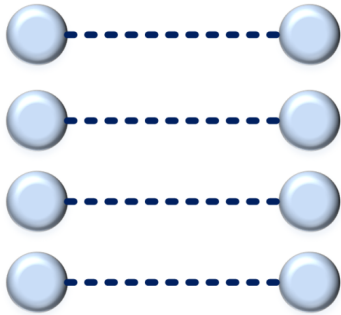
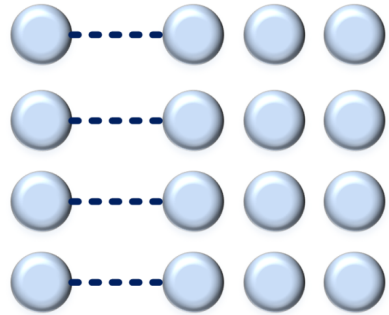


# Chapter 1: Introduction to NoSQL in Cosmos DB

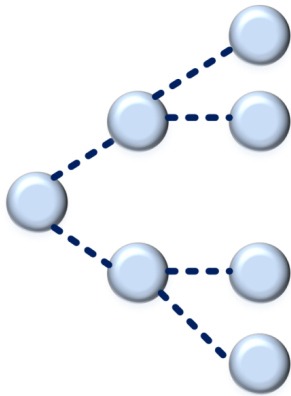




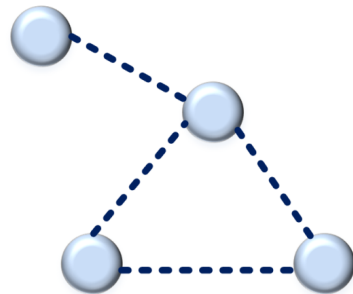
Key/Value



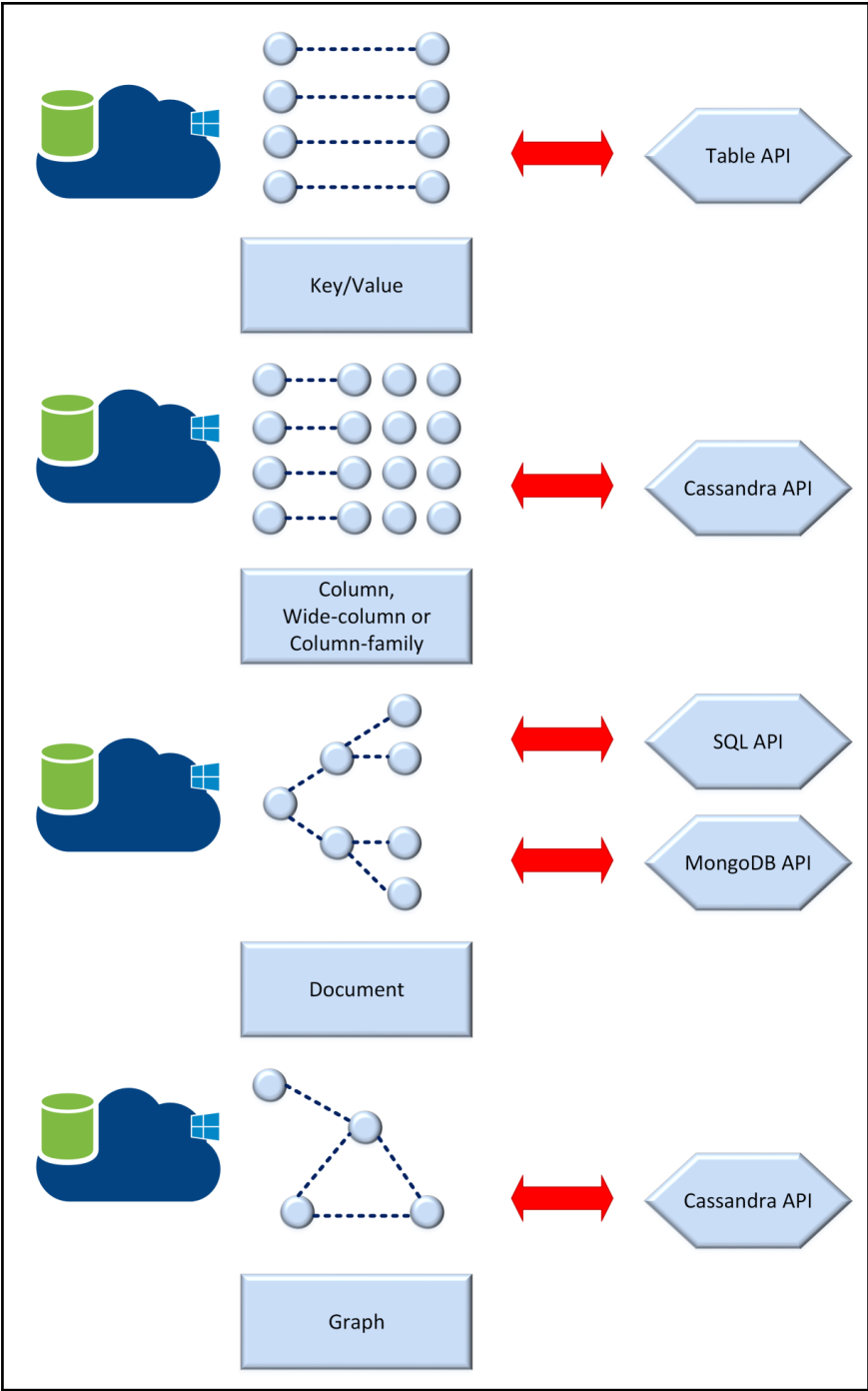
Column,  
Wide-column or  
Column-family

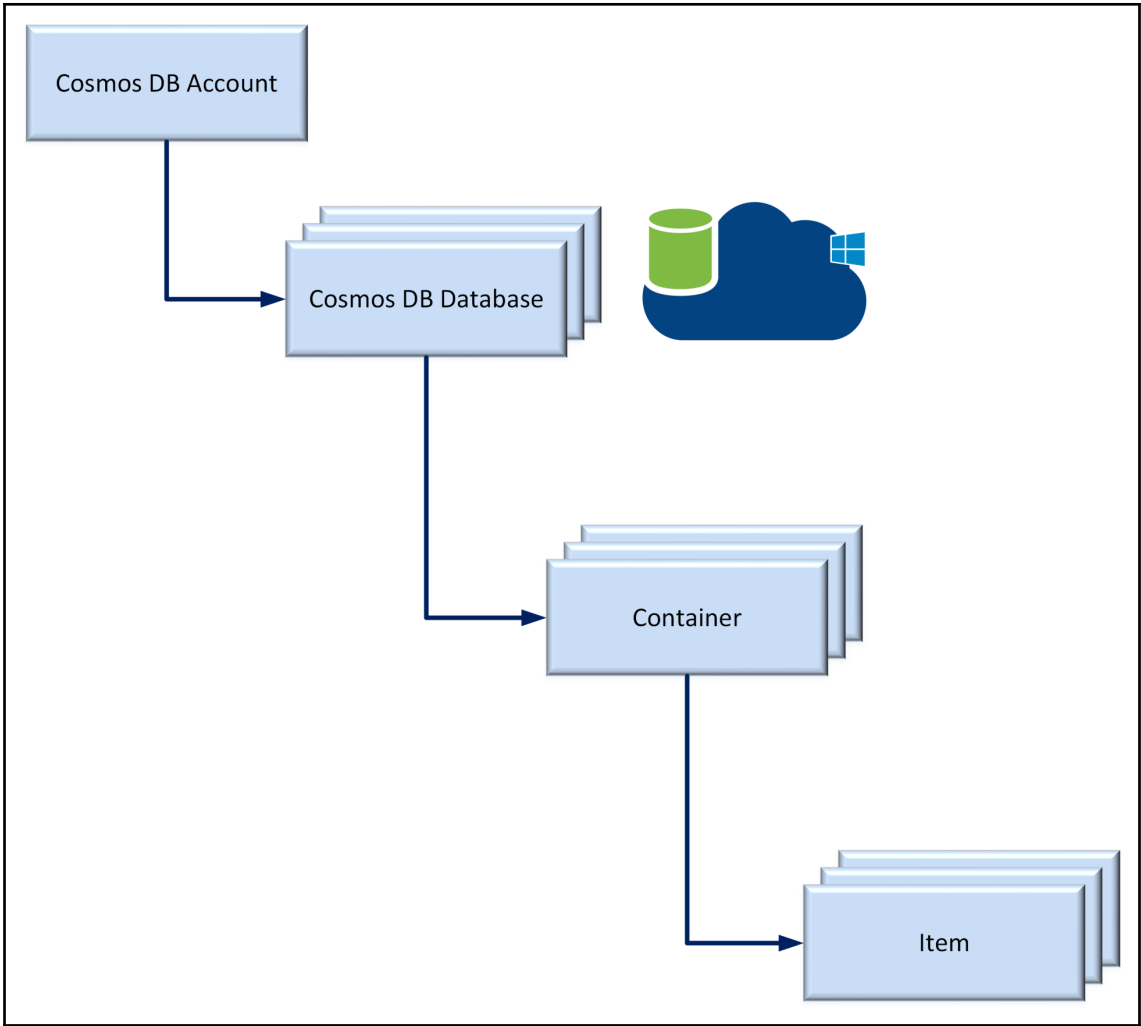


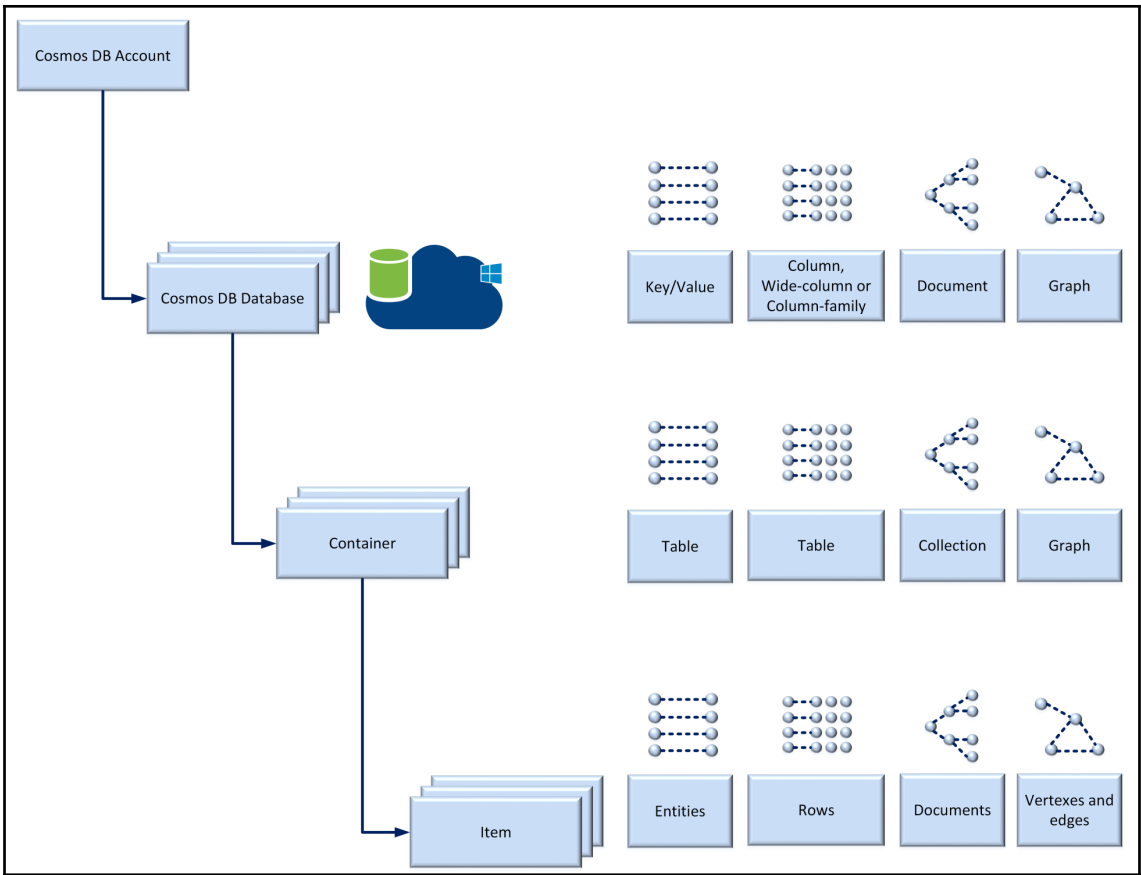
Document

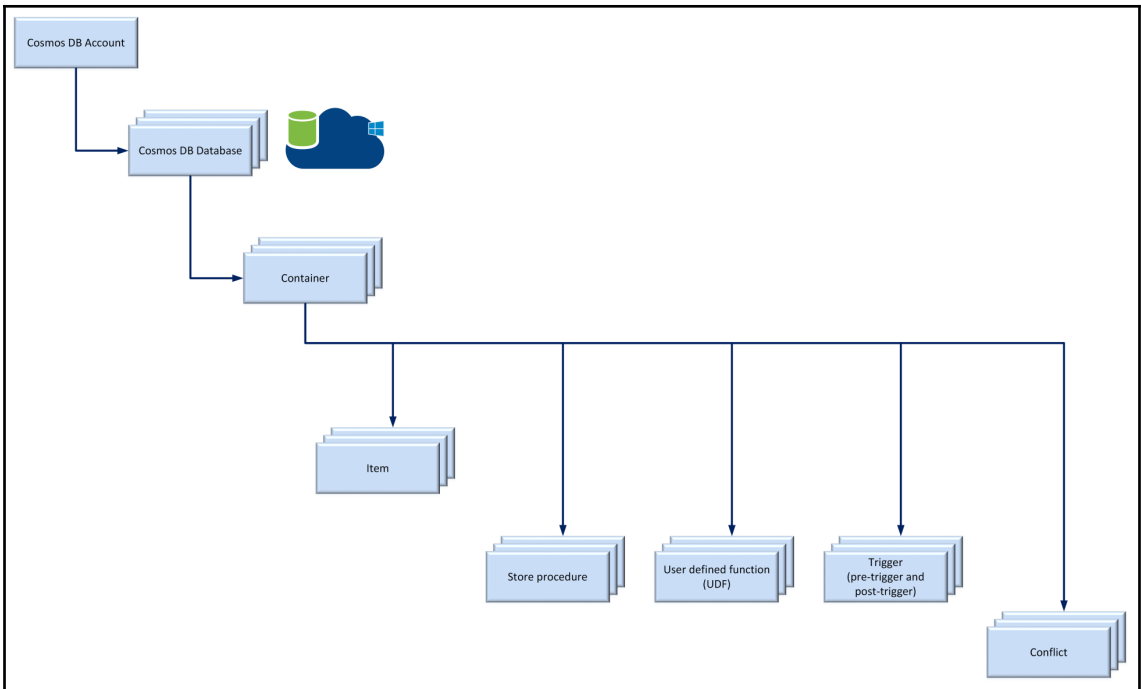


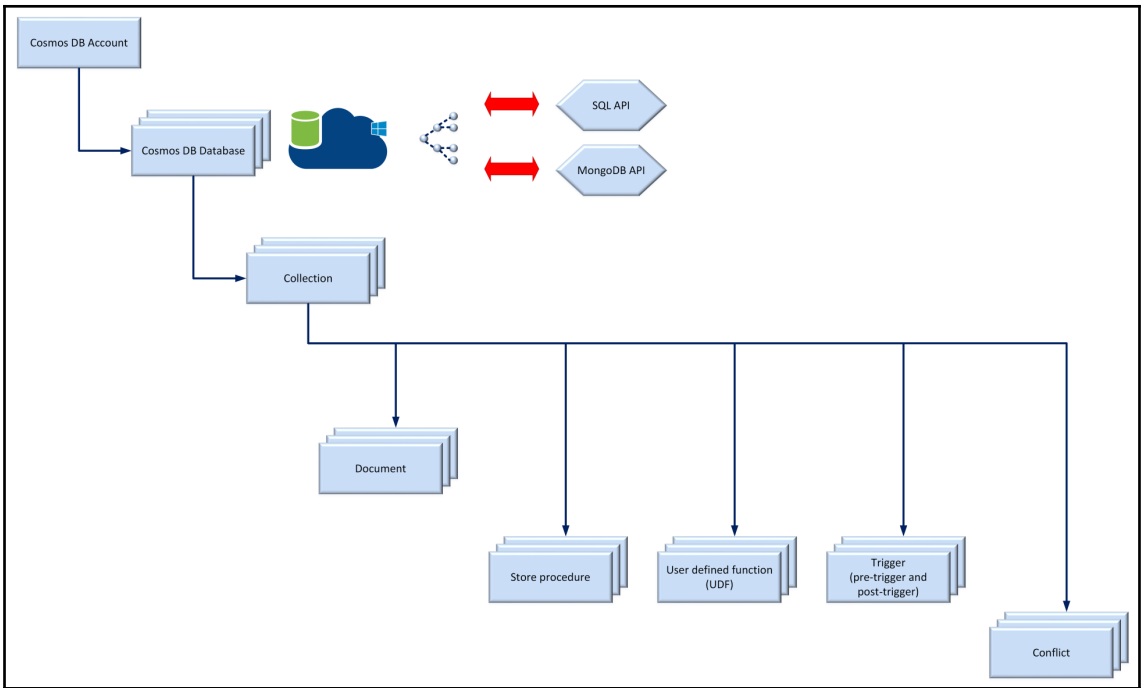
Graph

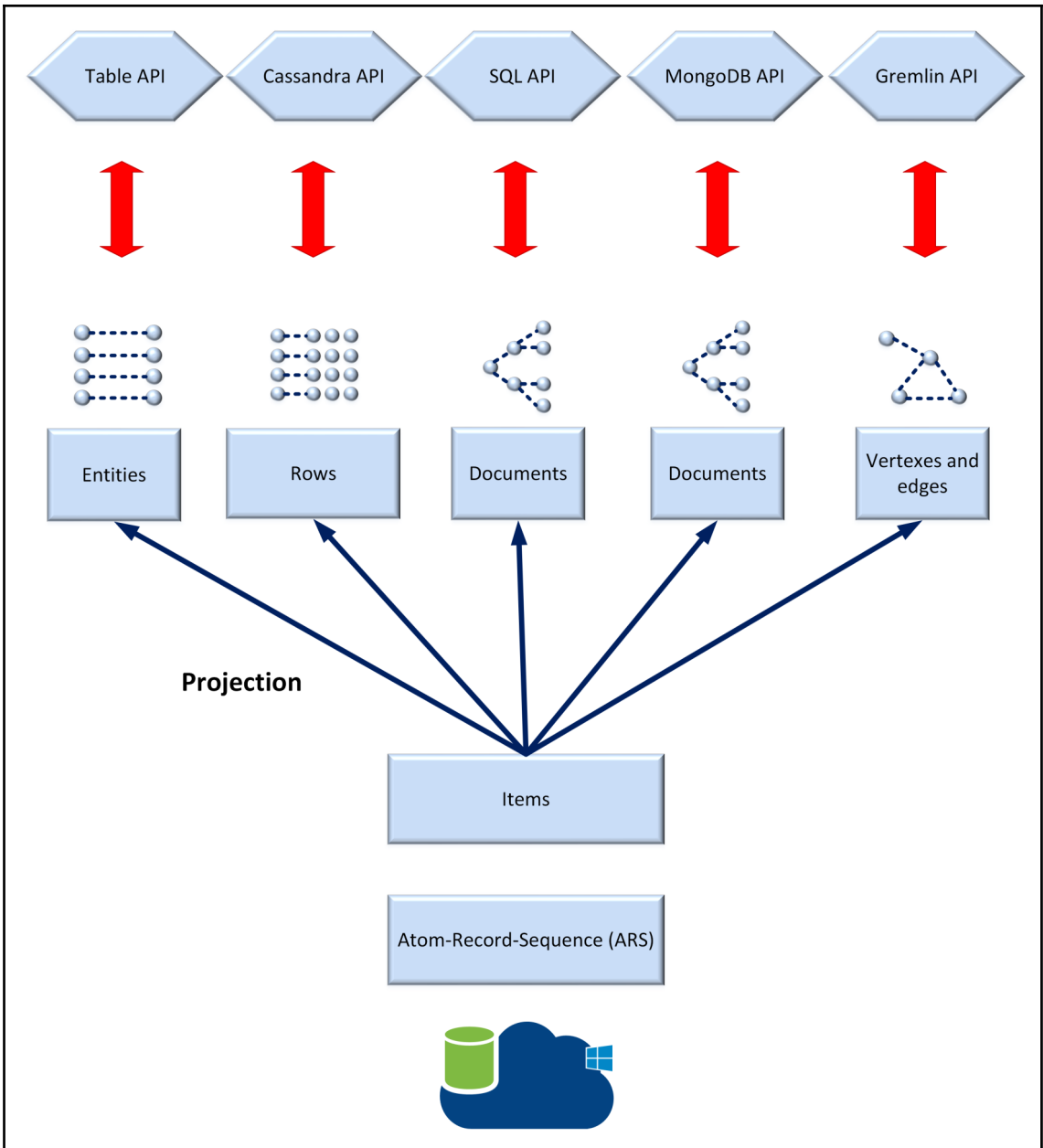




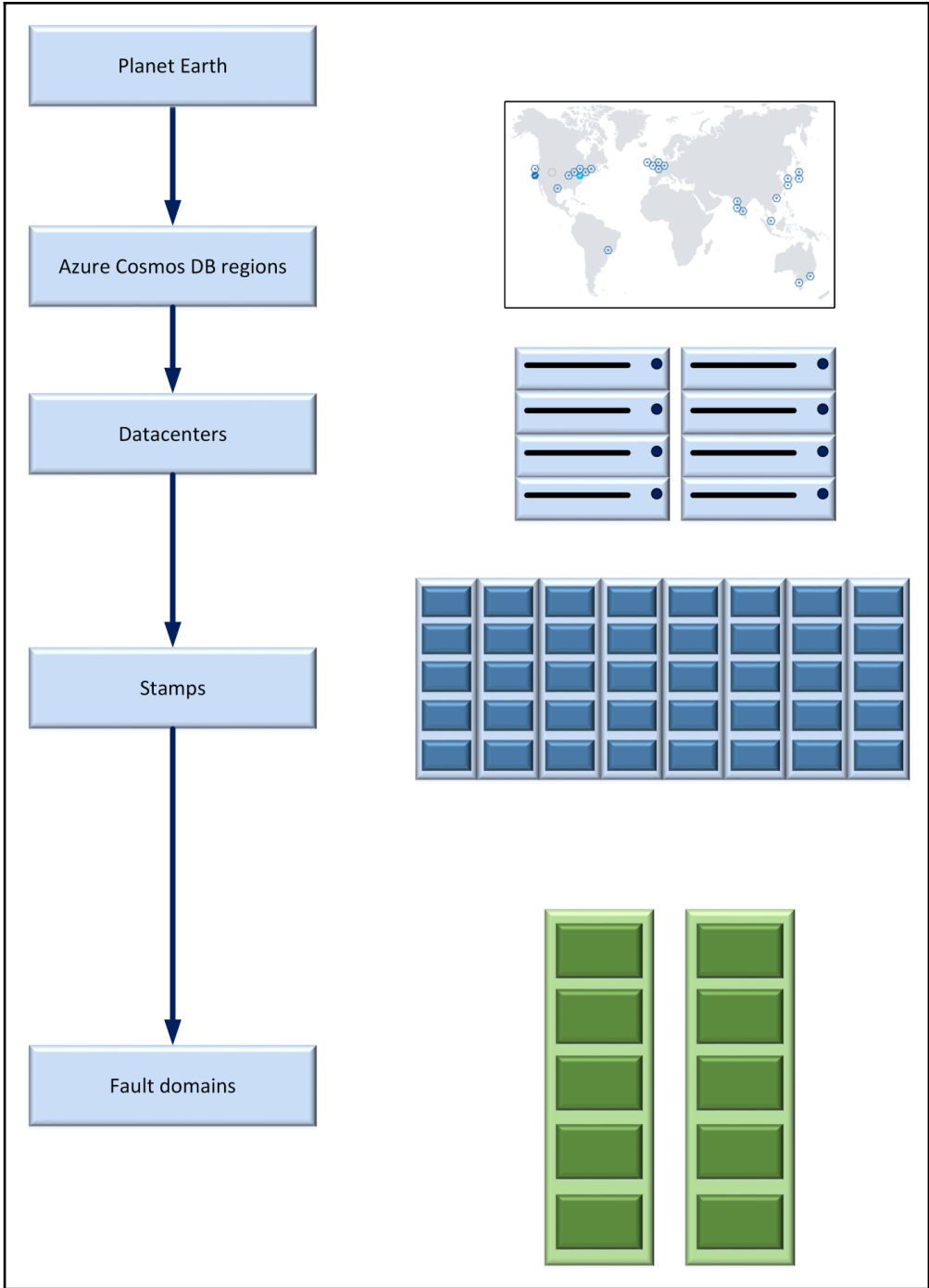


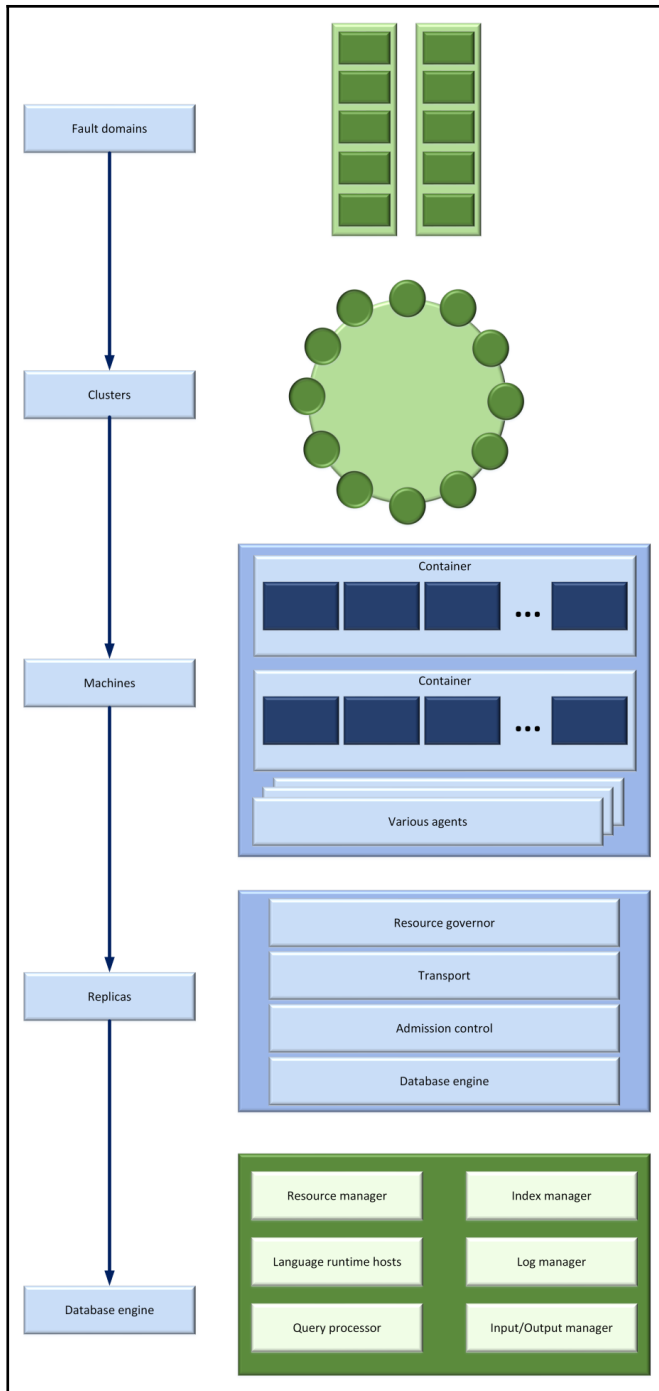


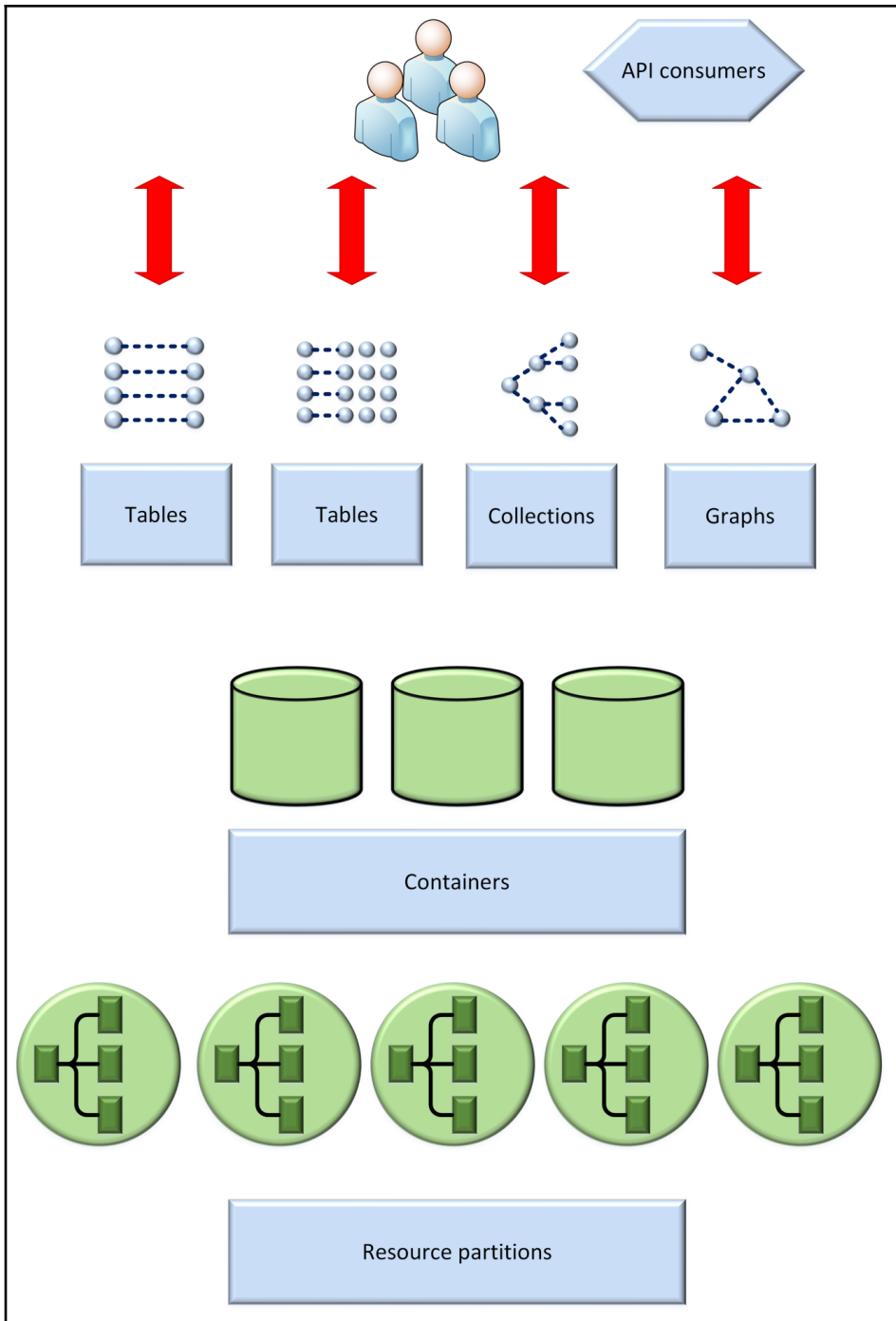


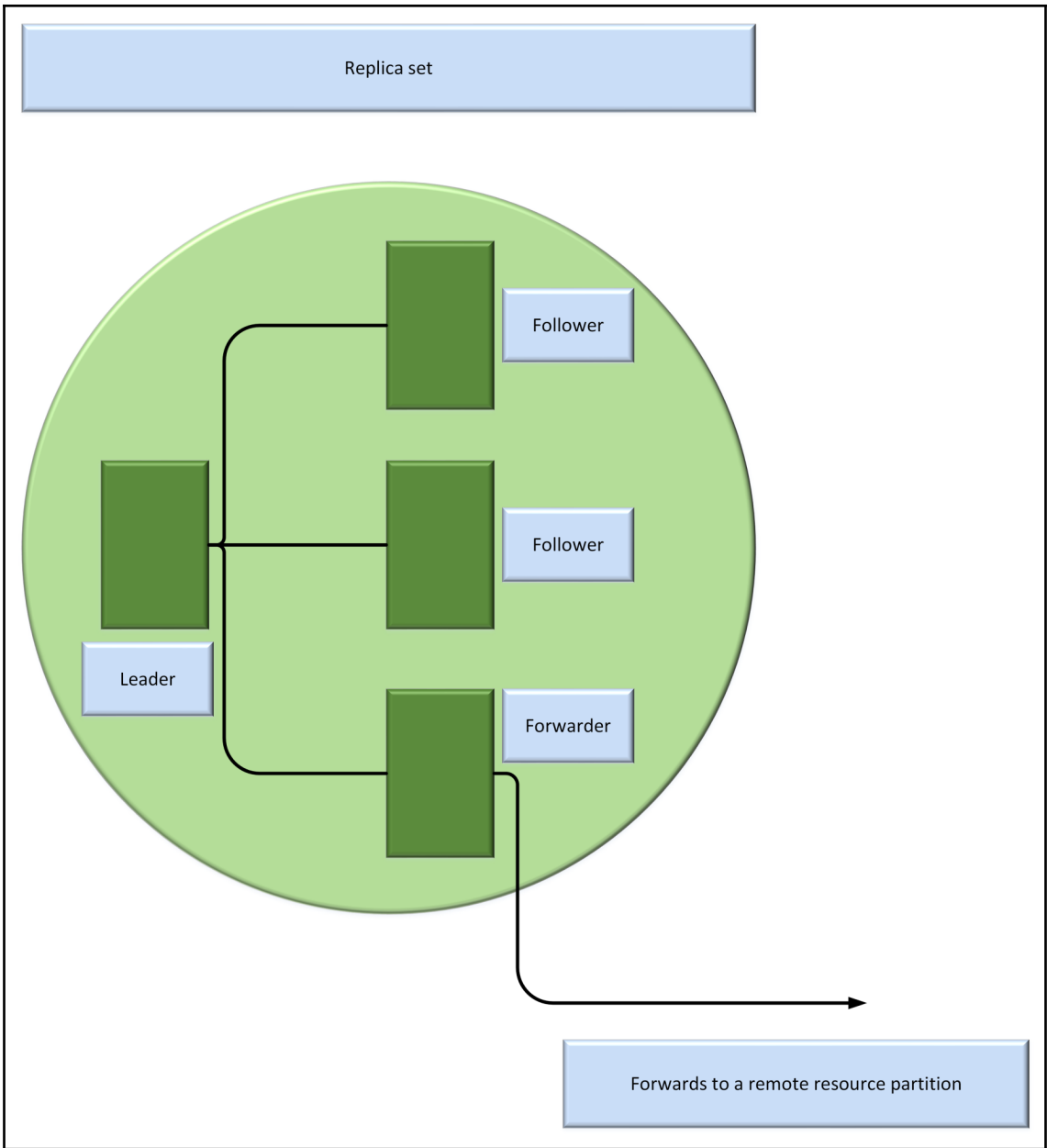




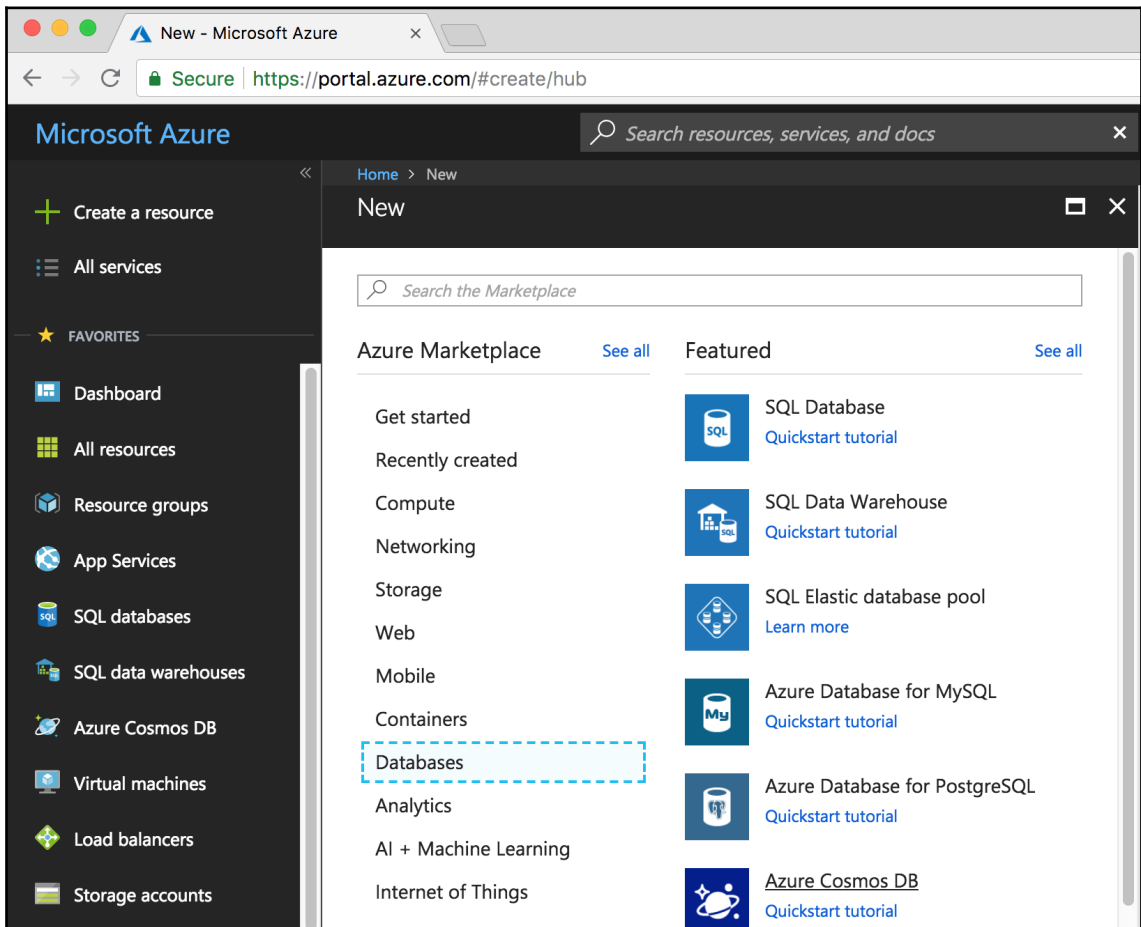








# Chapter 2: Getting Started with Cosmos DB Development and NoSQL Document Databases



Home > New > Azure Cosmos DB

## Azure Cosmos DB

New account

\* ID  
example001 ✓  
documents.azure.com

\* API ⓘ  
SQL

\* Subscription  
Visual Studio Ultimate with MSDN

\* Resource Group  
 Create new  Use existing  
example001 ✓

\* Location  
East US

Enable geo-redundancy ⓘ  
 Enable Multi Master ⓘ

---

Multi Master Preview  
Pending approval

---

Virtual networks  
Configure virtual networks ⓘ  
**Disabled** Enabled

**Create** Automation options

Notifications

Dismiss: [Informational](#) [Completed](#) [All](#)

Deployment in progress... Running

Deployment to resource group 'example001' is in progress.

Home > example001 - Quick start

example001 - Quick start  
Azure Cosmos DB account

Search (Ctrl+/)

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems  
Quick start  
Data Explorer

SETTINGS  
Replicate data globally

Congratulations! Your Azure Cosmos DB account was created.  
Now, let's connect to it using a sample app:

Choose a platform

.NET .NET Core Xamarin Java Node.js Python

1 Add a collection  
In Azure Cosmos DB, data is stored in collections.  
[Create 'Items' collection](#)  
Create 'Items' collection with 10GB storage capacity and 400 Request Units per second (RU/s) throughput capacity, for up to 400 reads/sec. Estimated hourly bill: \$0.033 USD

2 Download and run your .NET app  
Once collection is created, download a sample .NET app connected to it, extract, build and run.  
[Download](#)

Home > Azure Cosmos DB

Azure Cosmos DB  
Default Directory

+ Add Edit columns Refresh Assign tags

Subscriptions: 1 of 2 selected

Filter by name... Visual Studio Ultimate ... All resource groups All locations All tags No grouping

1 items

NAME	STATUS	LOCATION	SUBSCRIPTION
example001	Online	East US	Visual Studio Ultimate with MSDN

example001  
Azure Cosmos DB account


Search (Ctrl+)

+ Add Collection Refresh Move Delete Account Data Explorer Enable geo-redundancy

Status: Online  
 Resource group: example001  
 Subscription: Visual Studio Ultimate with MSDN  
 Subscription ID: cca381a2-  
 Read Locations: East US  
 Write Location: East US  
 URI: https://example001.documents.azure.co...

Collections  
 Looks like you don't have any collections yet. Data Explorer

Regions  
 Region Configuration: EXAMPLE001



Monitoring  
 Requests: 1 hour 24 hours 7 days  
 Total: Http 2xx, Http 400, Http 401, Http 403, Http 429, Http 5xx  
 No data to display  
 6 PM Aug 28 6 AM 12 PM  
 TOTAL REQUESTS

SETTINGS  
 Replicate data globally  
 Default consistency  
 Firewall and virtual networks  
 Keys  
 Add Azure Search  
 Add Azure Function  
 Locks  
 Automation script

COLLECTIONS  
 Browse  
 Scale

Home > example001 - Keys  
 example001 - Keys  
 Azure Cosmos DB account

Search (Ctrl+)

Data Explorer

SETTINGS  
 Replicate data globally  
 Default consistency  
 Firewall and virtual networks  
 Keys  
 Add Azure Search  
 Add Azure Function  
 Locks  
 Automation script

Read-write Keys Read-only Keys

URI  
 https://example001.documents.azure.com:443/

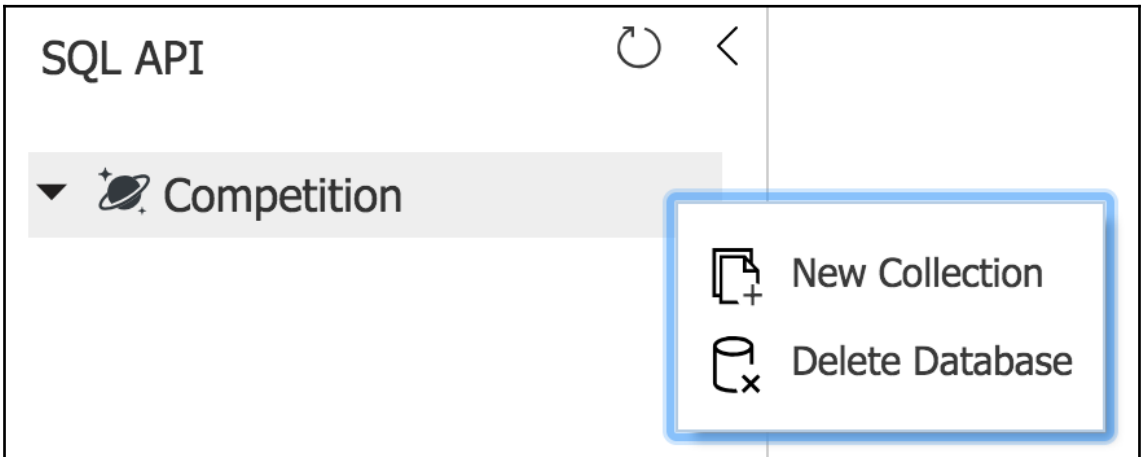
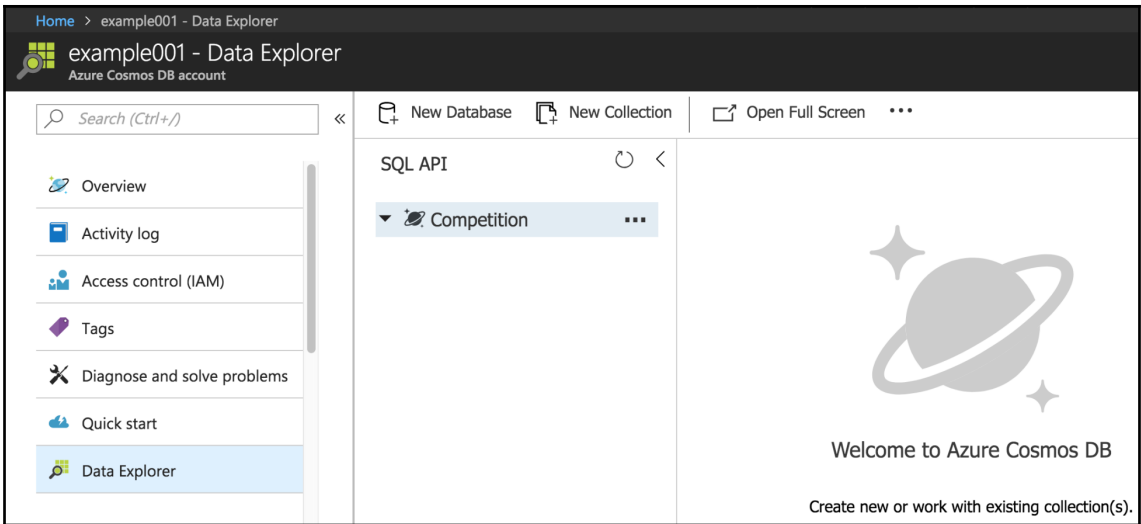
PRIMARY KEY  
 Yf36lYUZvcv5Wjbcse CdETg...

SECONDARY KEY  
 xulGz6XSjxh9eia99Tm B08irs...

PRIMARY CONNECTION STRING  
 AccountEndpoint=https://example001.documents.azure.com:443/;AccountKey=Yf36lYUZvcv5Wjbcse4TOW...

SECONDARY CONNECTION STRING  
 AccountEndpoint=https://example001.documents.azure.com:443/;AccountKey=xulGz6XSjxh9eia99TmVyZ3ll...





## Add Collection



\* Collection Id ⓘ

\* Storage capacity ⓘ

Fixed (10 GB)

Unlimited

\* Throughput (400 - 10,000 RU/s) ⓘ



Estimated spend (USD): \$0.080 hourly / \$1.92 daily.

Choose unlimited storage capacity for more than 10,000 RU/s.

Unique keys ⓘ



OK

Home > example001 - Data Explorer

example001 - Data Explorer  
Azure Cosmos DB account

New Database New Collection Open Full Screen New SQL Query New Stored Procedure Upload Delete Collection

SQL API

- Competition
  - VideoGames1
    - Documents
    - Scale & Settings
    - Stored Procedures
    - User Defined Functions
    - Triggers

Documents Scale & Settings

New Document

SELECT \* FROM c [Edit Filter](#)

id

Load more

Create new or work with existing document(s).

0 0 4

All Clear Notifications

- 12:36 AM Fetched 0 documents
- 12:16 AM Successfully created collection VideoGames1
- 11:46 PM Successfully created database Competition

SQL API

- Competition
  - VideoGames1
    - Documents
    - Scale & Settings
    - Stored Procedures
    - User Defined Functions
    - Triggers

Scale & Settings Documents

New Document Update Discard Delete

SELECT \* FROM c [Edit Filter](#)

id

Load more

```

1 {
2   "id": "1",
3   "name": "Battle Royale Kingdoms",
4   "lastCompetitionDate": "2018-09-29T04:36:22.7251173Z",
5   "tags": [
6     "mobile",
7     "2D",
8     "card game"
9   ],
10  "levels": [
11  {

```

0 0 2

All Clear Notifications

- 2:45 PM Successfully created new document for collection VideoGames1
- 2:39 PM Fetched 0 documents

SQL API >

Scale & Settings Documents Documents

New Document Update Discard Delete

SELECT \* FROM c [Edit Filter](#)

id	
1	<pre>24         "towerPower": 100 25     } 26 }, 27     "highestScores": [ 28     { 29         "player": { 30             "nickName": "Brandon in Wonderland", 31             "clan": "Wonderland Warriors" 32         }, 33         "score": "750" 34     } 35 ], 36 "_rid": "prUNAKtPjRoFAAAAAAAAAA==", 37 "_self": "dbs/prUNAA=/colls/prUNAKtPjRo=/docs/prUNAKtPjRoFAAAAAAAAAA==/", 38 "_etag": "\"22007fea-0000-0000-0000-5b86dbd60000\"", 39 "_attachments": "attachments/", 40 "_ts": 1535564758 41 }</pre>

Load more

SQL API

Scale & Settings Documents Documents

New Document Update Discard Delete

SELECT \* FROM c Edit Filter


id

id
1
2


Load more

```
1 {
2   "id": "2",
3   "name": "Fortnite vs Zombies",
4   "lastCompetitionDate": "2018-09-30T03:31:20.7251173Z",
5   "tags": [
6     "3D",
7     "battle royale",
8     "monsters",
9     "shooter"
10  ],
11  "platforms": [
12    "PS4",
13    "XBox",
14    "PC",
15    "Switch",
16    "iPad",
17    "iPhone",
18    "Android"
19  ],
}
```

0 0 6

 Open Full Screen

 New SQL Query

 New Stored Procedure

## Open Full Screen

Open this database account in a new browser tab with Cosmos DB Explorer. Or copy the read-write or read only access urls below to share with others. For security purposes, the URLs grant time-bound access to the account. When access expires, you can reconnect, using a valid connection string for the account.

Read-Write

Read

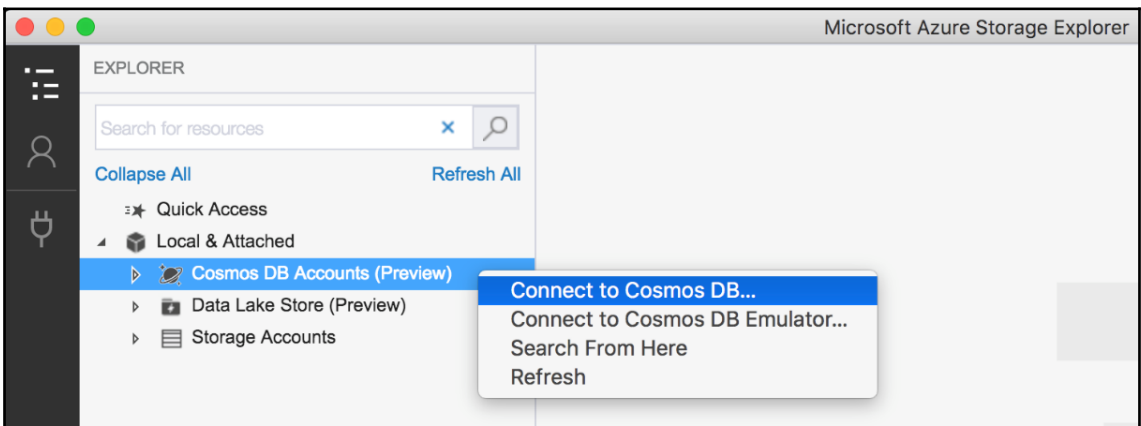
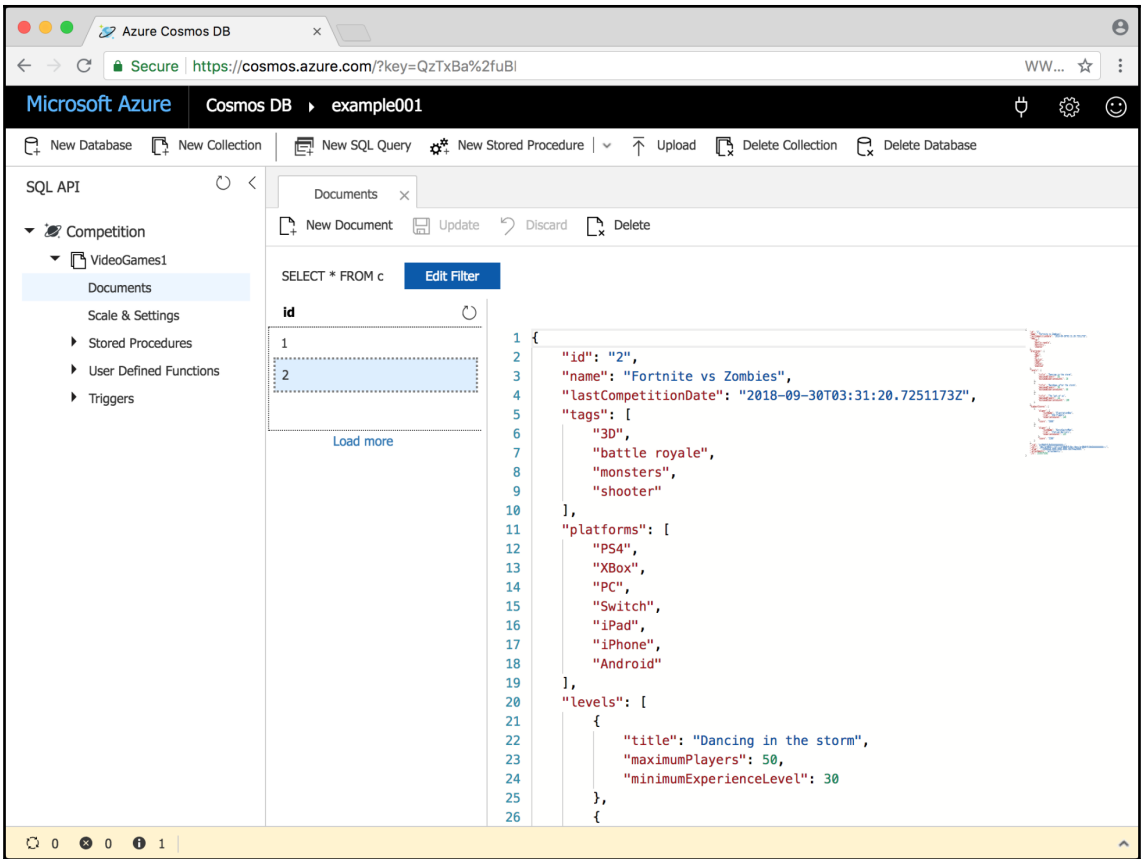
<https://cosmos.azure.com/?key=QzTxBa%2fuBLzu3...>



Open

Cancel

4:36



---

## Connect to Cosmos DB

Select API

SQL

Connection string

AccountEndpoint=https://example001.documents.azure.com:443/;AccountKey=Yf36iYUZvcv5V

Account label

example001

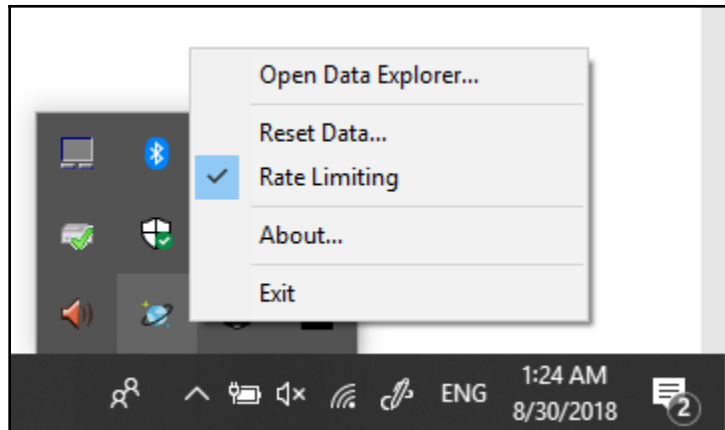
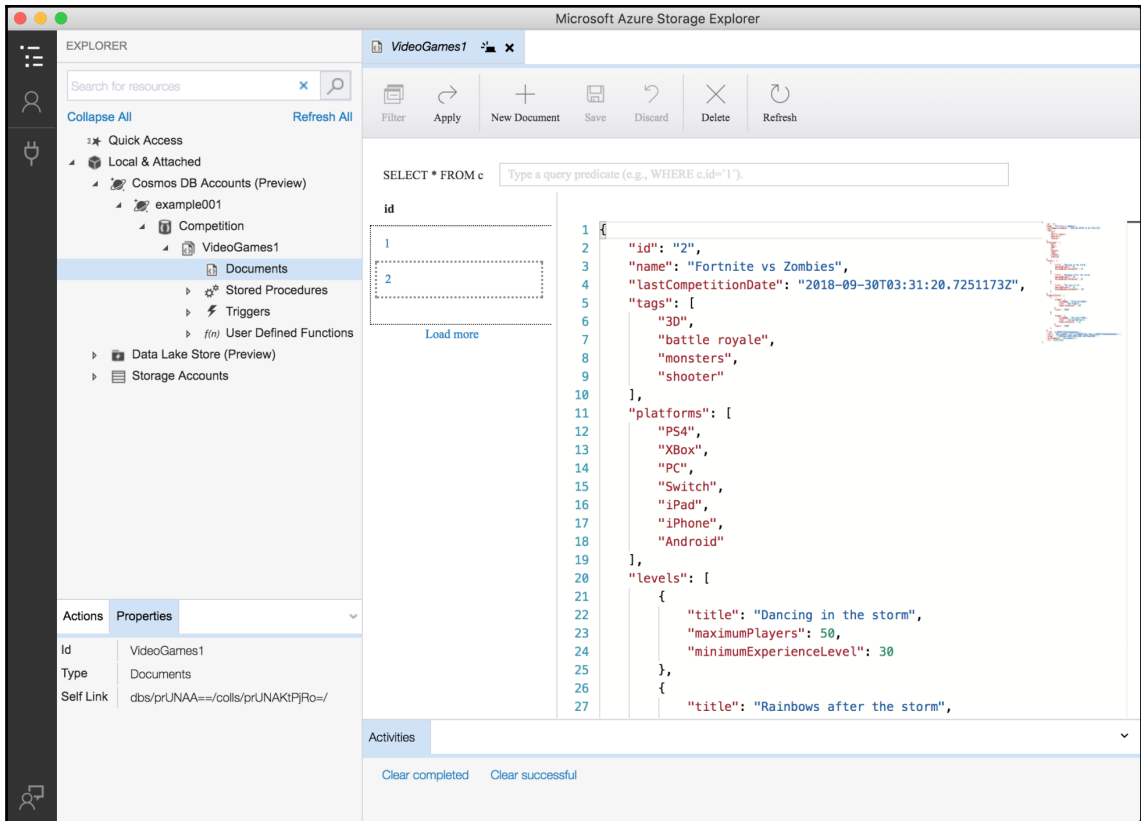
Back

Next

Ok

Cancel





Azure Cosmos DB Emulator

CREATE AN AZURE COSMOS DB ACCOUNT >

Quickstart

Congratulations! Your Azure Cosmos DB emulator is running.

Now, let's connect a sample app to it.

URI

`https://localhost:8081`

Primary Key

`C2y6yDjf5/R+ob0N8A7Cgv30VRDJIWEHLM+4QDU5DE2nQ9nDuVTqobD4b8mGGyPMbIZnqyMsEcaGQy67Xlw/Jw==`

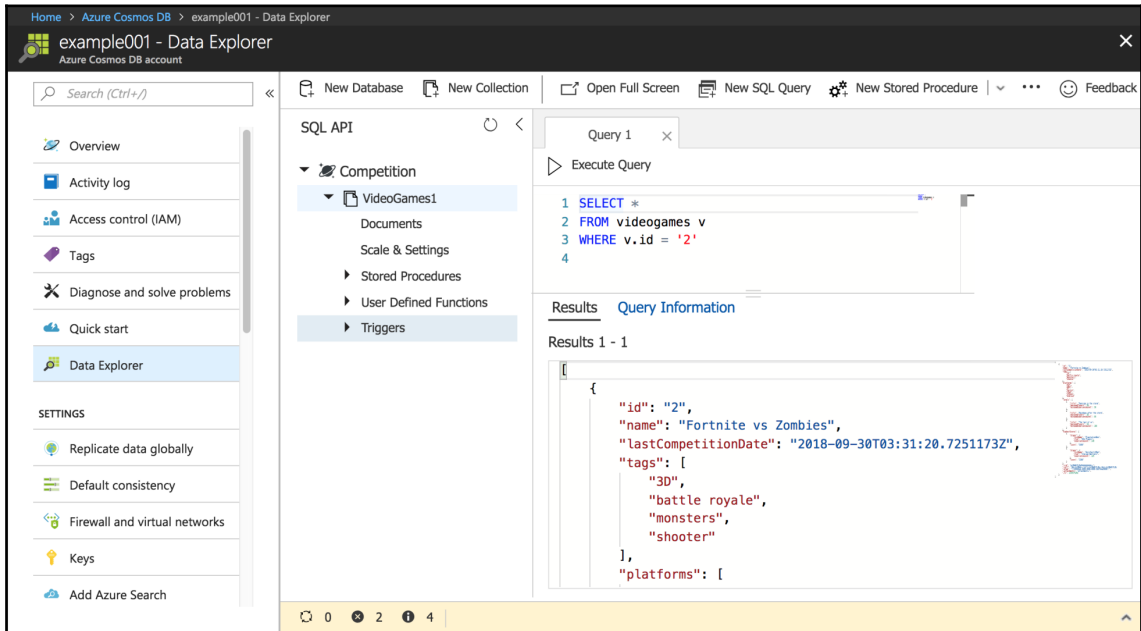
Primary Connection String

`AccountEndpoint=https://localhost:8081;AccountKey=C2y6yDjf5/R+ob0N8A7Cgv30VRDJIWEHLM+4QDU5DE2nQ9nDuVTqobD4b8mGGyPMbIZnqyMsEcaGQy67Xlw/Jw==`

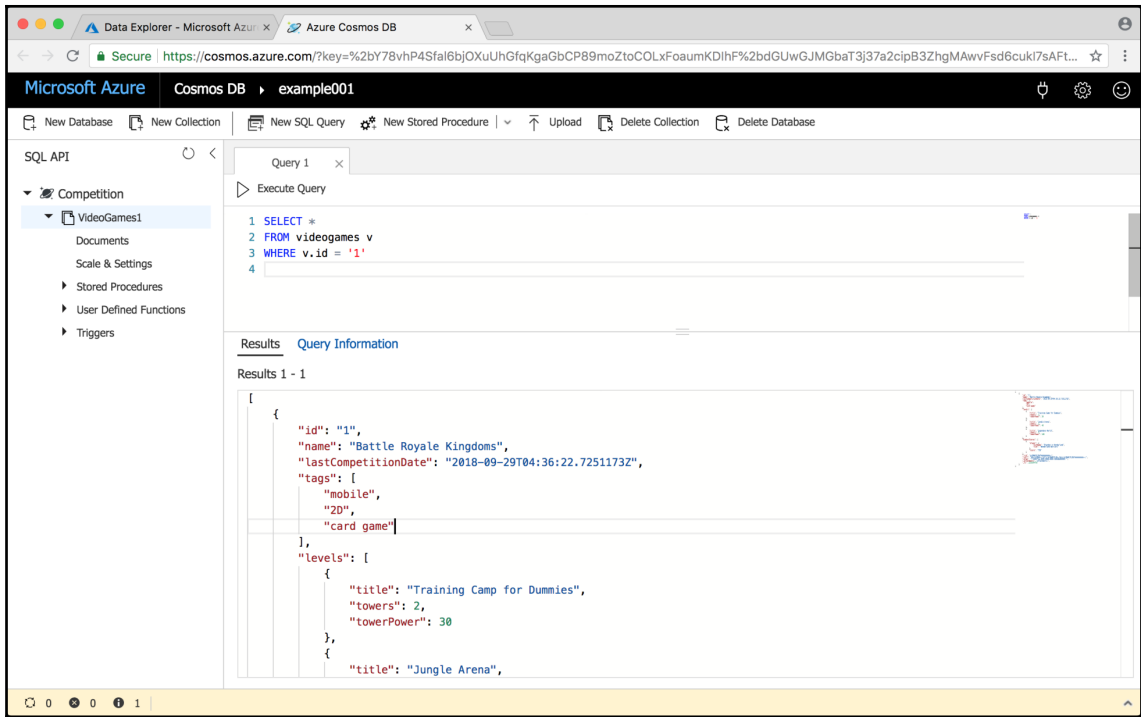
Mongo Connection String

`mongodb://localhost:C2y6yDjf5/R+ob0N8A7Cgv30VRDJIWEHLM+4QDU5DE2nQ9nDuVTqobD4b8mGGyPMbIZnqyMsEcaGQy67Xlw/Jw==@localhost:10255/admin?ssl=true`

# Chapter 3: Writing and Running Queries on NoSQL Document Databases



METRIC	VALUE
Request Charge	2.35 RUs
Showing Results	1 - 1
Activity id	13881475-a0c6-40cc-a413-a7f24914e11d



<a href="#">Results</a> <a href="#">Query Information</a>	
METRIC	VALUE
Request Charge	2.32 RUs
Showing Results	1 - 1
Activity id	c7f978c3-fd02-404b-8305-1bbe537385ef

Microsoft Azure Storage Explorer

EXPLORER

Search for resources

Collapse All Refresh All

- Quick Access
  - VideoGames1
    - Documents
    - Stored Procedures
    - Triggers
    - User Defined Functions
- Local & Attached
  - Cosmos DB Accounts (Preview)
    - example001
      - Competition
        - VideoGames1
          - Documents
          - Stored Procedures
          - Triggers
          - User Defined Function
  - Data Lake Store (Preview)

Actions Properties

- Open
- Open New Tab
- Open Query Tab

VideoGames1 x Query For VideoGames1 x

Execute Query

```
1 SELECT v.name, v.platforms
2 FROM Videogames v
3 ORDER BY v.name
4
```

Results Query Information

Results 1 - 2

```
[
  {
    "name": "Battle Royale Kingdoms"
  },
  {
    "name": "Fortnite vs Zombies",
    "platforms": [
      "PS4",
      "XBox",
      "PC",
      "Switch",
      "iPad",
      "iPhone",
      "Android"
    ]
  }
]
```

Activities

---

**Results**      Query Information

**METRIC**

**VALUE**

Request Charge

3.28 RUs

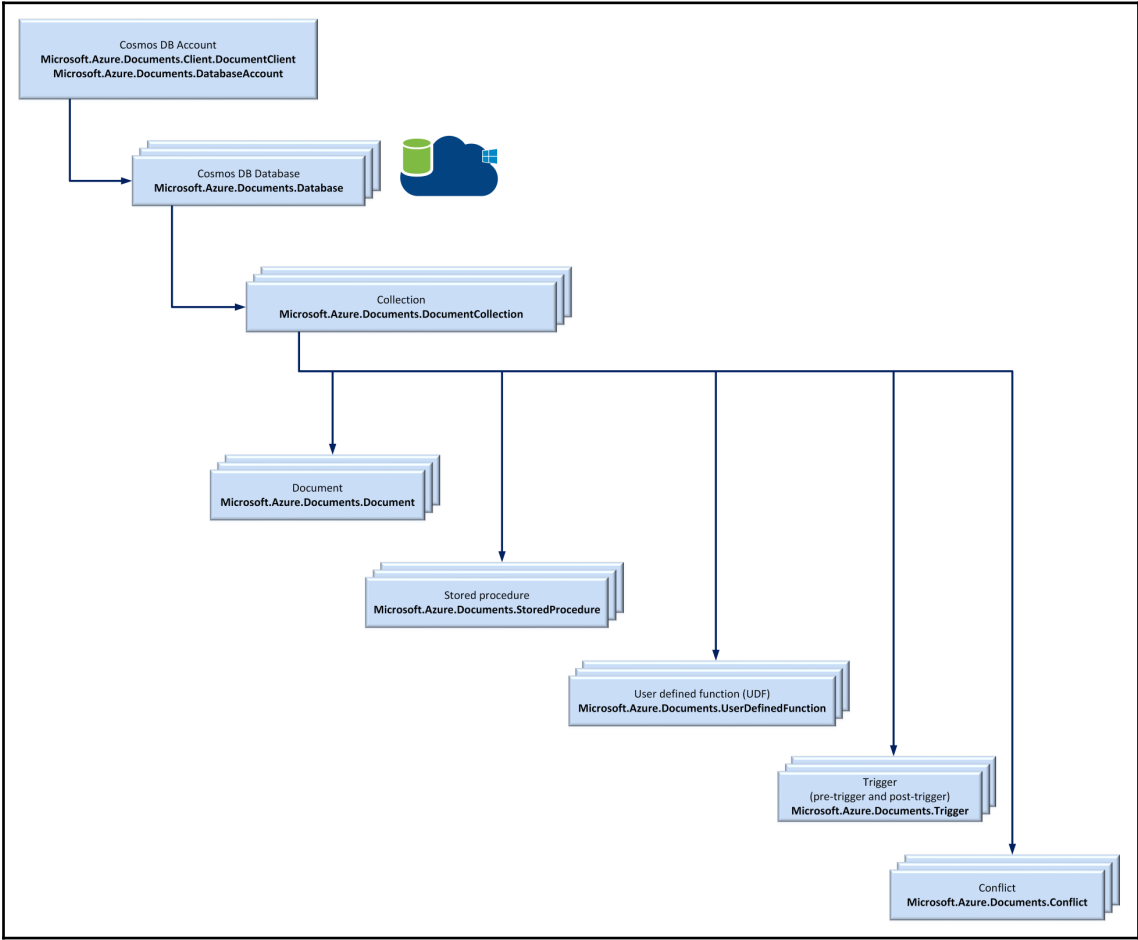
Showing Results

1 - 2

Activity id

a217f31e-4b92-4d80-8c49-a68322af66e7

# Chapter 4: Building an Application with C#, Cosmos DB, a NoSQL Document Database, and the SQL API



EXPLORER

Search for resources

Collapse All Refresh All

- Quick Access
- Local & Attached
  - Cosmos DB Accounts (Preview)
    - Emulator
    - example001
      - Competition
        - Competitions1
        - Documents
        - Stored Procedures
        - Triggers
        - User Defined Functions
        - VideoGames1
      - Data Lake Store (Preview)
      - Storage Accounts

Actions Properties

Id	Competitions1
Type	Documents
Self Link	dbs/prUNAA==/colls/prUNALhv7LU=/

Competitions1

Filter Apply New Document Save Discard Delete Refresh

id	/location/zipCode
1	90210
2	92075

Load more

```

1 {
2   "id": "2",
3   "title": "Defenders of the crown - San Diego 2018",
4   "location": {
5     "zipCode": "92075",
6     "state": "CA"
7   },
8   "platforms": [
9     "pc",
10    "PS4",
11    "XBox"
12  ],
13  "games": [
14    "Madden NFL 19",
15    "Fortnite"
16  ],
17  "numberOfRegisteredCompetitors": 250,
18  "status": "Scheduled",
19  "dateTime": "2018-11-10T04:00:56.0963801Z",
20  "_rid": "prUNALhv7LUgAAAAAAAAA==",
21  "_self": "dbs/prUNAA==/colls/prUNALhv7LU=/docs/prUNALhv7LUgAAAAAAAAA==/",
22  "_etag": "\2700c917-0000-0000-0000-5b973df70000\"",
23  "_attachments": "attachments/",
24  "_ts": 1536638455
25 }

```



---

# Chapter 5: Working with POCOs, LINQ, and a NoSQL Document Database

```
private static async Task<Competition> GetCompetitionByTitleWithLinq(string title)
{
    // Build a query to retrieve a Competition with a specific title
    var documentQuery = client.CreateDocumentQuery<Competition>(collectionUri,
        new FeedOptions()
        {
            EnableCrossPartitionQuery = true,
            MaxItemCount = 1,
        })
        .Where(c => c.Title == title)
        .Select(c => c)
        .AsDocumentQuery();
    while (documentQuery.HasMoreResults)
    {
        foreach (var competition in await documentQuery.ExecuteNextAsync<Competition>())
    }
}
```

Name	Value	Type
title	"League of legends - San Diego 2018"	string
documentQuery	{["query":"SELECT VALUE root FROM root WHERE (root['title'] = \"League of legends - San Diego 2018\") "]}]	Microsoft.Azure.Documents.Linq.IDocumentQuery<SampleApp1.Models.Competition>



---

# Chapter 6: Tuning and Managing Scalability with Cosmos DB

The screenshot shows the 'Scale & Settings' configuration page for a Cosmos DB database. On the left, a navigation pane under 'SQL API' lists 'Competition' with sub-items: 'VideoGames1', 'Competitions1', 'Documents', 'Scale & Settings' (highlighted), 'Stored Procedures', 'User Defined Functions', and 'Triggers'. The main panel has a title bar 'Scale & Setti...' and buttons for 'Save' and 'Discard'. It is divided into two sections: 'Scale' and 'Settings'. The 'Scale' section shows 'Storage capacity' as 'Unlimited' and 'Throughput (400 - unlimited RU/s)' with a text input set to '1000' and minus/plus controls. Below this, it states 'Estimated spend (USD): \$0.080 hourly / \$1.92 daily.' The 'Settings' section includes 'Time to Live' with radio buttons for 'Off' (selected), 'On (no default)', and 'On'. The 'Partition key' is set to '/location.zipCode' in a text box. The 'Indexing Policy' section is partially visible at the bottom.

---



Scale & Setti... x

 Save  Discard

▼ Scale

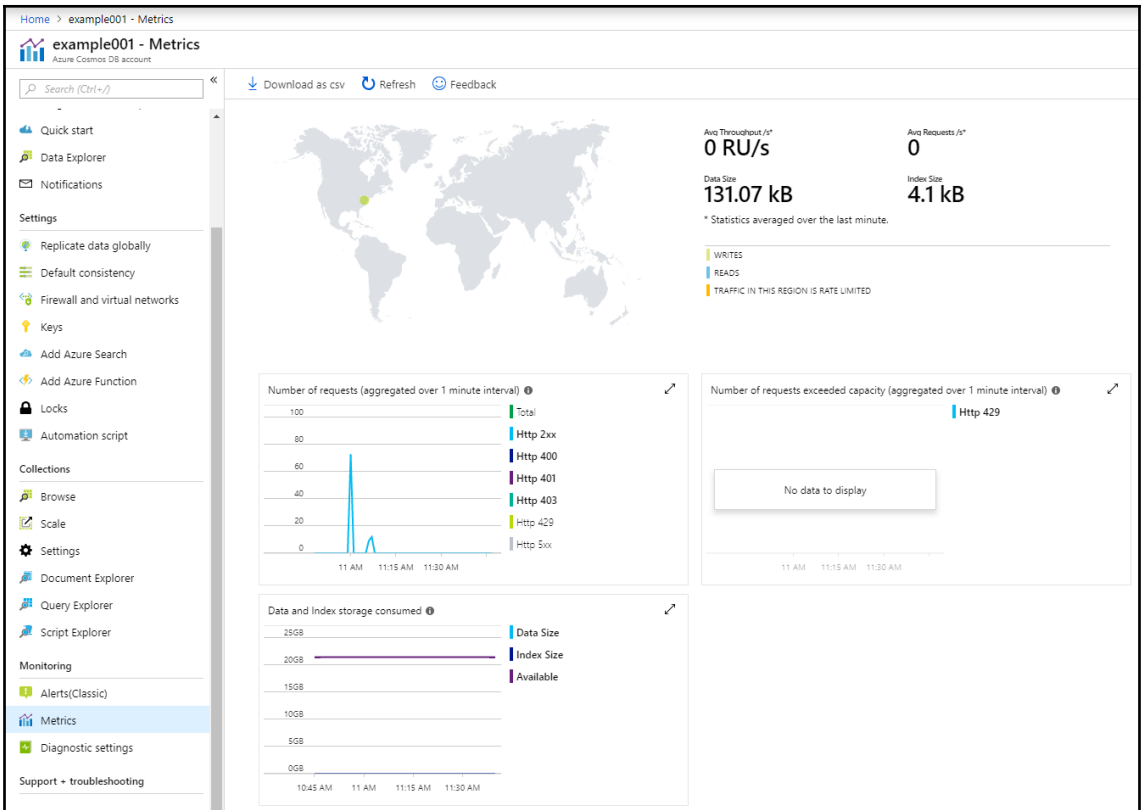
Storage capacity  
**Unlimited**

Throughput (400 - unlimited RU/s)

Estimated spend (USD): **\$0.16 hourly / \$3.84 daily.**

---



Program.cs

```

    });
    readCompetitionResponse.Document.DateTime = newDateTime;
}

```

Locals

Name	Value	Type
readCompetitionResponse	(Microsoft.Azure.Documents.Client.DocumentResponse < SampleApp1.Models.Competition >)	Microsoft.Azure.Documents.Client.DocumentResponse < SampleApp1.Models.Competition >
ActivityId	"bc997b33-44da-484a-8ade-315c127d9d63"	string
CollectionQuota	0	long
CollectionSizeQuota	10485760	long
CollectionSizeUsage	66	long
CollectionUsage	0	long
ContentLocation	"dbs/Competition/colls/Competitions1"	string
CurrentResourceQuotaUsage	"documentSize=0;documentsSize=65;documentsCount=4;collectionSize=66;"	string
DatabaseQuota	0	long
DatabaseUsage	0	long
Document	{["id": "4", "_rid": "prUNAJOG6ywwEAAAAAAAAA==", "_self": "dbs/prUNAA==/colls/prUNAJOG6yww SampleApp1.Models.Competition	
DocumentQuota	10485760	long
DocumentUsage	65	long
IndexTransformationProgress	-1	long
IsRUPerMinuteUsed	false	bool
LazyIndexingProgress	-1	long
MaxResourceQuota	"documentSize=10240;documentsSize=10485760;documentsCount=-1;collectionSize=10485760;"	string
PermissionQuota	0	long
PermissionUsage	0	long
RequestCharge	1	double
RequestDiagnosticsString	--	string
RequestLatency	(00:00:00)	System.TimeSpan
ResponseHeaders	(System.Collections.Specialized.NameValueCollection)	System.Collections.Specialized.NameValueCollection
ResponseStream	(System.IO.MemoryStream)	System.IO.Stream (System.IO.MemoryStream)
SessionToken	"033"	string
Status Code	OK	System.Net.HttpStatusCode
StoredProceduresQuota	0	long
StoredProceduresUsage	0	long
TriggersQuota	0	long
TriggersUsage	0	long
UserDefinedFunctionsQuota	0	long
UserDefinedFunctionsUsage	0	long
UserQuota	0	long

Error List | Locals | Call Stack | Watch 1 | Exception Settings | Immediate Window

Program.cs

```

    if (updatedCompetitionResponse.StatusCode == System.Net.HttpStatusCode.OK)
    {

```

Locals

Name	Value	Type
updatedCompetitionResponse	(Microsoft.Azure.Documents.Client.ResourceResponse < Microsoft.Azure.Documents.Document >)	Microsoft.Azure.Documents.Client.ResourceResponse < Microsoft.Azure.Documents.Document >
ActivityId	"1377afbb-af8e-4b74-97df-ac742e9c8f9c"	string
CollectionQuota	0	long
CollectionSizeQuota	10485760	long
CollectionSizeUsage	66	long
CollectionUsage	0	long
ContentLocation	"dbs/Competition/colls/Competitions1"	string
CurrentResourceQuotaUsage	"documentSize=0;documentsSize=65;documentsCount=4;collectionSize=66;"	string
DatabaseQuota	0	long
DatabaseUsage	0	long
DocumentQuota	10485760	long
DocumentUsage	65	long
IndexTransformationProgress	-1	long
IsRUPerMinuteUsed	false	bool
LazyIndexingProgress	-1	long
MaxResourceQuota	"documentSize=10240;documentsSize=10485760;documentsCount=-1;collectionSize=10485760;"	string
PermissionQuota	0	long
PermissionUsage	0	long
RequestCharge	13.26	double
RequestDiagnosticsString	--	string
RequestLatency	(00:00:00)	System.TimeSpan
Resource	{["title": "League of legends - San Diego 2019", "location": { "zipCode": "92075", "state": "CA" } ]}	Microsoft.Azure.Documents.Document
ResponseHeaders	(System.Collections.Specialized.NameValueCollection)	System.Collections.Specialized.NameValueCollection
ResponseStream	(System.IO.MemoryStream)	System.IO.Stream (System.IO.MemoryStream)
SessionToken	"034"	string
Status Code	OK	System.Net.HttpStatusCode
StoredProceduresQuota	0	long
StoredProceduresUsage	0	long
TriggersQuota	0	long
TriggersUsage	0	long
UserDefinedFunctionsQuota	0	long
UserDefinedFunctionsUsage	0	long
UserQuota	0	long

Error List | Locals | Call Stack | Watch 1 | Exception Settings | Immediate Window

Program.cs

```

var feedResponse = await documentQuery.ExecuteNextAsync<Competition>();
foreach (var competition in feedResponse)
{
    Console.WriteLine(competition);
}

```

Locals

Name	Value	Type
title	"League of legends - San Diego 2018"	string
documentQuery	[[{"query":"SELECT VALUE root FROM root WHERE (root['title'] = \"League of legends - San Diego 2018\")"}]]	Microsoft.Azure.Documents.Linq.IDocumentQuery<SampleApp1.Models.Competition> [Microsoft.Azure.Documents.Li
feedResponse	[[{"Microsoft.Azure.Documents.Client.FeedResponse - SampleApp1.Models.Competition"}]]	Microsoft.Azure.Documents.Client.FeedResponse<SampleApp1.Models.Competition>
ActivityId	"ac3187d6-5363-4b14-ad21-373da947c19d"	string
CollectionQuota	0	long
CollectionSizeQuota	10485760	long
CollectionSizeUsage	66	long
CollectionUsage	0	long
ContentLocation	"dbs/Competition/colls/Competitions1"	string
Count	1	int
CurrentResourceQuotaUsage	"documentSize=0;documentsSize=65;documentsCount=4;collectionSize=66;"	string
DatabaseQuota	0	long
DatabaseUsage	0	long
ETag	null	string
ISRUPerMinuteUsed	false	bool
MaxResourceQuota	"documentSize=10240;documentsSize=10485760;documentsCount=1;collectionSi"	string
PermissionQuota	0	long
PermissionUsage	0	long
QueryMetrics	null	System.Collections.Generic.IReadOnlyDictionary<string, Microsoft.Azure.Documents.QueryMetrics>
RequestCharge	0	double
ResponseContinuation	null	string
ResponseHeaders	{System.Collections.Specialized.NameValueCollection}	System.Collections.Specialized.NameValueCollection
SessionToken	"038"	string
StoredProceduresQuota	0	long
StoredProceduresUsage	0	long
TriggersQuota	0	long
TriggersUsage	0	long
UserDefinedFunctionsQuota	0	long
UserDefinedFunctionsUsage	0	long
UserQuota	0	long
UserUsage	0	long

Error List | Locals | Call Stack | Watch 1 | Exception Settings | Immediate Window



Container (10K RU/s)  
Collection / Graph / Table

Partition-key Range (5K RU/s)  
Beef Product, Vegetables, Baked Products  
*Physical Partition*

Partition-key Value: Beef Product  
*Logical Partition*

Partition-key Value: Vegetables  
*Logical Partition*

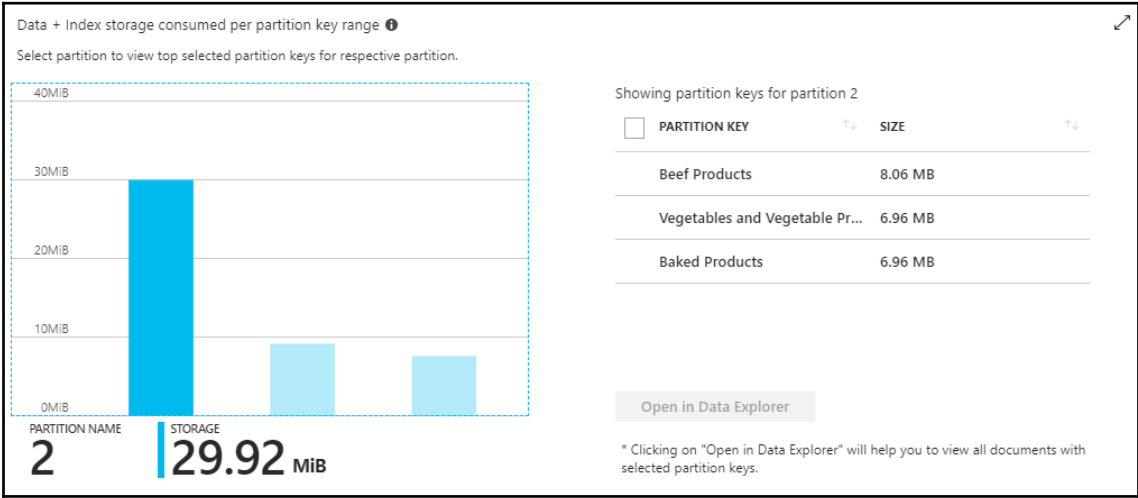
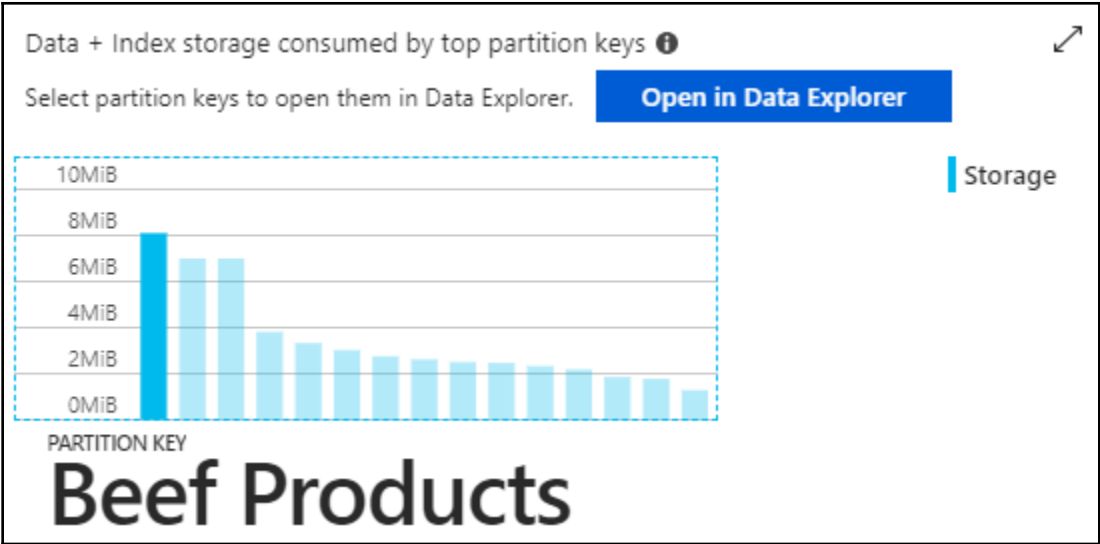
Partition-key Value: Baked Products  
*Logical Partition*

Partition-key Range (5K RU/s)  
Legumes, Pork, Finfish  
*Physical Partition*

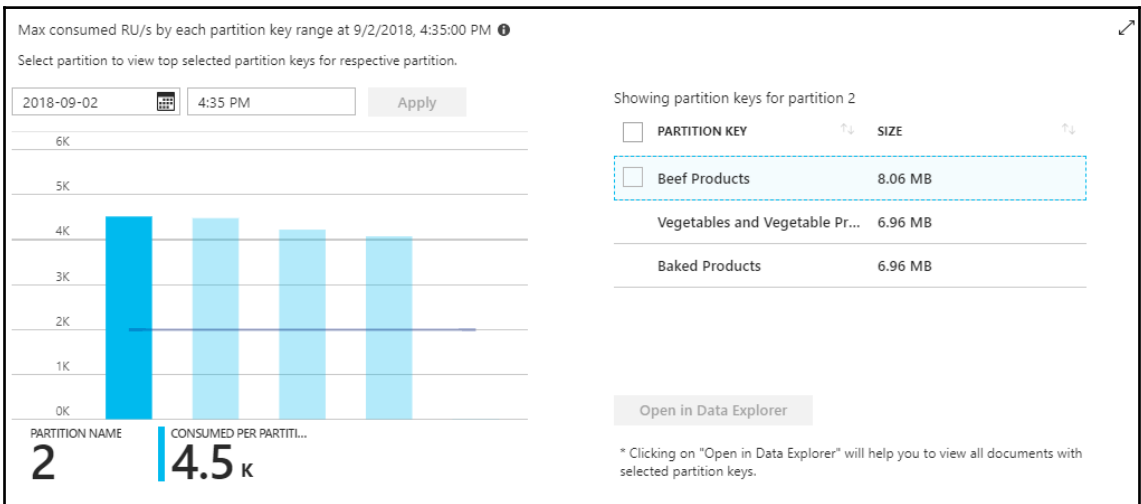
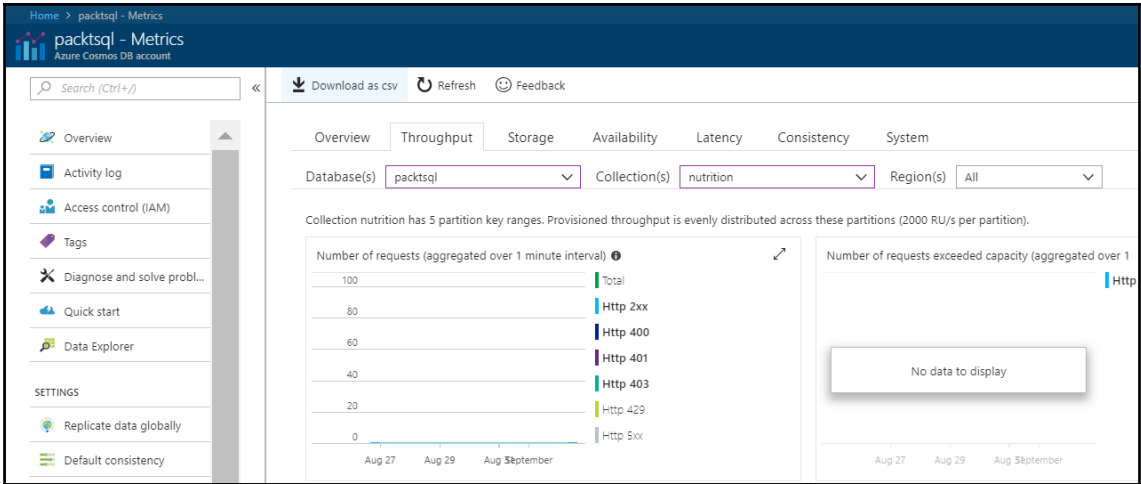
Partition-key Value: Legumes  
*Logical Partition*

Partition-key Value: Pork  
*Logical Partition*

Partition-key Value: Finfish  
*Logical Partition*







Home > Resource groups > cosmos-db-packt > packtsql - Replicate data globally

## packtsql - Replicate data globally

Azure Cosmos DB account

Search (Ctrl+/) Save Discard Manual Failover Automatic Failover


- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Quick start
- Data Explorer

SETTINGS

- Replicate data globally**
- Default consistency
- Firewall and virtual networks
- Keys
- Add Azure Search
- Add Azure Function
- Locks
- Automation script

COLLECTIONS







- Browse
- Scale



**WRITE REGION**

West Europe

**READ REGIONS**

- North Europe 
- Southeast Asia 
- West US 
- Brazil South 
- Australia East 
- Japan East 

Home > Resource groups > cosmos-db-packt > packtsql - Default consistency


packtsql - Default consistency  
Azure Cosmos DB account

Search (Ctrl+/) Save Discard

STRONG BOUNDED STALENESS **SESSION** CONSISTENT PREFIX EVENTUAL

**i** Session consistency is most widely used consistency level both for single region as well as, globally distributed applications.

**Understand Session consistency**  
It provides write latencies, availability and read throughput comparable to that of eventual consistency but also provides the consistency guarantees that suit the needs of



The image shows a screenshot of the Azure portal interface for configuring the default consistency level of an Azure Cosmos DB account. The breadcrumb trail at the top indicates the path: Home > Resource groups > cosmos-db-packt > packtsql - Default consistency. The page title is 'packtsql - Default consistency' and it identifies the account as 'Azure Cosmos DB account'. On the left, there is a navigation pane with sections for 'Overview' (including Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, Data Explorer), 'SETTINGS' (including Replicate data globally, Default consistency, Firewall and virtual networks, Keys, Add Azure Search, Add Azure Function, Locks, Automation script), and 'COLLECTIONS' (including Browse). The 'Default consistency' option is currently selected. At the top of the main content area, there are tabs for consistency levels: STRONG, BOUNDED STALENESS, SESSION (which is highlighted), CONSISTENT PREFIX, and EVENTUAL. Below the tabs, an information icon (i) is followed by the text: 'Session consistency is most widely used consistency level both for single region as well as, globally distributed applications.' Underneath, a section titled 'Understand Session consistency' explains that it provides write latencies, availability, and read throughput comparable to eventual consistency, but also offers consistency guarantees suitable for specific needs. At the bottom of this section is a world map with several colored dots (blue, green, grey) scattered across different continents, representing the global distribution of data centers.

## Automatic Failover




Enable Automatic Failover ⓘ

ON

OFF

Drag-and-drop read regions items to reorder the failover priorities.

Tip: Drag  on the left of the hovered row to reorder the list.

### WRITE REGION

West Europe

READ REGIONS	PRIORITIES
Southeast Asia	1
North Europe	2
West US	3
Brazil South	4
Australia East	5
Japan East	6