



Get started with Azure Cosmos DB for NoSQL



Agenda

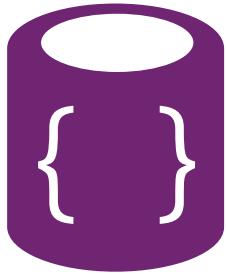
- Introduction to Azure Cosmos DB for NoSQL
- Try Azure Cosmos DB for NoSQL

Introduction to Azure Cosmos DB for NoSQL



NoSQL Concepts

NoSQL databases are defined by these shared characteristics:



Non-relational



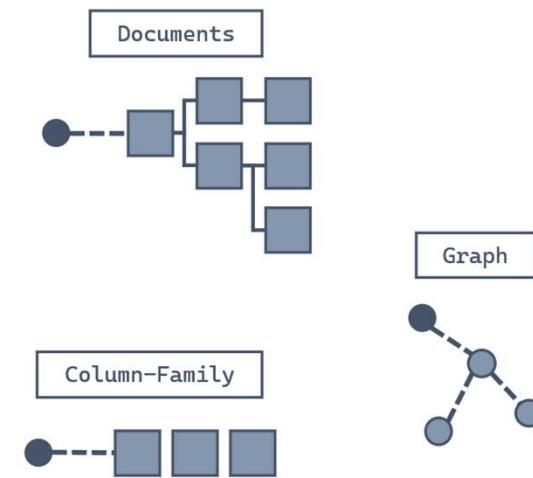
Schema less



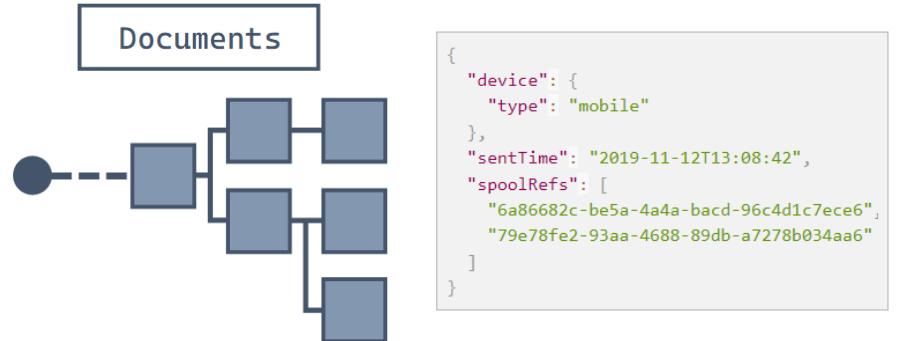
Designed for scale-out

NoSQL data models

NoSQL data models supported by Azure Cosmos DB



Azure Cosmos DB for NoSQL, the NoSQL document data model and JSON documents

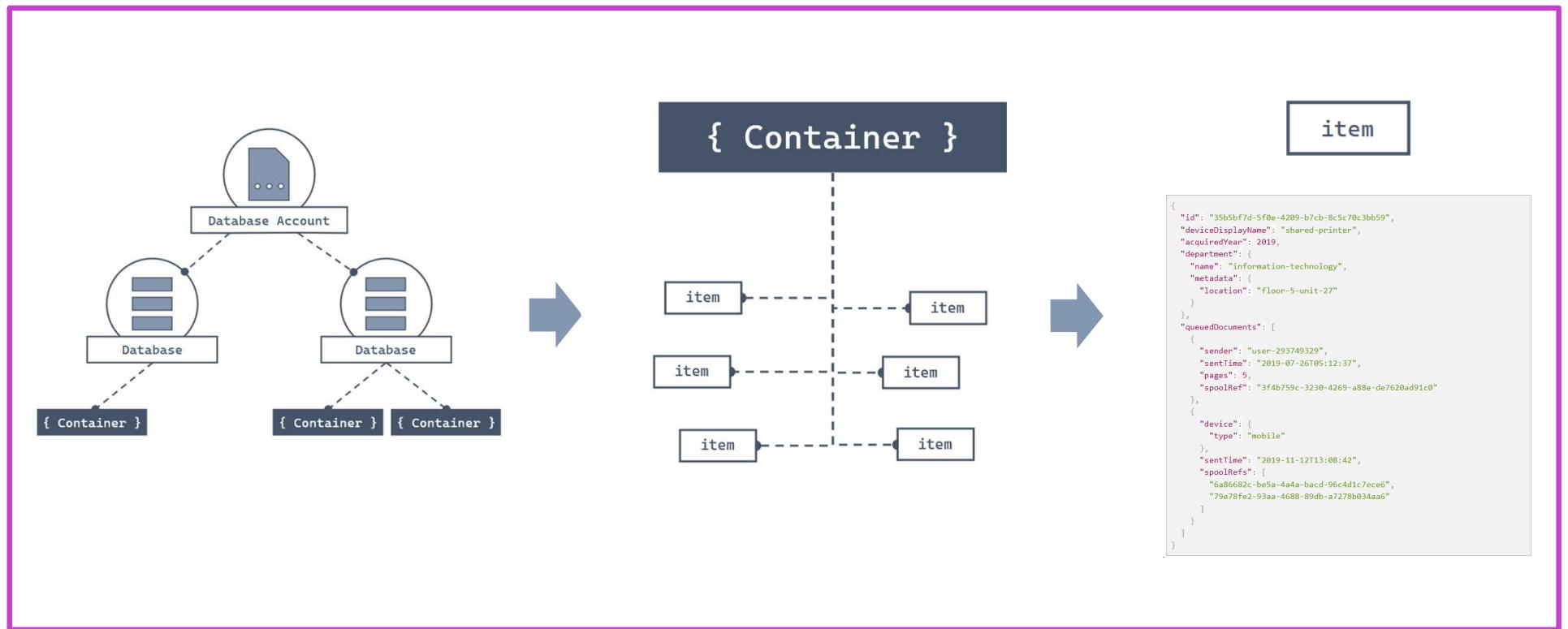


What is Azure Cosmos DB for NoSQL?

It's a fast fully managed NoSQL database service that offers:



What are the components of Azure Cosmos DB for NoSQL?



When should you use Azure Cosmos DB for NoSQL?

To support applications that require a database platform that is:

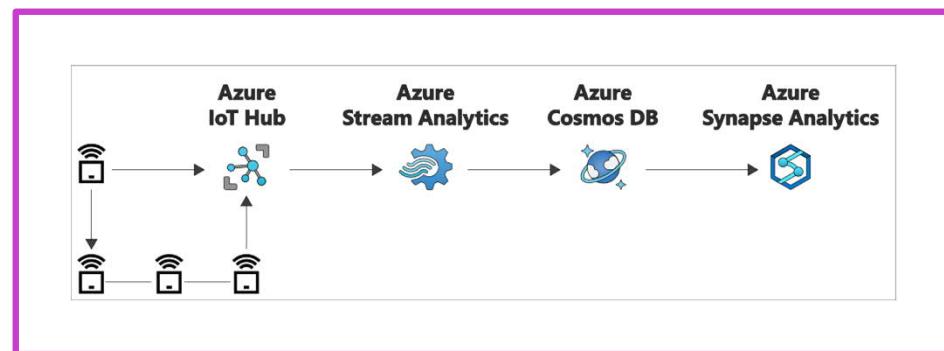
- Flexible
- Has a low response time
- Globally available
- Processing transactions at massive volume or velocity
- Elastically scalable to meet the application needs

To support application workloads like:

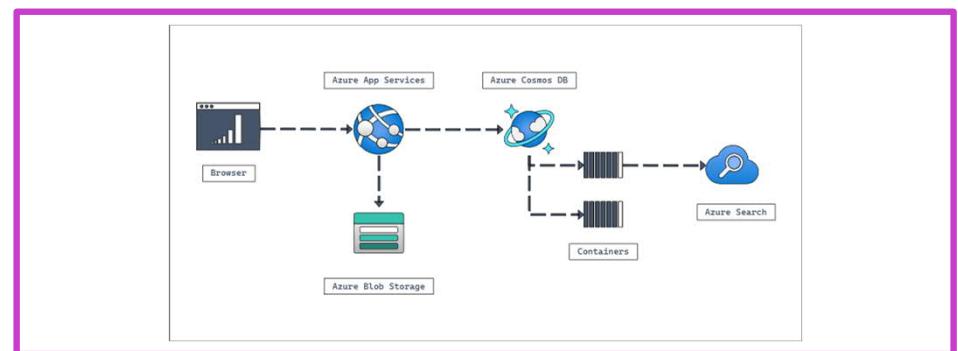
- Web
- Retail
- IoT
- Gaming
- Mobile

Common use cases for the Azure Cosmos DB for NoSQL

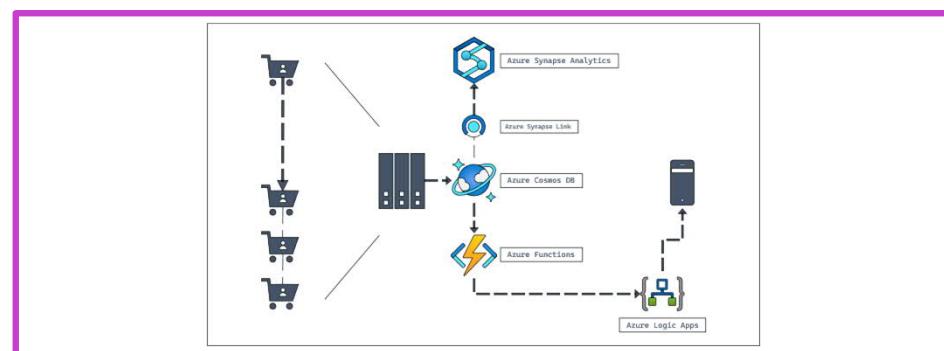
IoT/telemetry



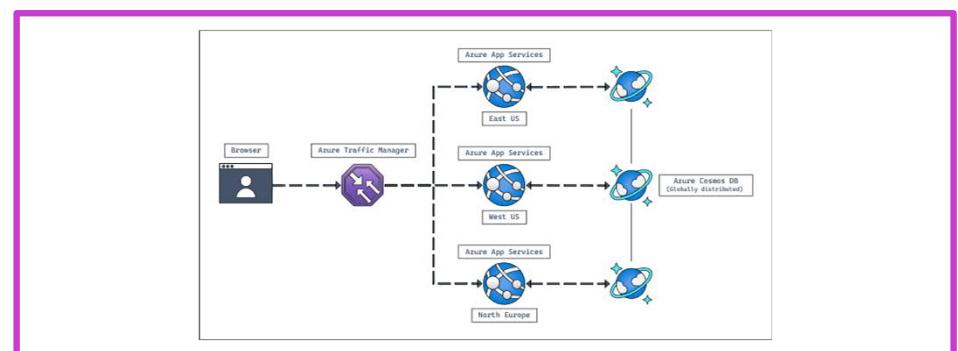
Retail/marketing



Online Storefront



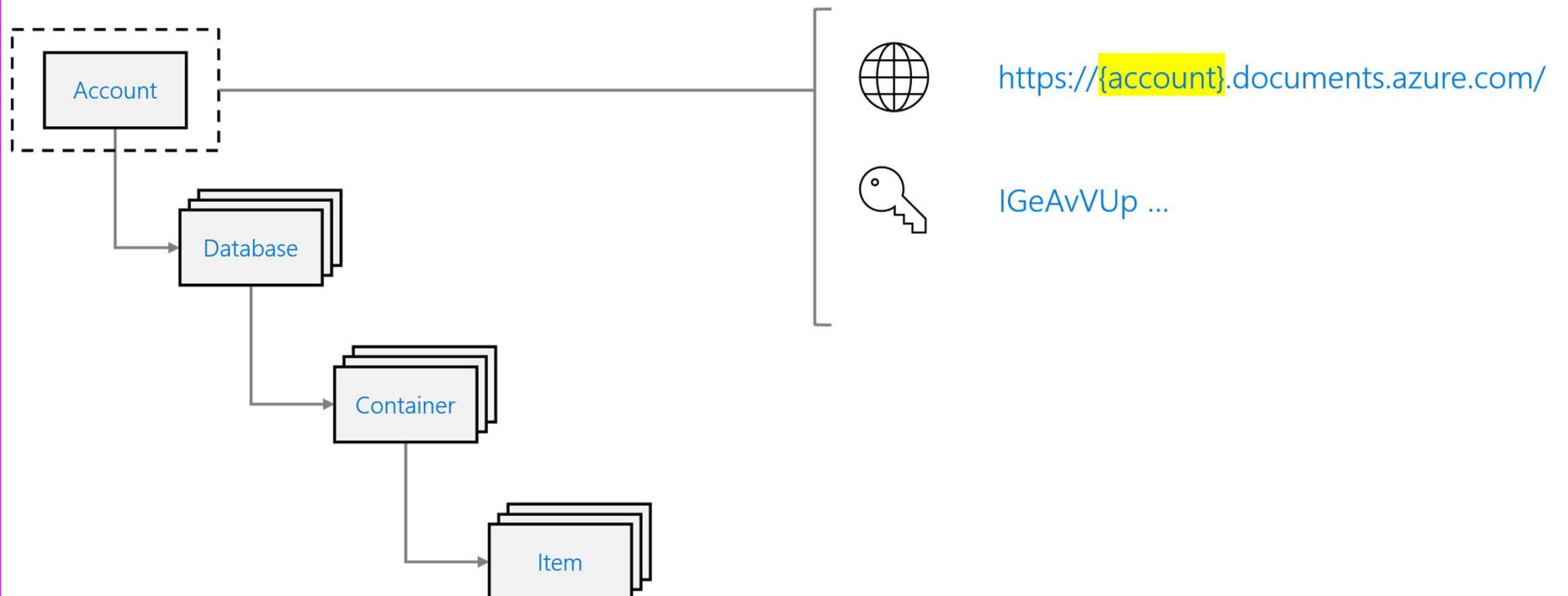
Web/mobile



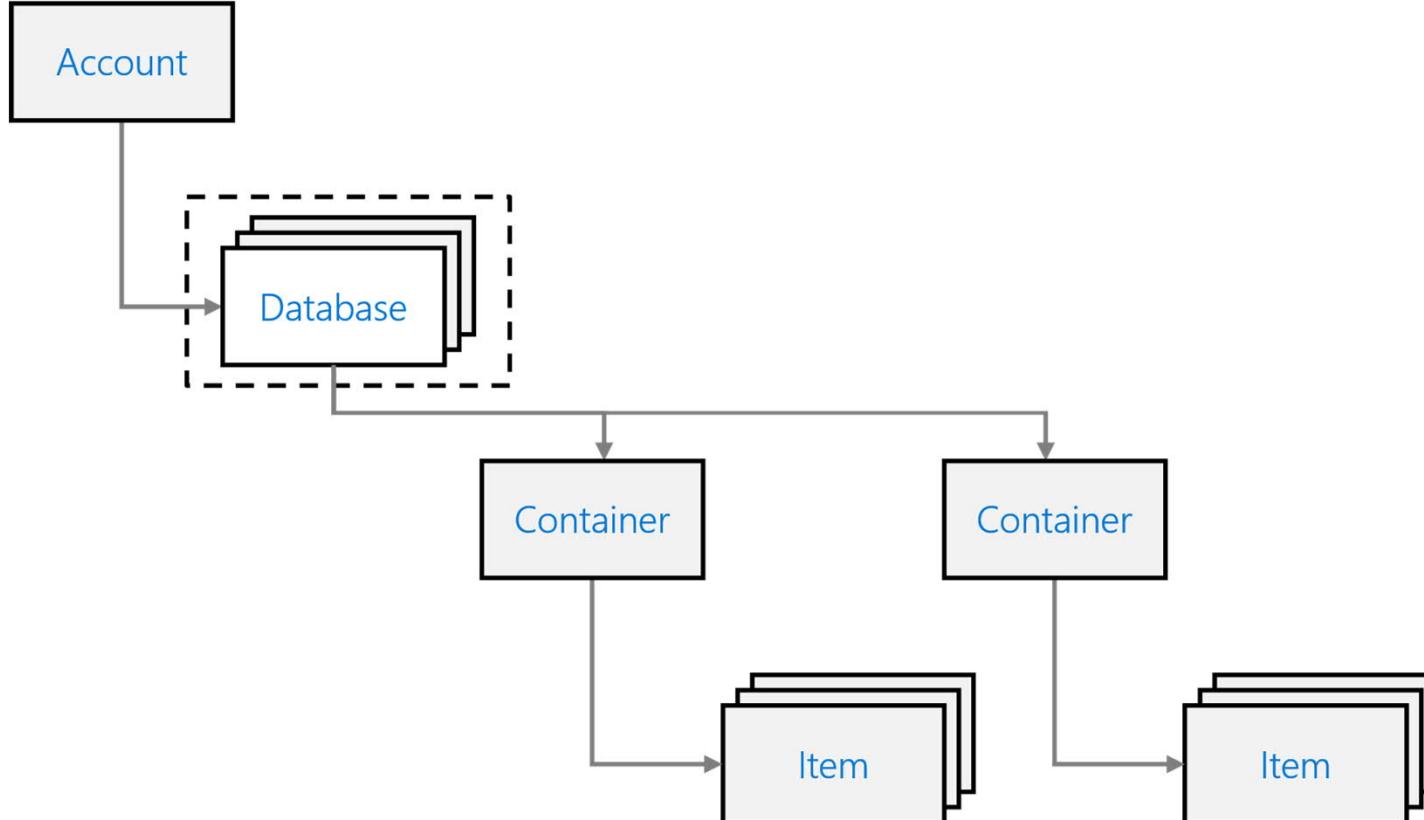
Try Azure Cosmos DB for NoSQL



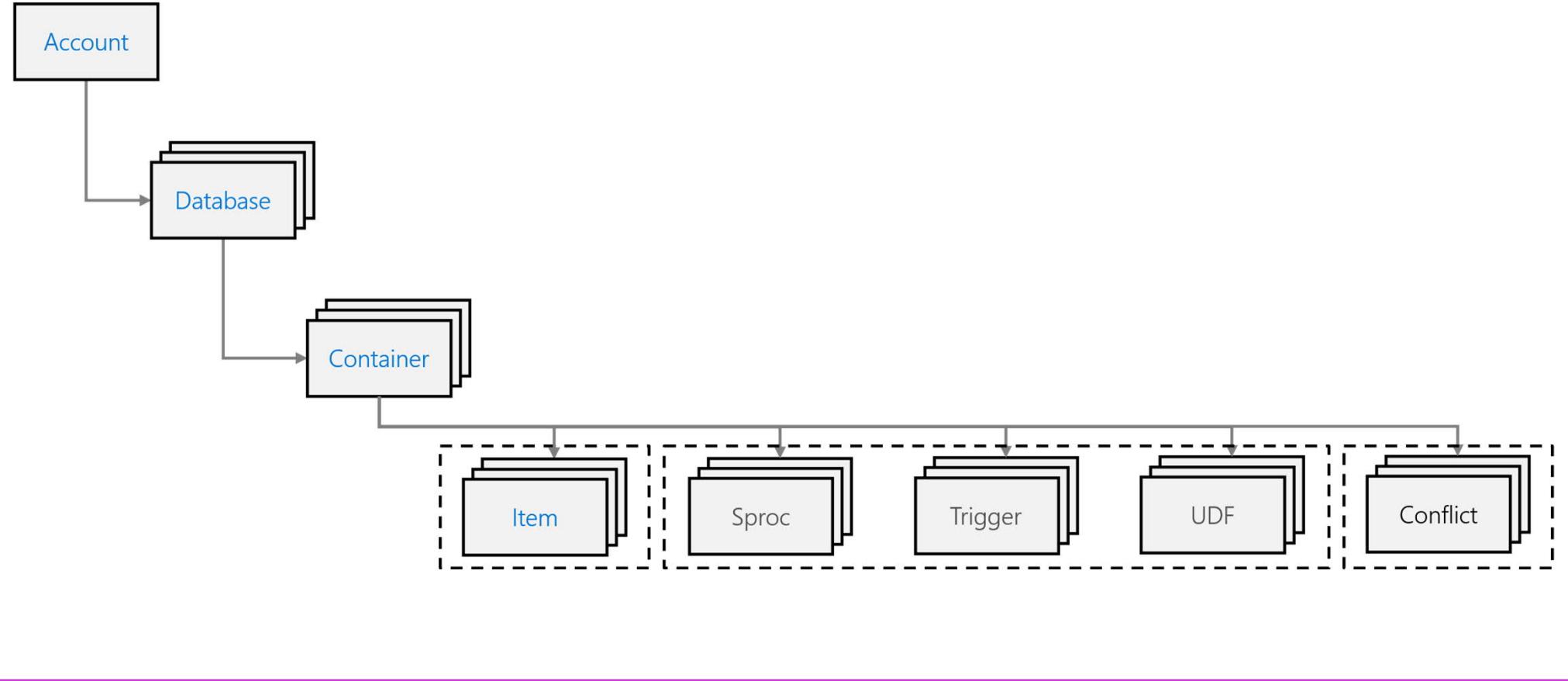
Explore Azure Cosmos DB for NoSQL resources



Explore Azure Cosmos DB for NoSQL resources



Explore Azure Cosmos DB for NoSQL resources



Creating a new Azure Cosmos DB for NoSQL account

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Azure Cosmos DB for NoSQL Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java. Create Learn more	Azure Cosmos DB for MongoDB Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB. Create Learn more	Azure Cosmos DB for Apache Cassandra Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB. Create Learn more
Azure Cosmos DB for Table Fully managed database service for apps written for Azure Table storage. Recommended if you have existing Azure Table storage workloads that you plan to migrate to Azure Cosmos DB. Create Learn more	Azure Cosmos DB for Apache Gremlin Fully managed graph database service using the Gremlin query language, based on Apache TinkerPop project. Recommended for new workloads that need to store relationships between data. Create Learn more	Azure Cosmos DB for PostgreSQL Fully-managed relational database service for PostgreSQL with distributed query execution, powered by the Citus open source extension. Build new apps on single or multi-node clusters—with support for JSONB, geospatial, rich indexing, and high-performance scale-out. Create Learn more

Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

[Basics](#) [Global Distribution](#) [Networking](#) [Backup Policy](#) [Encryption](#) [Tags](#) [Review + create](#)

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. Try it for free, for 30 days with unlimited renewals. Go to production starting at \$24/month per database. [Learn more](#)

Project Details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Concierge Subscription
Resource Group * Create new

Instance Details

Account Name *

Location *

Capacity mode Provisioned throughput Serverless
[Learn more about capacity mode](#)

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated \$64/month discount per account.

Apply Free Tier Discount: Apply Do Not Apply

Limit total account throughput: Limit the total amount of throughput that can be provisioned on this account
This limit will prevent unexpected charges related to provisioned throughput. You can update or remove this limit after your account is created.

Creating a database, container and item

The image displays three sequential screenshots from the Azure portal interface, illustrating the step-by-step creation of a database, container, and item.

- New Database:** This screen shows the configuration for creating a new database. It includes fields for "Database id" (with placeholder "Type a new database id"), "Provision throughput" (checked), "Database throughput (autoscale)" (radio button selected), and "Autoscale" (radio button selected). A note indicates to "Estimate your required RU/s with capacity calculator." The "Database Max RU/s" field is set to 4000. A cost estimate of \$35.04 - \$350.40 is shown. The "OK" button is at the bottom.
- New Container:** This screen shows the configuration for creating a new container. It includes fields for "Database id" (with placeholder "Type a new database id"), "Create new" (radio button selected), "Share throughput across containers" (checked), "Database throughput (autoscale)" (radio button selected), and "Autoscale" (radio button selected). A note indicates to "Estimate your required RU/s with capacity calculator." The "Database Max RU/s" field is set to 4000. A cost estimate of \$35.04 - \$350.40 is shown. It also includes fields for "Container id" (e.g., Container1) and "Partition key" (/pk). The "Unique keys" section has an "Add unique key" link.
- Items:** This screen shows the list of items in the container. The query "SELECT * FROM c" is displayed. The results table shows three items with the following data:

id	/pk
1	
2	"id": "replace_with_new_document_id"
3	

A "Load more" button is visible at the bottom of the table.

Lab – Create an Azure Cosmos DB for NoSQL account

Create a new Azure
Cosmos DB account

Use the Data Explorer
to create new items

Use the Data Explorer
to create a new
database and
container

Use the Data Explorer
to issue a basic query

Review



1 Which API is considered the "native API" for Azure Cosmos DB?

- MongoDB.
- Cassandra.
- NoSQL.

2 Which property is required to be specified when creating a new Azure Cosmos DB for NoSQL container?

- Time-to-live.
- Partition key path.
- Indexing policy.

3 Which one of these resources is a child resource of an Azure Cosmos DB for NoSQL account and parent resource for a container?

- Triggers
- Database
- Stored procedure



© Copyright Microsoft Corporation. All rights reserved.