DB is Microsoft's new globally distributed, multi-model database service. Azure Cosmos DB offers a turn-key database service that allows you to create a database and distribute the data globally so that the data users access or in datacenters closer to them.

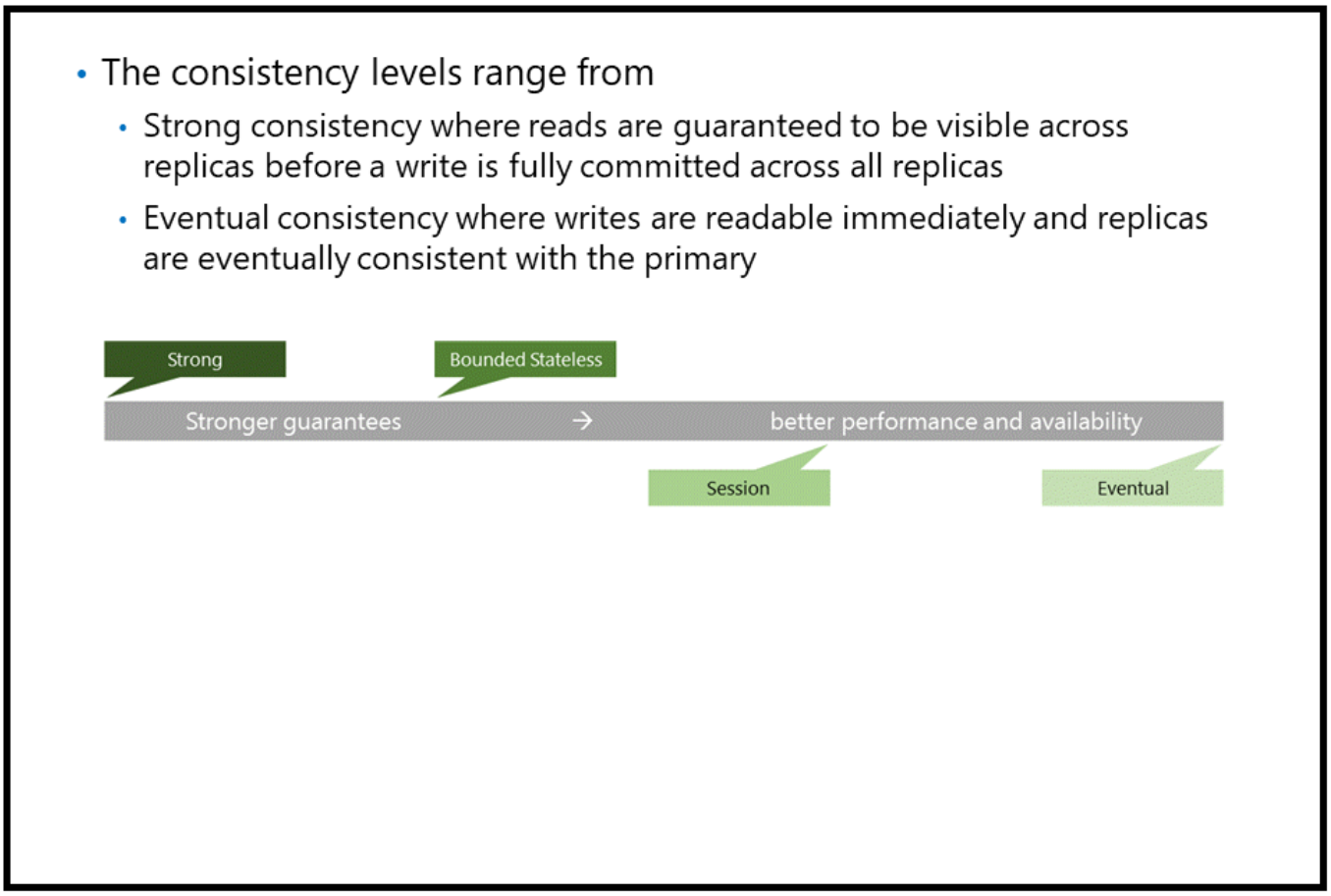
Azure Cosmos DB APIs

Today, Azure Cosmos DB can be accessed using four different APIs:

- **DocumentDB (SQL) API**
- **MongoDB API**
- **Graph (Gremlin) API**
- **Tables (Key/Value) API**

Over time, Azure Cosmos DB will be expanded to offer new APIs and data models that are relevant to the latest distributed applications.

Consistency Levels



When you create a set of data in Azure Cosmos DB, your data is transparently replicated to ensure high availability. To accomplish this, the Azure Cosmos DB service automatically creates partitions, behind the scenes, and distribute your data across these partitions.

In Azure Cosmos DB, a partition is a fixed amount of high-performance storage that contains your data. When your data grows beyond the capacity of a partition, the Azure Cosmos DB service automatically determines the quantity of partitions needed and how to distribute the data across those partitions.

Additionally, you can specify a partition key to influence how your data is distributed. A partition key is a JSON path or property that is used by DocumentDB to ensure that related documents are stored in the same parition. This means that documents with the same partition key would be stored within the same partition. This also means that queries within a single partition perform better than queries that cross multiple partitions.

Consistency Levels

DocumentDB allows you to specify one of four potential consistency levels per account. A consistency level specified at the database-level is applied automatically to all databases and collections within your account.

The consistency levels range from very strong consistency where reads are guaranteed to be visible across replicas before a write is fully committed across all replicas to a eventual consistency where writes are readable immediately and replicas are eventually consistent with the primary.

CONSISTENCY LEVEL	DESCRIPTION
STRONG	When a write operation is performed on your primary database, the write operation is replicated to the replica instances. The write operation is only committed (and visible) on the primary after it has been committed and confirmed by ALL replicas.

CONSISTENCY LEVEL	DESCRIPTION
BOUNDED STATELESS	This level is similar to the Strong level with the major difference is that you can configure how stale documents can be within replicas. Staleness refers to the quantity of time (or version count) a replica document can be behind the primary document.
SESSION	This level guarantees that all read and write operations are consistent within a user session. Within the user session, all reads and writes are monotonic and guaranteed to be consistent across primary and replica instances.
EVENTUAL	This level is the loosest consistency and essentially commits any write operation against the primary immediately. Replica transactions are asynchronously handle and will eventually (over time) be consistent with the primary. This tier is the most performant as the primary database does not need to wait for replicas to commit to finalize it's transactions.

Picking a Consistency Strategy

There are two main things to consider when thinking about your consistency level. First a consistency level on the strong side of the list will ensure that your versions of documents in your replica do not lag behind the primary. If your application requires all replica documents to exactly match the primary at any point in time, this strategy makes a lot of sense. The downside is that the primary write operation will be a lot slower than usual because that operation must wait for every replica to confirm that the operation has been committed.

A consistency level on the eventual (loose) side will ensure that your database operates at peak efficiency. This occurs because operations against the primary database commit immediately and do not wait for the replicas to confirm that they are committed. This is useful for scenarios where you need the highest tier of performance. The downside here is that there is a potential for any read operations against a replica to be a couple of versions behind the primary and return inconsistent data.

Lesson 4: Data Storage & Integration

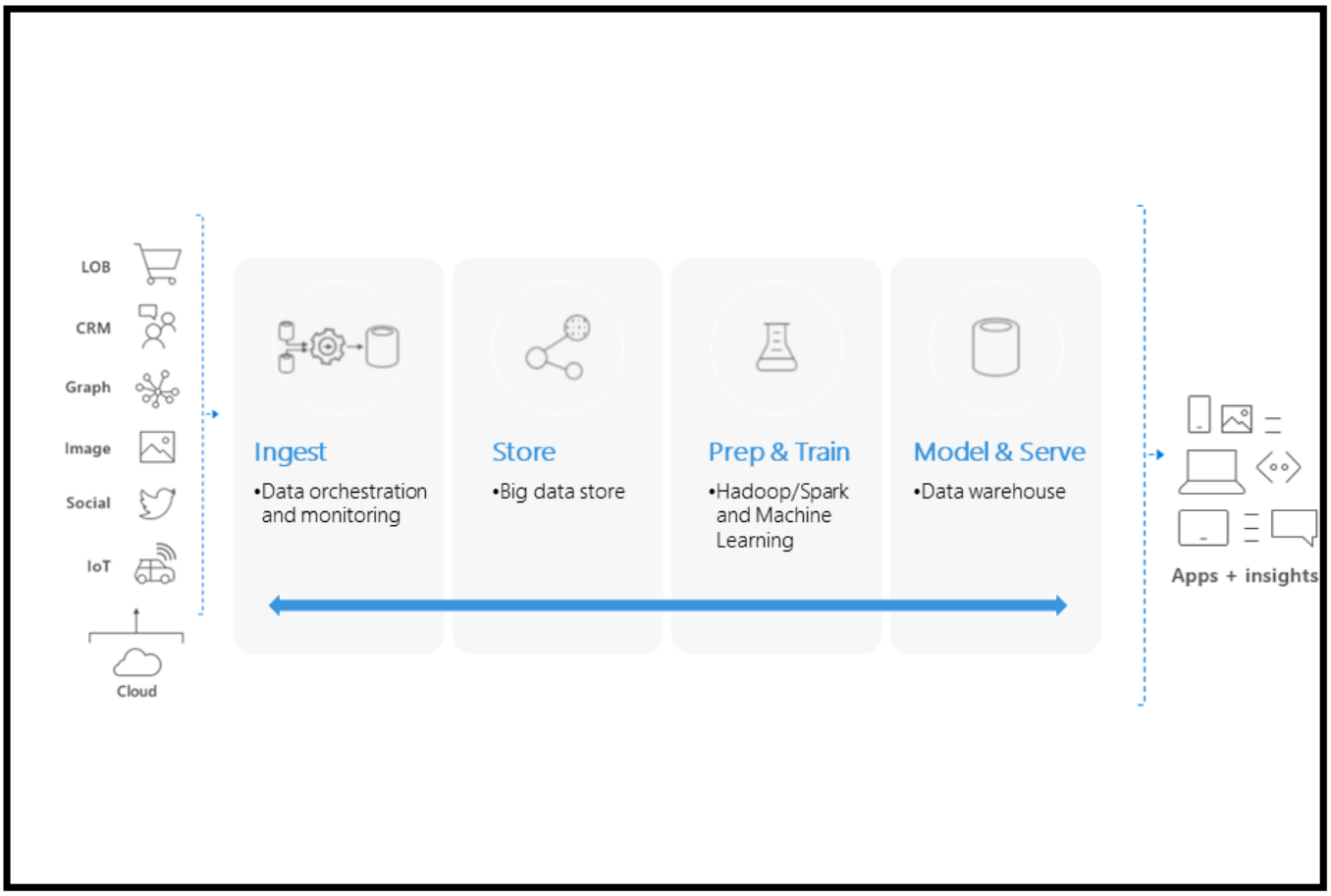
This lesson is a brief overview of various Data Storage & Data Integration options available on Azure.

Lesson objectives

After completing this lesson, you will be able to:

- Describe the SQL Data Warehouse and Data Lake services.
- Describe the Data Factory service.
- Decide when it is appropriate to use SQL Data Warehouse, Data Lake or Data Factory.
- Incorporate Data Factory into a design that uses either SQL Data Warehouse or Data Factory.

Data Storage & Integration Options



SQL Data Warehouse

SQL Data Warehouse is a cloud-based Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data. Use SQL Data Warehouse as a key component of a big data solution. Import big data into SQL Data Warehouse with simple PolyBase T-SQL queries, and then use the power of MPP to run high-performance analytics. As you integrate and analyze, the data warehouse will become the single version of truth your business can count on for insights.

In a cloud data solution, data is ingested into big data stores from a variety of sources. Once in a big data store, Hadoop, Spark, and machine learning algorithms prepare and train the data. When the data is ready for complex analysis, SQL Data Warehouse uses PolyBase to query the big data stores. PolyBase uses standard T-SQL queries to bring the data into SQL Data Warehouse.

SQL Data Warehouse stores data into relational tables with columnar storage. This format significantly reduces the data storage costs, and improves query performance. Once data is stored in SQL Data Warehouse, you can run analytics at massive scale. Compared to traditional database systems, analysis queries finish in seconds instead of minutes, or hours instead of days.

The analysis results can go to worldwide reporting databases or applications. Business analysts can then gain insights to make well-informed business decisions.

Azure Data Lake

Azure Data Lake Store is an enterprise-wide hyper-scale repository for big data analytic workloads. Azure Data Lake enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics.

Azure Data Lake Store can be accessed from Hadoop (available with HDInsight cluster) using the WebHDFS-compatible REST APIs. It is specifically designed to enable analytics on the stored data and is tuned for performance for data analytics scenarios. Out of the box, it includes all the enterprise-grade capabilities—security, manageability, scalability, reliability, and availability—essential for real-world enterprise use cases.

Some of the key capabilities of the Azure Data Lake include the following.

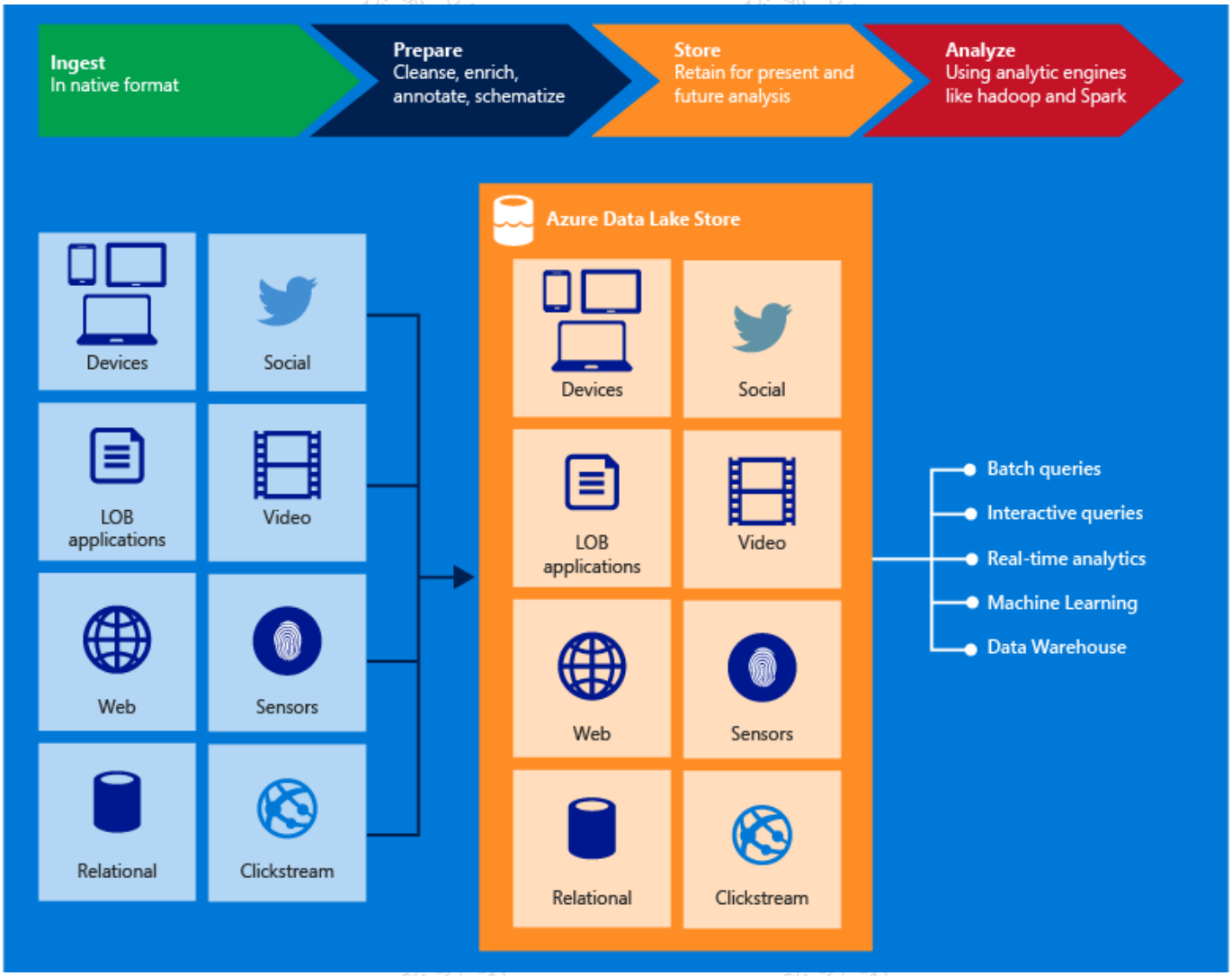
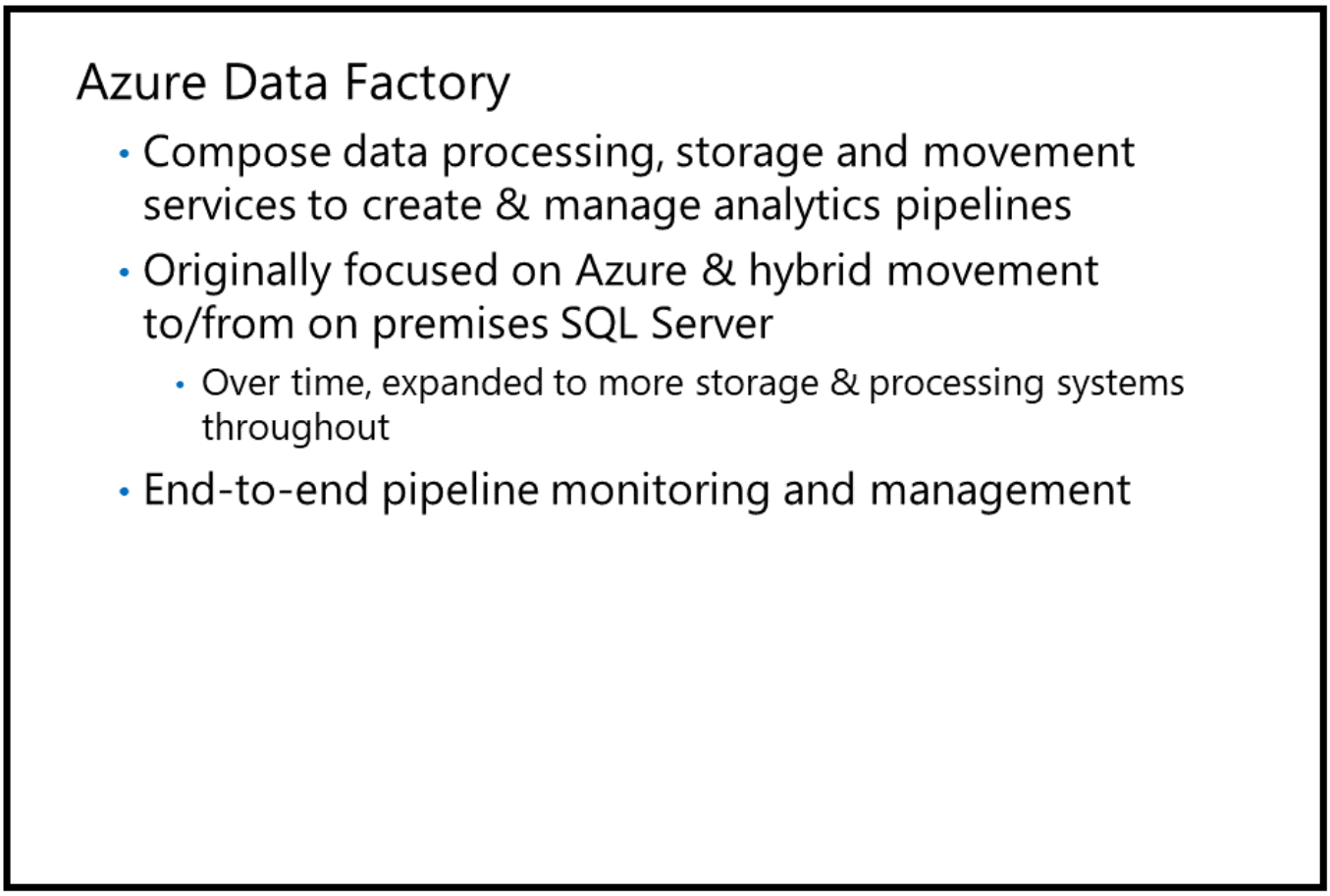


FIGURE 7.3: KEY DATA LAKE FEATURES

Azure Data Lake Analytics is an on-demand analytics job service to simplify big data analytics. You can focus on writing, running, and managing jobs rather than on operating distributed infrastructure. Instead of deploying, configuring, and tuning hardware, you write queries to transform your data and extract valuable insights. The analytics service can handle jobs of any scale instantly by setting the dial for how much power you need. You only pay for your job when it is running, making it cost-effective. The analytics service supports Azure Active Directory letting you manage access and roles, integrated with your on-premises identity system. It also includes U-SQL, a language that unifies the benefits of SQL with the expressive power of user code. U-SQL's scalable distributed runtime enables you to efficiently analyze data in the store and across SQL Servers in Azure, Azure SQL Database, and Azure SQL Data Warehouse.

Data Integration



Azure Data Factory

Azure Data Factory is a platform created to help refine big data, enormous stores of raw data, into actionable business insights. It is a managed cloud service that's built to handle extract-transfer-load, data integration, and hybrid extract-transfer-load projects. You can use it to schedule and create data-driven workflows, other words known as pipelines, that can ingest data from data stores. Azure Data Factory operates in conjuncture with services such as Azure Data Lake Analysts, Azure Machine learning, and Spark to process and transform data.

Azure Data Factory is the service that can help with scenarios of this kind. Azure Data Factory is a cloud-based data integration service. It allows users to create data-driven workflows in the cloud, as well as methods to automate data movement and transformation. You can also issue output data to data stores to consume. Though Azure Data Factory, raw data can be organized into meaningful data lakes to assist in business decisions.

To give an example of the uses of Azure Data Factory imagine a social media company that collects petabytes of member information and logs that are produced by members in the cloud. The company would want to examine to gain insights on member preference, usage behavior, and demographics. The company would also want to discover opportunities to cross-sell or up-sell, generate new features, provide an excellent experience for the members, and drive business growth.

To examine this information the company needs to make use of various reference data. This data includes the member's information, marketing campaign information, and information on the social media platform which is all stored in an on-premises data store. The company could apply data from an additional log data from a cloud store, combining it with the on-premises data store.

When using Azure Data Factory, your data-driven workflows (better known as pipelines) typically performs the following four steps: Connect & Collect, Transform & Enrich, Publish, and Monitor. Data Factory connects to all required sources such as file shares, FTP web, databases, and software-as-a-service (SaaS) services. From here you can use Copy Activity within the data pipeline to move data to a centralized data store in the cloud for analysis.

After the raw data has been collected and moved to the centralized data store, Azure Data Factory processes the collected data using compute services such as Spark, Data Lake Analytics, and Machine Learning. The refined data, now in a business-ready consumable form, is loaded to an analytics engine such as Azure SQL database to be published to be used with your business intelligence tools. After building and deploying the data integration pipeline, you can monitor the scheduled activities and pipelines for failure and success rates.

Lesson 5: Data Analysis

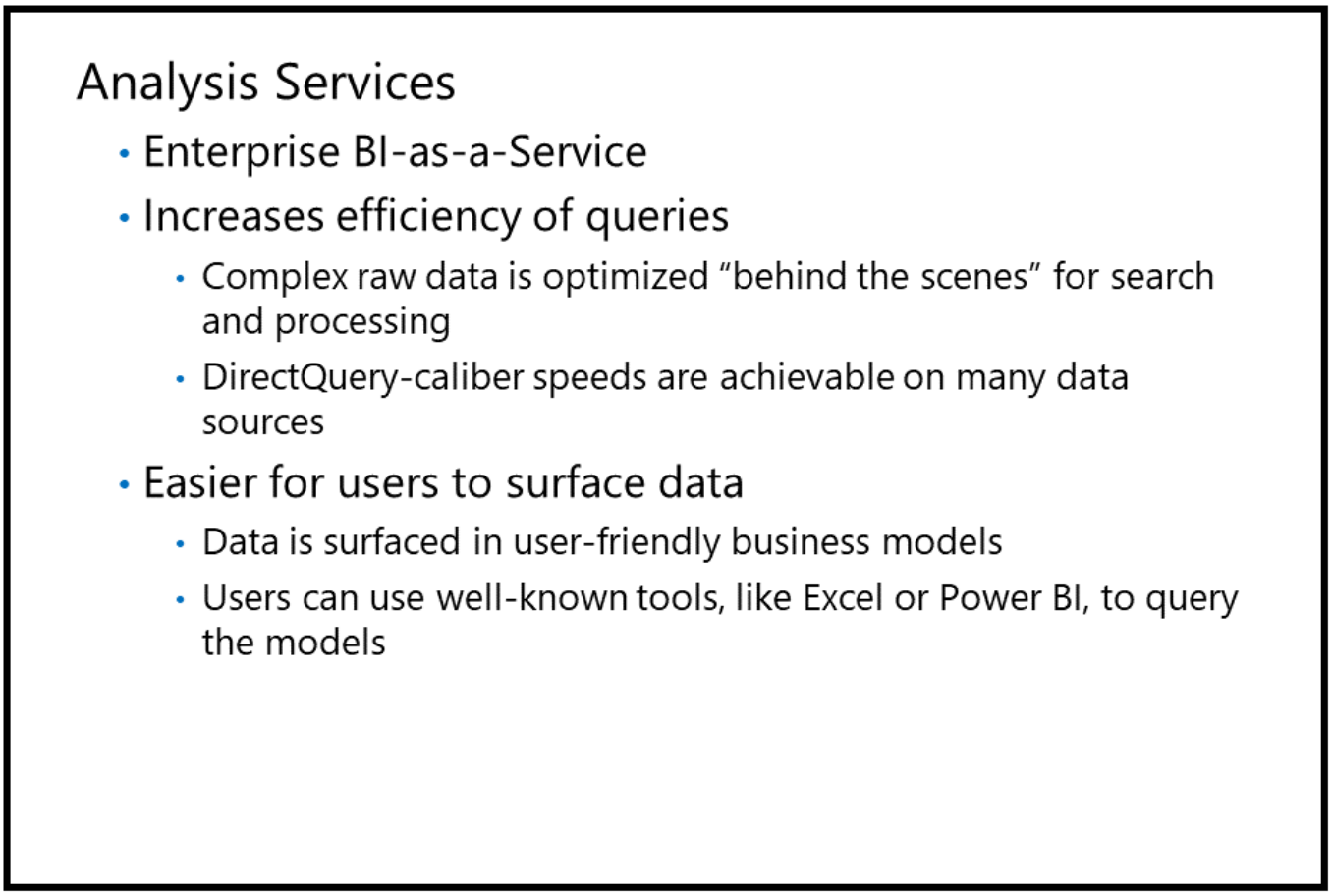
This lesson is a brief overview of various Data Analysis services available on Azure.

Lesson objectives

After completing this lesson, you will be able to:

- Design a solution that uses Azure Analysis Services for end-user ad-hoc data set queries.
- Design a solution that surfaces “tribal knowledge” using Azure Data Catalog.
- Detail how Azure HDInsight can be integrated into solutions using various Azure Data storage, integration or analysis services.

Data Analysis Options



Azure Analysis Services

Azure Analysis Services provides enterprise-grade data modeling in the cloud. It is a fully managed platform as a service (PaaS), integrated with Azure data platform services.

With Analysis Services, you can mashup and combine data from multiple sources, define metrics, and secure your data in a single, trusted semantic data model. The data model provides an easier and faster way for your users to browse massive amounts of data with client applications like Power BI, Excel, Reporting Services, third-party, and custom apps.

Azure Analysis Services is compatible with many features already in SQL Server Analysis Services Enterprise Edition. Azure Analysis Services supports tabular models at the 1200 and 1400 compatibility levels. Partitions, row-level security, bi-directional relationships, and translations are all supported. In-memory and DirectQuery modes are also available for fast queries over massive and complex datasets.

For developers, tabular models include the Tabular Object Model (TOM) to describe model objects. TOM is exposed in JSON through the Tabular Model Scripting Language (TMSL) and the AMO data definition language.

HDInsight

Apache Hadoop was the original open-source framework for distributed processing and analysis of big data sets on clusters. The Hadoop technology stack includes related software and utilities, including Apache Hive, HBase, Spark, Kafka, and many others. To see available Hadoop technology stack components on HDInsight, see Components and versions available with HDInsight. To read more about Hadoop in HDInsight, see the Azure features page for HDInsight.

Apache Spark is an open-source parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications.

Azure HDInsight is the Azure distribution of the Hadoop components from the Hortonworks Data Platform (HDP). Azure HDInsight makes it easy, fast, and cost-effective to process massive amounts of data. You can use the most popular open-source frameworks such as Hadoop, Spark, Hive, LLAP, Kafka, Storm, R, and more to enable a broad range of scenarios such as extract, transform, and load (ETL); data warehousing; machine learning; and IoT.

Azure Data Catalog

Azure Data Catalog is a fully managed cloud service whose users can discover the data sources they need and understand the data sources they find. At the same time, Data Catalog helps organizations get more value from their existing investments.

With Data Catalog, any user (analyst, data scientist, or developer) can discover, understand, and consume data sources. Data Catalog includes a crowdsourcing model of metadata and annotations. It is a single, central place for all of an organization's users to contribute their knowledge and build a community and culture of data.

Data Catalog provides a cloud-based service into which a data source can be registered. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data-source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

After a data source has been registered, its metadata can then be enriched, either by the user who registered it or by other users in the enterprise. Any user can annotate a data source by providing descriptions, tags, or other metadata, such as documentation and processes for requesting data source access. This descriptive metadata supplements the structural metadata (such as column names and data types) that's registered from the data source.

Discovering and understanding data sources and their use is the primary purpose of registering the sources. Enterprise users might need data for business intelligence, application development, data science, or any other task where the right data is required. They can use the Data Catalog discovery experience to quickly find data that

matches their needs, understand the data to evaluate its fitness for the purpose, and consume the data by opening the data source in their tool of choice.

At the same time, users can contribute to the catalog by tagging, documenting, and annotating data sources that have already been registered. They can also register new data sources, which can then be discovered, understood, and consumed by the community of catalog users.

Lesson 6: Web Apps & SQL Case Study

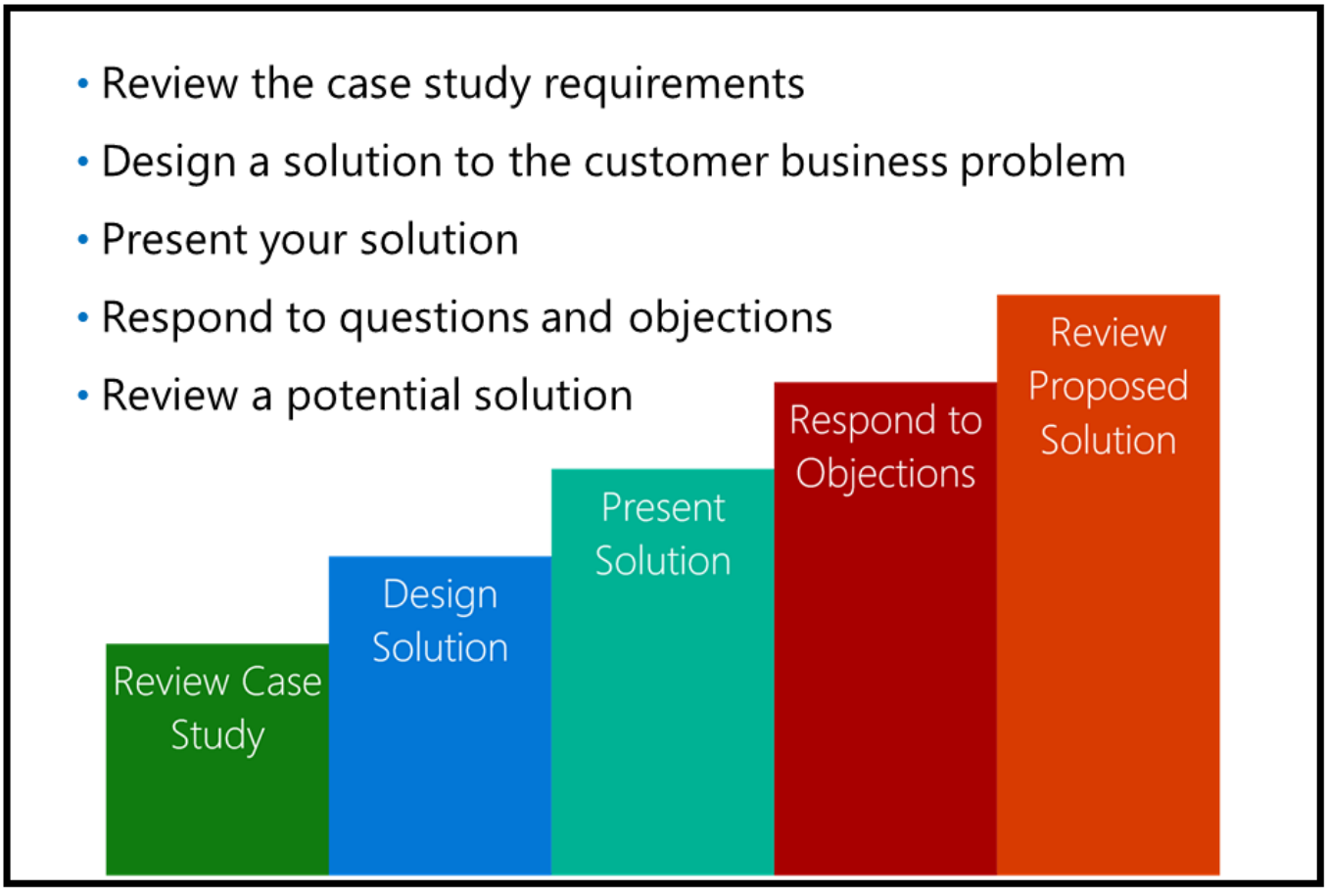
In this case study, we will look at a customer problem that requires an architectural recommendation.

Lesson objectives

After this case study, you should:

- Identify customer problems as they are related to networking.
- Design a solution that will meet the customer's objectives.
- Ensure your designed solution accounts for customer objections.

Case Study Overview



Who is the Customer?

Founded in 1954, Adventure Works Cycles has grown from a boutique manufacturer of high-quality bicycles and parts into one of the world's largest makers of premium race and commuter bicycles. The Adventure Works mission

has remained the same: to passionately pursue advanced, innovative technologies that help cyclists of all abilities find more enjoyment in the sport. Adventure Works Cycles Company manufactures and sells bicycles, bicycle parts, and bicycle accessories under the Adventure Works Cycles and Tailspin brands worldwide.

During the global recession of 2009, most of the company's IT infrastructure was located in the company's Provo, Utah, headquarters, but Adventure Works also had a sizable third-party colocation datacenter, costing US$30,000 to $40,000 a month, and other servers scattered around the United States.

Adventure Works knows that its datacenters are filled with dozens of smaller web servers and databases that run on underutilized hardware, and it has customer data scattered in multiple places. Faced with the prospect of a very large capital expenditure owing to the fact that the vast majority of their servers are now due for a hardware refresh, Adventure Works is looking for other options that eliminate the costly and high risk hardware refresh cycles.

What Does the Customer Already Have?

In reviewing all of Adventure Works' web applications, it was determined that there were three archetypical database backed web applications present:

- **Product Catalog:** Resellers and consumers in North America, Asia and Europe access the product catalog via a website, the data for which is currently stored in SQL Server. Currently, both web app and database are hosted in its Utah datacenter. The database is 50GB in size, and is not expected to grow past 100GB.
- **Inventory:** The inventory web application is a mission critical system primarily used by operations running in North America. Currently, both web app and SQL Server database are hosted in its Utah datacenter. This is Adventure Works' largest database at 3 TB in size, but it is not expected to grow beyond 5 TB.
- **Departmental:** These web applications support the regional offices only. Both web app and database are hosted in the Utah data center. While individually these have web applications have low demands and data sizes in the 500MB-1GB range, Adventure Works has 100's of SQL Server databases supporting the numerous web apps, and fully expects new databases to be created to support departmental efforts.

Most of Adventure Works developers are already trained in Microsoft tools as all of Adventure Works' web applications and services are built using .NET. Additionally, they have already deployed Active Directory and are currently using a replicated directory in support of departments within each regional office.

What is the Customer's Goal?

Not only are all the servers expensive to acquire and maintain, but scaling the infrastructure takes significant time. "During and after a high-profile race, there's a great deal of interest in our product, and our web apps receive a lot of hits," says Hayley Leigh, Manager of Solution Development for Adventure Works Cycles. "It is difficult to scale our web hosting environment fast enough, and consumers and resellers could experience slow response times and even downtime." Another challenge: Adventure Works conducts weekly server hardware maintenance, which causes downtime for some of its global offices.

Adventure Works wants to move many consumer-facing web apps, enterprise databases, and enterprise web services to Azure. "By using Microsoft global datacenters, we're able to move infrastructure for key applications and

web apps closer to the people who use them," Leigh says. A big problem for Adventure Works is resellers and consumers in Japan and China have to use applications that run in a Utah datacenter, and because of the distance, encounter performance problems. Adventure Works would like to resolve this without the difficulty, expense and time requirements incurred by setting up infrastructure on the other side of the world using Adventure Works-owned servers.

What Does the Customer Need?

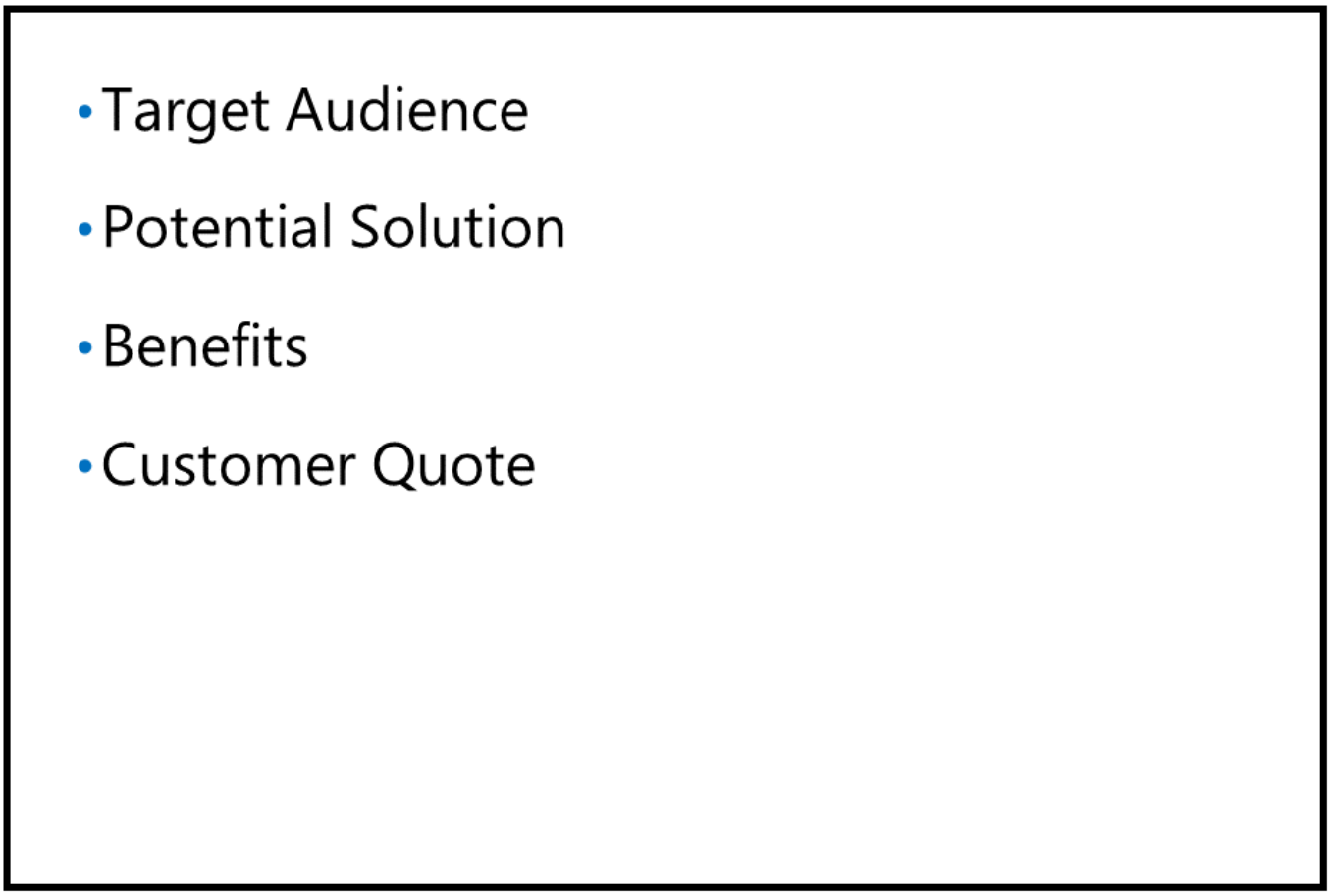
- In reviewing all of Adventure Works' web applications, it was determined that there were three archetypical database backed web applications present:
- Product Catalog: Resellers and consumers in North America, Asia and Europe access the product catalog via a website, the data for which is currently stored in SQL Server. Currently, both web app and database are hosted in its Utah datacenter. The database is 50GB in size, and is not expected to grow past 100GB.
- Inventory: The inventory web application is a mission critical system primarily used by operations running in North America. Currently, both web app and SQL Server database are hosted in its Utah datacenter. This is Adventure Works' largest database at 3 TB in size, but it is not expected to grow beyond 5 TB.
- Departmental: These web applications support the regional offices only. Both web app and database are hosted in the Utah data center. While individually these have web applications have low demands and data sizes in the 500MB-1GB range, Adventure Works has 100's of SQL Server databases supporting the numerous web apps, and fully expects new databases to be created to support departmental efforts.
- Most of Adventure Works developers are already trained in Microsoft tools as all of Adventure Works' web applications and services are built using .NET. Additionally, they have already deployed Active Directory and are currently using a replicated directory in support of departments within each regional office.

What Things Worry the Customer?

- **Scale & Performance**
  - I do not want to have to make code changes (or re-deploy) in order to change the scale of a website.
  - I hear Azure Web Apps is only useful for web apps with small amounts of traffic; will it really support the heavy traffic we receive?
  - We would prefer to avoid performing a database migration (e.g., to another server) in order to scale the throughput of our database.
  - We have heard SQL Database does not provide consistent performance, is this true?

- **Business Continuity**
  - How can we certain our data will survive in the event of a catastrophe in a certain part of the world?
  - We need to be able to recover from mistakes made by administrators that accidentally delete production data (we know they happen, we would love an “undo”).
  - Do we need to have multiple web server instances for each property to have a high SLA?
- **Tool Familiarity**
  - Will we need to learn new tools to develop for Azure Web Apps and SQL Database?
  - What about diagnosing problems? Are there new tools we need purchase and learn?
- **Connectivity**
  - Some of our enterprise web services need to access data and other services located on-premises, is this supported?
  - How can we ensure we are delivering the lowest latency possible to our website visitors?
  - We need to ensure that if we have multiple web servers backing a given website, that no one web server gets all the traffic.
- **Management**
  - We would prefer not to have to manage patching of web servers and databases.
  - With all of our web apps and databases around the world, how do we keep tabs on which is up and which is down and which is struggling?
  - We need a simple solution to schedule and automate backup of the website and database.
- **Security**
  - Is it possible to allow our visitors to use a mix of legacy and modern browsers and still provide for secure transactions?
  - What does Azure offer to help us with auditing access to our web servers and databases?
  - Our staff is accustomed to accustomed to a single sign-on experience — will this still be possible?

Case Study Solution



Preferred Target Audience

Hayley Leigh, Manager of Solution Development for Adventure Works Cycles

The primary audience is the business decision makers and technology decision makers. From the case study scenario, this would include the Manager of Solution Development. Usually we talk to the Infrastructure Managers who report into the CIO's, or to application sponsors (like a VP LOB, CMO) or to those that represent the Business Unit IT or developers that report into application sponsors.

Preferred Solution

Adventure Works Cycles decided that cloud computing could solve just about all these problems. Moving some workloads into a cloud environment—where virtualized compute, storage, and network resources run in public datacenters, are shared by multiple parties, and are delivered over the Internet—could reduce costs, improve scalability and business agility, and enable the desktop management team to manage remote computers.

Cowles's team evaluated cloud offerings from multiple providers and selected Microsoft Azure, the Microsoft cloud platform that provides on-demand compute, storage, content delivery, and networking capabilities from Microsoft datacenters. "The other companies offered infrastructure-as-a-service but not software-as-a-service or platform-as-a-service as Microsoft did," Cowles says. "We wanted the whole spectrum of cloud options to fit a range of cloud needs. Also, most of our developers are trained in Microsoft tools, so it was much easier for us to connect our in-house systems to Microsoft Azure."

Adventure Works then moved many consumer-facing web apps, enterprise databases, and enterprise web services to Azure. "By using Microsoft global datacenters, we're able to move infrastructure for key applications and web apps closer to the people who use them," Cowles says. For example, resellers and consumers in Japan and China previously had to use applications that ran in a Utah datacenter, and because of the distance, encountered performance problems. In about an hour, Cowles's team set up the same services in a Microsoft datacenter in Asia and completely eliminated the performance lags. "It would have been extremely difficult, expensive, and time-

consuming to set up this infrastructure on the other side of the world using Adventure Works-owned servers," Cowles says.

Proof of Concept

The PoC scenario would be to address the latency issue faced by resellers around the world that currently rely on to access the Utah datacenter. The PoC would address showing how by using Azure Traffic Manager along with Azure Web Apps in multiple regions, one can ensure that these users are always being routed to the Web app that is "closest" to them in terms of minimizing network latency. The PoC could demonstrate this by provisioning a representative portion of the Product Catalog web application in an Azure Website, along with the storage required for it (e.g., Blob Storage, Table Storage and SQL Database) in the West US region, and then also provisioning the same infrastructure in the Japan West region. Traffic Manager would then be configured to route traffic using the Performance load balancing method.

The following diagram illustrates this:

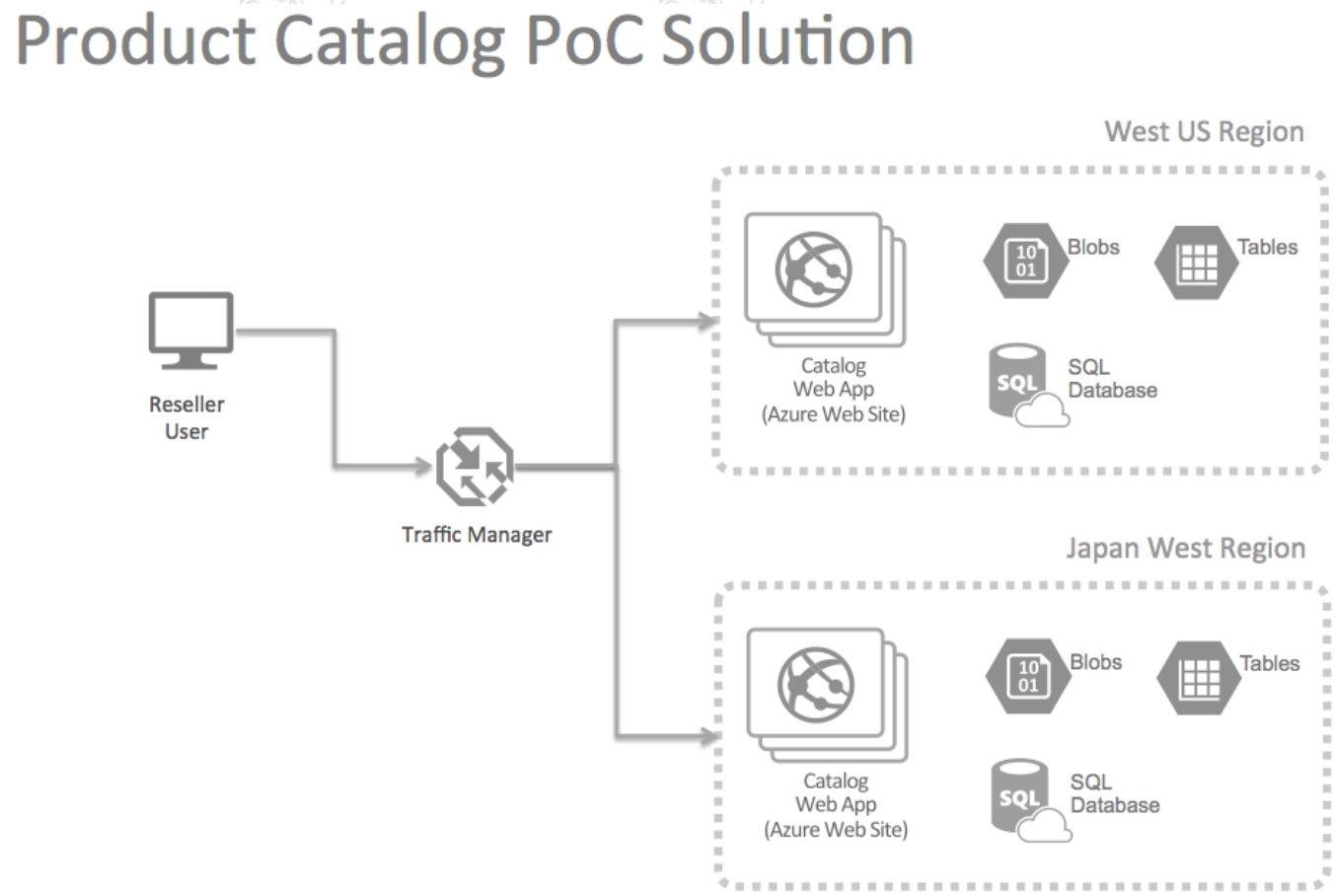


FIGURE 7.4: GEO-DISPERSE WEB APPLICATION

To complete the PoC, either Adventure Works staff or resellers located near each region could be asked to visit the PoC reseller web app, measure their experience (e.g., page load times, time to first paint, etc.) and report on the perceived performance (e.g., does it feel more responsive than the current site?).

On the database side, the PoC should be augmented to use the SQL Database Premium tier. The PoC would address showing using SQL Database Premium along with Azure Web Apps in multiple regions, whereby one configures SQL Database Premium to provide Active Geo-Replication across two regions (West US and Japan West as examples). SQL Database Premium could be configured with the primary in the West US Region, and a readable replica available in the Japan West Region.

The following diagram illustrates this:

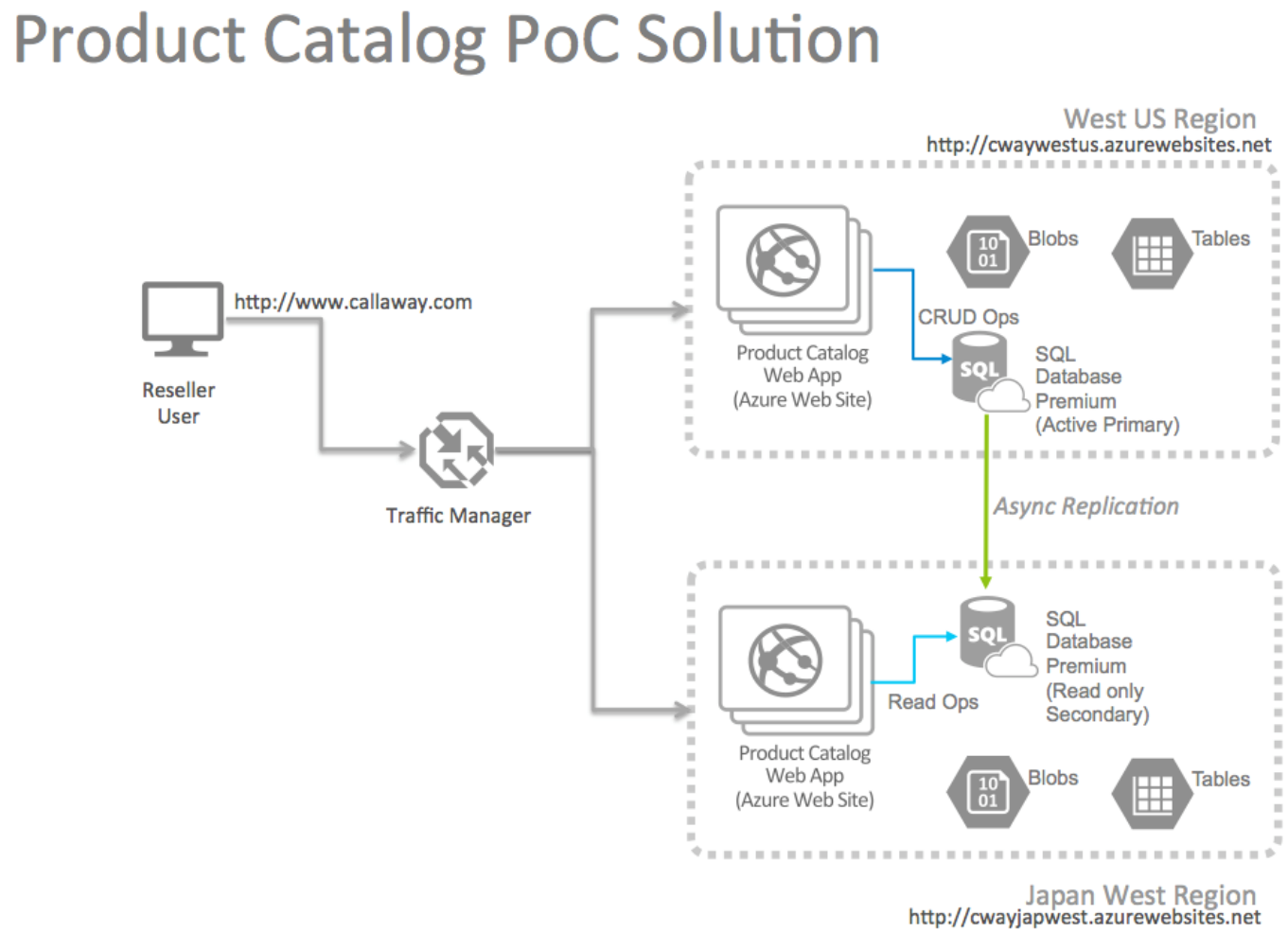


FIGURE 7.5: SYNCING DATABASE SOURCES

By approaching it this way, the PoC could explore the benefits of accessing read-only data from the database nearest the Web app, while reserving data modifications operations for the active primary database instance.

To complete the PoC, Adventure Works could simulate a regional failure and switch to using the SQL Database in the Japan West region as the active primary. They could go one step further and stop the Web Apps in the West US region and thereby simulate a complete regional failure of the West US region. By having Traffic Manager in place, all traffic would flow to the Japan West Region, and the active database would be in that region as well.

Risks & Mitigation

By demonstrating the performance load balancing method, the PoC helps Adventure Works to immediately establish confidence in Azure's global capabilities to deliver a high performance experience to their customers and resellers around the world. By demonstrating the fail over between regions, the PoC helps Adventure Works to immediately establish confidence in Azure's global capabilities to deliver a highly available experience to their customers and resellers around the world, even in the face of a regional failure.

By migrating the Product Catalog web app, it can also help to mitigate the risk the components used by the reseller website are incompatible with Azure Web Apps (for example, if one of the components used by the web app application requires an MSI based install).

This PoC is a good opportunity to evaluate Adventure Works's use of "naked domains" (e.g., a domain with no "www" subdomains such as http://www.adventure-works.com/) for the product catalog, because Azure Traffic Manager cannot directly resolve requests against naked domains. If naked domains are required, Adventure Works would need to configure DNS forwarding with their DNS provider (if their DNS provider supports it) or leverage another DNS forwarding service to redirect naked domain requests to resource having a subdomain (such as http://www.adventure-works.com/).

By migrating a representative portion of the Product Catalog Web App and its data, it can also help to mitigate the risk that SQL Server features used by the database are incompatible with SQL Database (for example, if they are using CLR stored procedures).

Success Criteria

The Product Catalog Web Application (or portion thereof) and supporting data can be successfully migrated to Azure. Users in each region are reporting improved performance with reduced latency. When a failover is simulated, the web app remains functional (albeit users may experience greater latency).

Implementation

Inventory Web App

For the inventory web app, whose database sees large database sizes (3-5TB), only SQL Server in a VM can scale to that capacity. SQL Database is not a good recommendation here for two reasons:

- SQL Database has a maximum of 500GB per database
- To grow beyond this using SQL Database would require sharding, which would require re-designing the web application. One of the customer needs is to avoid such re-architecting of the database structure and the making of large changes to the application.

Departmental Web App

For the departmental web apps, the database sizes were small (500 MB to 1 GB). In choosing between SQL Database and SQL in a VM, one should consider the ease of migrating brownfield apps. The customer stated having 100's of such databases, so moving these to SQL Database could prove overly expensive and time consuming on account of any incompatibilities that would need to be worked thru. Therefore, for the existing departmental apps, these databases should be created with SQL Server in a VM.

For new, greenfield, applications that do not have such incompatibilities, the departmental customer is very likely to appreciate the minimal IT involvement required: to quickly provision a SQL Database for their departmental application without IT support, to benefit from the Point-in-Time Restore should a departmental user make a mistake that results in data loss, and to have high availability within a single data center without the extra configuration that setting up a SQL Server cluster or Availability Group would require. Given the small database sizes, most departmental solutions could leverage the cost efficient SQL Database Basic tier.

Checklist of Preferred Objection Handling

- **Scale & Performance**
  - **I do not want to have to make code changes (or re-deploy) in order to change the scale of a website.**
  - With Azure Web Apps, you do not need to make changes or re-deploy in order to change the scale of a Web app.
  - **I hear Azure Web Apps is only useful for web apps with small amounts of traffic;**

will it really support the heavy traffic we receive?

- Azure Web Apps is capable of supporting Web Apps with loads ranging from small amounts of traffic to large amounts of traffic—this is enabled by its ability to scale up the instance size and to scale out the number of instances in order to meet demand.
- **We would prefer to avoid performing a database migration (e.g., to another server) in order to scale the throughput of our database.**
- You can move between SQL Database service tiers or performance levels using the Azure Portal or Azure PowerShell without having to migrate the database.
- **We have heard SQL Database does not provide consistent performance, is this true?**
- It was true with Web and Business editions, but these are now in the process of being deprecated. The SQL Database Basic, Standard and Premium offerings allow you to purchase a database meeting specific performance criteria.

- **Business Continuity**
  - **How can we certain our data will survive in the event of a catastrophe in a certain part of the world?**
  - By deploying Web Apps to multiple regions and using Azure Traffic Manager to route between them for performance, you will still get the benefit failing over to another web app in another region should all of the web app endpoints in one region become unavailable. It is worth noting that such a failover may not be to the next “closest” region in such scenarios.
  - With Active Geo-Replication, a feature of SQL Database Premium, you can create and maintain up to four readable secondary databases across geographic regions. All transactions applied to the primary database are replicated to each of the secondary databases. The secondary databases can be used for read workloads, database migration, and protection against data loss during application upgrade as a failback option.
  - SQL Database Standard provides Standard Geo-Replication which enables you to have a single offline secondary in a pre-set region that is different from the active primary. This secondary is offline in the sense that, while it is synchronized with the primary, it will refuse any other connections until a failover happens and the datacenter hosting the primary becomes unavailable.
  - Geo-Restore is a feature available to Basic, Standard and Premium SQL Database. It enables you to request a restore of your database using the latest weekly full backup plus differential backup, to any server in any Azure region. These backups are stored in geographically redundant storage.

- **We need to be able to recover from mistakes made by administrators that accidentally delete production data (we know they happen, we would love an "undo").**
- Microsoft Azure SQL Database creates backups of your data, and gives you the ability to recover your data from unwanted deletions or modifications. With Point in Time Restore on SQL Database Premium, you can restore to a database state as far back as 35 days.
- **Do we need to have multiple web server instances for each property to have a high SLA?**
- Unlike other Azure compute services, Web Apps provides a high availability SLA using only a single standard instance.

- **Tool Familiarity**
  - **Will we need to learn new tools to develop for Azure Web Apps?**
  - No. You can use familiar tools like Visual Studio to develop for Azure Web Apps.
  - Visual Studio and SQL Server Management Studio can both be used to manage SQL Database.
  - **What about diagnosing problems? Are there new tools we need purchase and learn?**
  - No. While there are new tool options, such as using the Azure Portal, Server Control Manager (Kudu) or examining logs stored in Blob or Table storage, when it comes to diagnosing problems you can still use familiar tools like Visual Studio.
  - SQL Server Management Studio and Dynamic Management Views can be used to diagnose problems with SQL Database and SQL Server in a VM.
- **Connectivity**
  - **Some of our enterprise web services need to access data and other services located on-premises, is this supported?**
  - Yes. Using the Service Bus Relay, Virtual Networks VPN or ExpressRoute access to data and services located on-premises is made possible.
  - **How can we ensure we are delivering the lowest latency possible to our website visitors?**
  - Use Traffic Manager with the Performance load balancing method and deploy your solution to Web Apps in multiple regions nearest to your web app visitor populations.
  - **We need to ensure that if we have multiple web servers backing a given website, that no one web server gets all the traffic.**

- Within a datacenter, the Azure load balancer automatically handles round-robin load balancing between instances.
- Outside the datacenter, Traffic manager can be used in the Round-Robin load balancing method to route requests across data centers.

- **Management**
  - **We would prefer not to manage patching of web servers and databases.**
  - With Azure Web Apps, patching of the underlying virtual machines is performed automatically and is transparent to you.
  - With SQL Database, patching of the host OS and the database is handled for you.
  - **With all of our web apps and databases around the world, how do we keep tabs on which is up and which is down and which is struggling?**
  - Azure Web Apps provides support for end point monitoring, which enables you to collect responsiveness metrics of a given Web app from multiple endpoints around the world.
  - You can monitor these metrics and also configure the sending of alerts when certain thresholds are exceeded using the Azure Portal.
  - You can acquire an overview of Database health for each of your SQL Databases from the Azure Portal.
  - **We need a simple solution to schedule and automate backup of the website.**
  - With Web app Backup, you can create scheduled backups of your Web app. The offline copy is stored in Blob storage.

- **Security**
  - **Is it possible to allow our visitors to use a mix of legacy and modern browsers and still provide for secure transactions?**
  - Azure Web Apps provides support for both IP Based SSL and SNI SSL. The former is mechanism that should be used to support both legacy and modern browsers.
  - **What does Azure offer to help us with auditing access to our web servers?**
  - Operations logs are available from the Azure portal that can be used to track various management operations. These logs can be retrieved by REST API for collecting them into more permanent storage.
  - **Our staff is accustomed to accustomed to a single sign-on experience-- will this still be possible?**
  - Yes, this is possible using Azure Active Directory.

Potential Benefits

By integrating Microsoft cloud solutions into its datacenter strategy, Adventure Works Cycles has been able to reduce IT costs by more than $300,000 a year while gaining greater datacenter scalability and datacenter agility. The IT team can respond to server requests in hours instead of months and "turn off" servers when they are no longer needed. With servers running in Microsoft datacenters around the world, Adventure Works can provide better application performance and availability to offices and customers that are located far from the company's California datacenter.

Improve Scalability and Agility

Adventure Works has gained a level of IT scalability and agility that it never before had.

Cowles's team can respond much faster to requests for IT resources. "With Azure, we can respond to business requests in hours versus the months that it took before," Cowles says. "It's especially valuable in responding to requests in non-US regions, because it's difficult to set up infrastructure in Carlsbad and provide remote access to those applications."

Enhance Performance and Availability of Core Business Systems

Since moving important pieces of its infrastructure to Azure, Adventure Works has enjoyed higher overall availability of critical applications and web properties. "With Azure, we no longer have maintenance-related downtime that negatively affects our business around the world," Cowles says.

By migrating its business-to-business website to Microsoft Azure, Adventure Works will eliminate the cost of that on-premises infrastructure and also realize reliability and performance gains. "It's a lot easier to load-balance virtual machines in Azure than physical servers in our datacenter," Cowles says. "We can also mitigate the impact of server security updates. By moving the B2B infrastructure to Azure, we can apply updates only to the regional servers that need them without affecting uptime in all of our offices."

Customer Quote

"Previously, when our web traffic spiked, we experienced slowdowns. With web properties running in Microsoft Azure, we can scale our infrastructure quickly and proactively and improve customer experiences."

Hayley Leigh, Manager of Solution Development, Adventure Works Cycles

Lab: Deploying Database Instances in Azure

Scenario

A local app development firm has hired your company to show them how to automate the deployment of their API and databases. The team has used Cosmos DB for a few months to power their mobile application's REST API. They have a test team, located on another continent, that will need the ability to deploy instances of their backend API application and database multiple times a month to perform integration and regressions testing.

Objectives

- Build an ARM Template to deploy a CosmosDB database instance and an API App with

code.

Lab setup

Estimated Time: 90 minutes

Virtual machine: **20535A-SEA-ARCH**

User name: **Admin**

Password: **Pa55w.rd**

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Deploy a CosmosDB Database Instance

Exercise 2: Validate the REST API

Exercise 3: Cleanup Subscription

Review Question(s)

Module review and takeaways

Review Question(s)