Microsoft Azure

Optimizing performance and managing costs with Azure Cosmos DB

Best practices for new customers and workloads

Stay up to date with the latest announcements, tips, and documentation at:

AzureCosmosDB.com

devblogs.microsoft.com/cosmosdb/

Twitter.com/AzureCosmosDB

YouTube.com/AzureCosmosDB

Welcome

Get the most out of Azure Cosmos DB for your applications and workloads by ensuring your data modeling, partitioning strategy, and data distribution across Azure regions (also known as geo-replication or geo-redundancy) are optimized. Each of these areas can impact database performance, scalability, and costs. The best practices in this guide will help as you get started building databases or applications with Azure Cosmos DB.

Before you begin:

1. Review the differences between the serverless and provisioned throughput capacity modes, to ensure you've selected the right one for your needs. Should you wish to switch modes, you'll need to create a new Azure Cosmos DB account and select the desired capacity mode in the "Instance Details" settings.
2. Check your account settings to ensure your instance's Azure region (location), data distribution (geo-redundancy), multi-region writes (also known as "multi-master") and others are set as you wish.
3. Read Relational vs. NoSQL data if you're new to non-relational (NoSQL) databases

Best Practice 1: Data modeling

Data modeling is the process of establishing patterns and rules for how data is stored in database documents and containers and how it is accessed. You can significantly improve speed and performance - especially for your most frequent queries - by choosing the data model that best fits the needs of your application. We recommend spending time thinking about how your data will be queried, how many reads vs writes you anticipate, and how much data will be stored.

To learn about data modeling in Azure Cosmos DB:

1. Watch this 2-minute overview of data modeling
2. Watch a deep-dive video on data modeling for low-latency
3. Read documentation on data modeling

Best Practice 2: Partitioning

To achieve superior speed at any scale, you need a good partitioning strategy that optimizes database reads and writes. When you specify a partition key for each Azure Cosmos DB container it defines how data will be stored across many partitions. This is critical as your database scales, and data is spread across more partitions and servers.

To effectively manage partition keys:

1. Watch this 2-minute overview of data partitioning
2. Watch: choosing the right partition key and avoiding "hot" partitions
3. Read about best practices for choosing a partition key
4. See this real world example of how to model and partition data

   TIP: If you need to change your partition key, create a new container with your desired key and transfer data using Azure Data Factory. If you need to do an online data migration, read our guide to changing the partition key.

Best Practice 3: Multi-region data distribution (geo-replication) and multi-region writes (multi-master)

Azure Cosmos DB can replicate and distribute your data to any or all Azure regions. Single region Azure Cosmos DB accounts have guaranteed 99.99% availability and adding additional regions to your Azure Cosmos DB account can further improve availability. Adding additional regions also helps lower latency by placing data physically closer to your users.

Note: multi-region data distribution is not currently available on serverless accounts.

To manage multi-region settings, performance, and costs:

1. Watch the multi-region data distribution overview video
2. Deep dive into multi-region writes ("multi-master")
3. Check your selected regions and monitor activity; add or remove regions at any time

Best Practice 4: Performance & cost optimization

When you created your new account, you had to choose the database operations pricing model that best meets the requirements of the application or workload. This is an important decision, and it's not currently possible to switch between models once an account has been created. If you haven't yet, take some time to review the differences between the provisioned throughput and serverless models. If you think you picked the wrong one, no worries! You can still delete your account and re-create it under the right model.

Provisioned throughput offers guaranteed speed and availability and requires you to plan and manage throughput capacity. It is best suited for large, high-throughput workloads with high performance requirements. You can manually provision throughput or enable autoscale throughput.

Serverless is consumption-based and charges for the resources used by database operations, with no minimum cost. It is best suited for workloads with low traffic, occasional spikes or bursts, and moderate performance requirements.

To manage costs:

1. Watch "2-minute pricing with Azure Cosmos DB"
2. Take a full tour of the database operations models to discover which is best for your application or workload
3. Find ways to optimize your development and testing costs
4. Understand how to optimize costs for multi-region workloads