When should you use Azure Cosmos DB for NoSQL

5 minutes

Azure Cosmos DB for NoSQL is a fully managed NoSQL database service for modern app development. It provides guaranteed single-digit millisecond response times and 99.999-percent availability, backed by SLAs with automatic and instant scalability.

For enterprise scenarios, Azure Cosmos DB for NoSQL has a comprehensive suite of financially backed service level agreements (SLAs) ⧉ that cover throughput, consistency, availability, and latency.

Common use cases for the Azure Cosmos DB for NoSQL

As a fast NoSQL database with a flexible API, Azure Cosmos DB for NoSQL is well suited for high-performance applications with global ambition. Speed and flexibility make Azure Cosmos DB for NoSQL great for web, retail, IoT, gaming, and mobile applications. Azure Cosmos DB for NoSQL is a good fit for applications that require flexibility, low response time, and transactions at massive volume or velocity.

IoT/telemetry

IoT sensor workloads can be diverse and unpredictable. These workloads require a database platform that is responsive and capable of elastic scale to meet the needs of as many, or as few, data collection devices deployed at any point in time. In many cases, IoT workloads can also write massive volumes of data and would benefit from a write-optimized database platform. Azure Cosmos DB for NoSQL is optimized for write-heavy workloads. Operations on a single document are atomic with fast and predictable indexed write speeds. Azure Cosmos DB for NoSQL is also globally distributed and elastic to meet the needs of various IoT workloads.
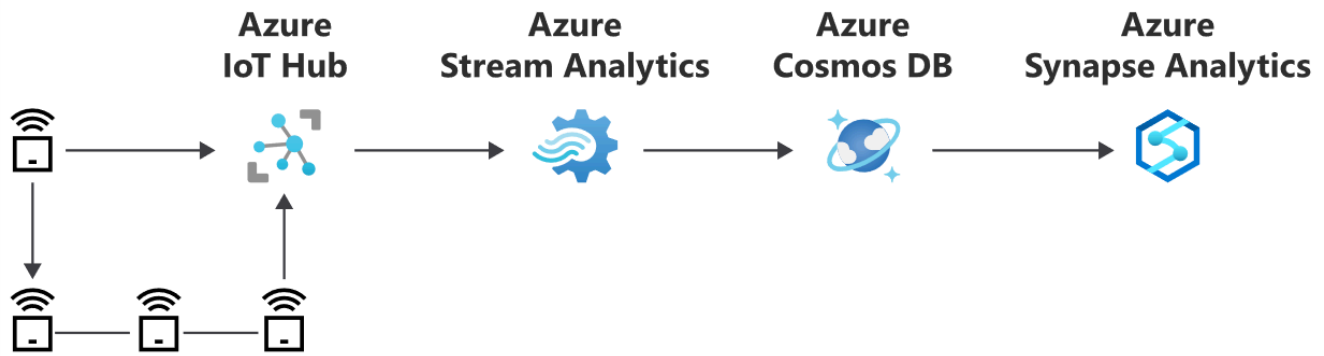
In this example, a growing number of IoT devices are sending sensor data to an Azure IoT Hub. Azure Stream Analytics then ingests and aggregates the data for storage in Azure Cosmos DB for NoSQL. Azure Synapse Analytics then performs near real-time analysis over the data using Azure Synapse Link for Azure Cosmos DB.

Retail/marketing

Azure Cosmos DB for NoSQL is a great fit for retail and marketing workloads that can experience dramatic and unexpected swings in usage at any point throughout the year. The elastic scale of Azure Cosmos DB for NoSQL ensures that the database platform can handle requests during peak usage, and save money during nonpeak times.
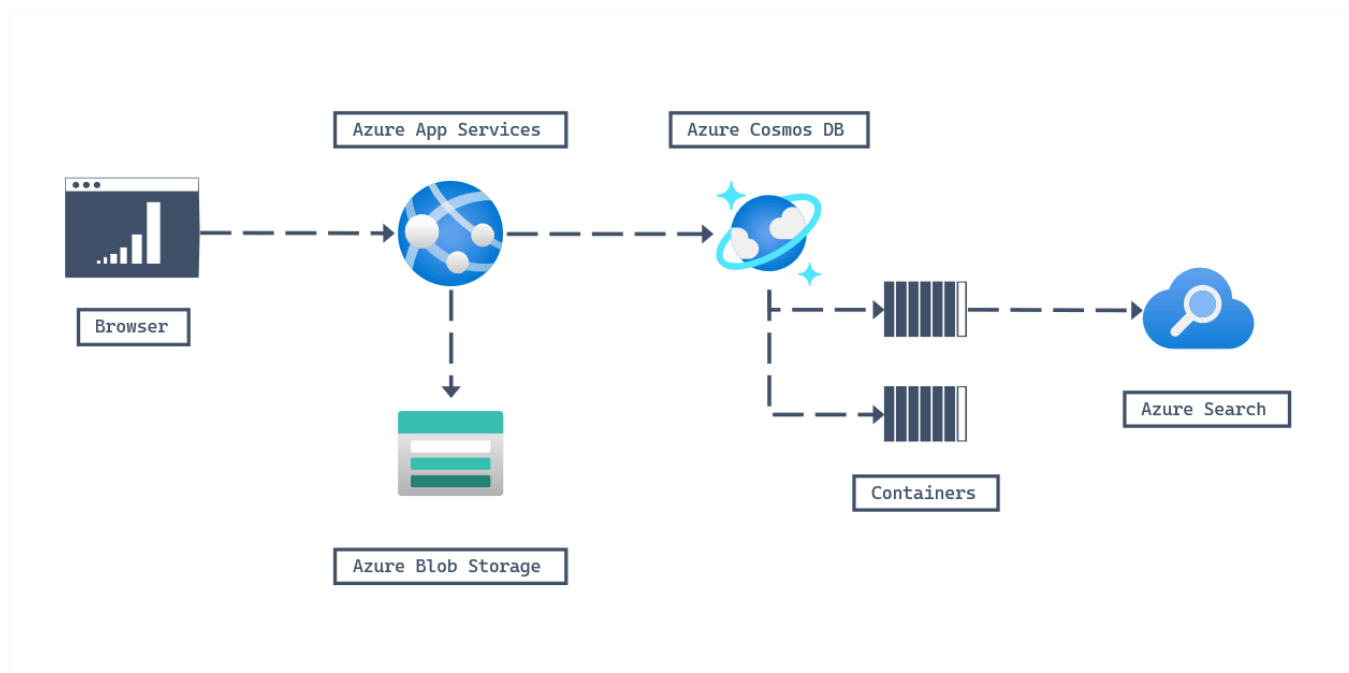
In this example, a JavaScript web application, built on content stored in Azure Blob Storage, uses Azure Cosmos DB for NoSQL as it's backing database. Multiple accounts are used to manage different facets of the solution such as the shopping cart, inventory, or catalog. The solution then uses Azure Search to index the Azure Cosmos DB for NoSQL data, providing a rich search experience to end users.

Web/mobile

Many modern social applications generate a plethora of user-generated content that is diverse in quantity, shape, and volume. Azure Cosmos DB for NoSQL is a great candidate for this workload as this API can store data of varying schemas. Consider the NoSQL API for data that may have schemas that change or evolve over time as the company's initiatives expand into new areas.
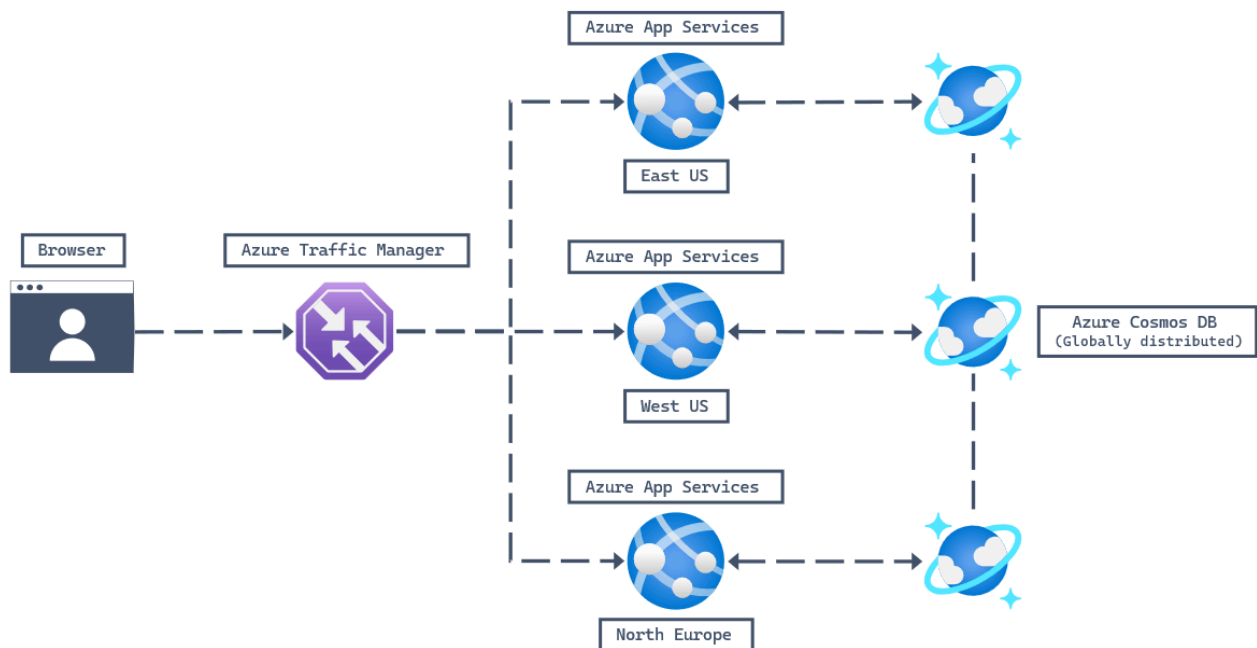
In this example, a user is using a URL to access a web site in their browser. The URL points to Azure Traffic Manager, which then uses a built-in algorithm to determine which Azure App Service endpoint to redirect the user to. Since Azure Cosmos DB for NoSQL is capable of global distribution, you only need one account that is replicated across multiple regions.

Module Scenario

Consider the scenario from the beginning of this module:

> Suppose you work as the lead developer at a retail company. Your team is building your online storefront. You're designing the new storefront to be accessible across various devices including mobile. The team expects a spike in demand when the storefront is published and various "grand opening" sales begin.

One key part of your store's success is the ability for the company to notify users of shipping updates regardless of what device they place the order on or are currently using. Your team has worked hard on a sophisticated system to manage detailed order status tracking. The tight integration of Azure Cosmos DB with other Azure services, let's you consider building solutions

that use order data in Azure Cosmos DB for NoSQL to send notification to your user's mobile devices. The notifications alert them when their package ships, or is out for delivery.
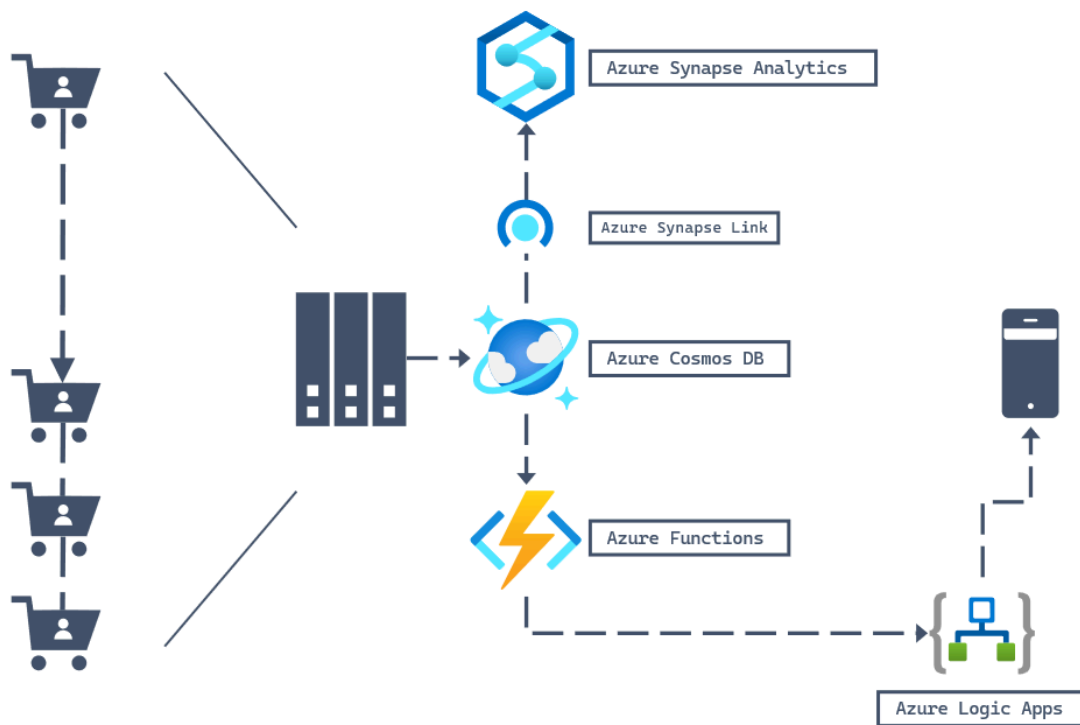
This example is similar to the example from the introduction of this module. To build on the first example, your team has decided to introduce Azure Cosmos DB for NoSQL as the database of choice. Now, your team can use Azure Synapse Link to prepare and aggregate data for deeper analysis using Azure Synapse Analytics. Your team can also use services such as Azure Functions to react to data events with Azure Cosmos DB, and then trigger an Azure Logic Apps workflow that sends notifications to mobile devices.

Next unit: Knowledge check

Continue >