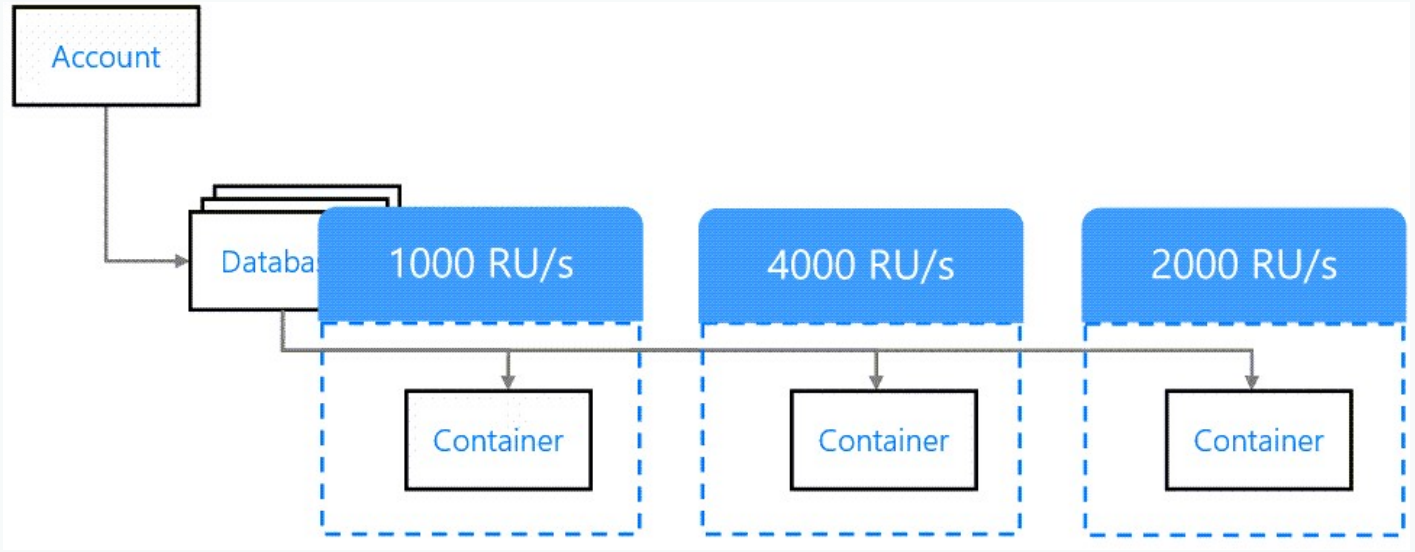
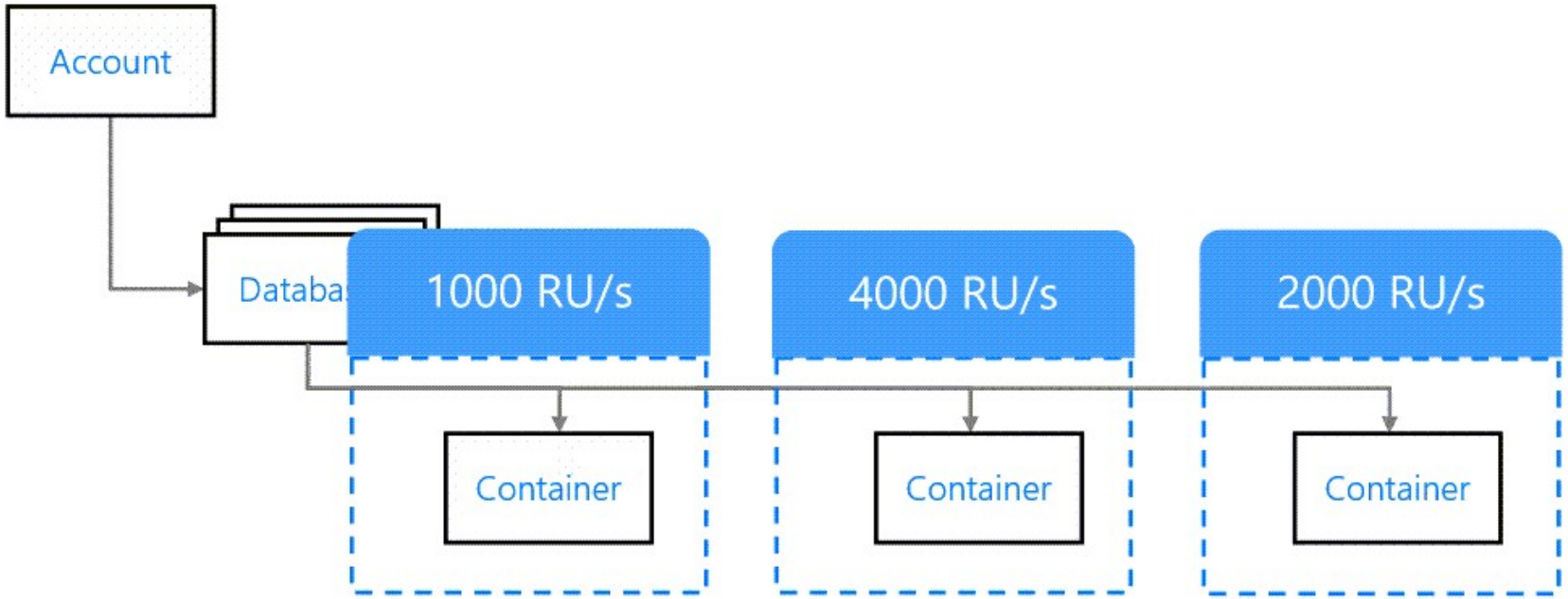
Cosmos DB - Plan Resource Requirements

What is throughput

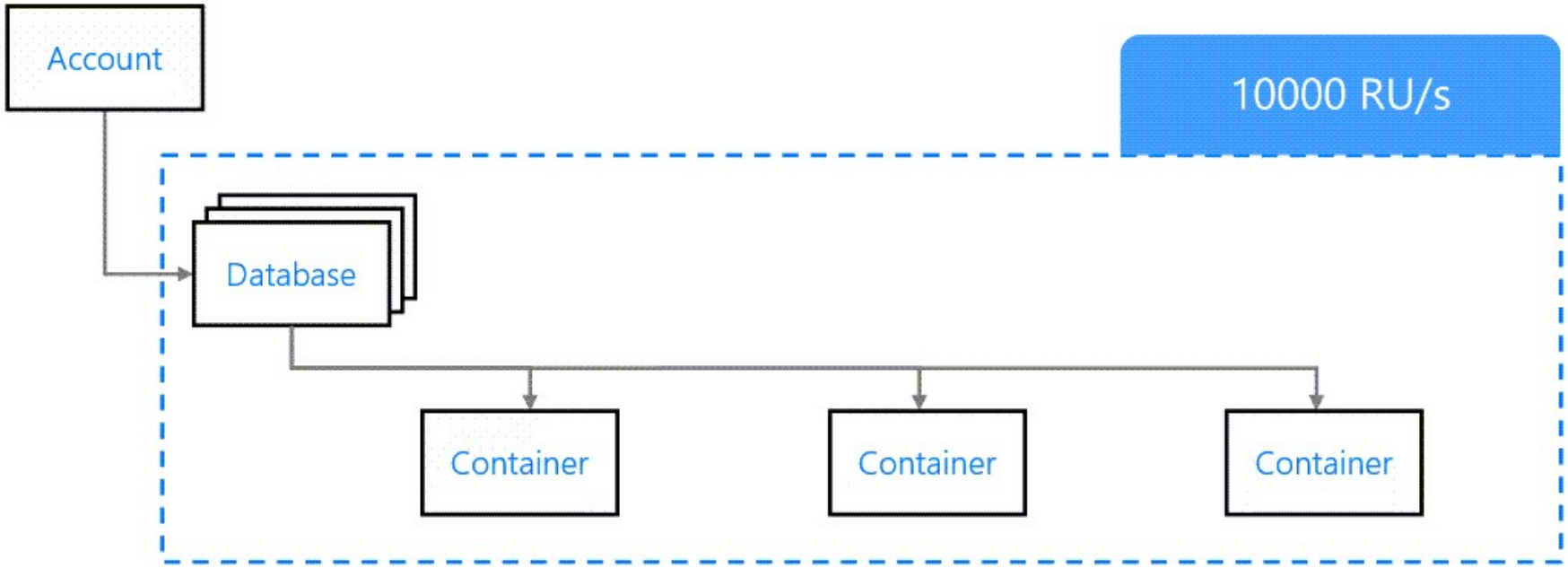
- Each container is a unit of scalability for both throughput and storage.
- Can provision throughput at
  - Database levels
  - Container levels



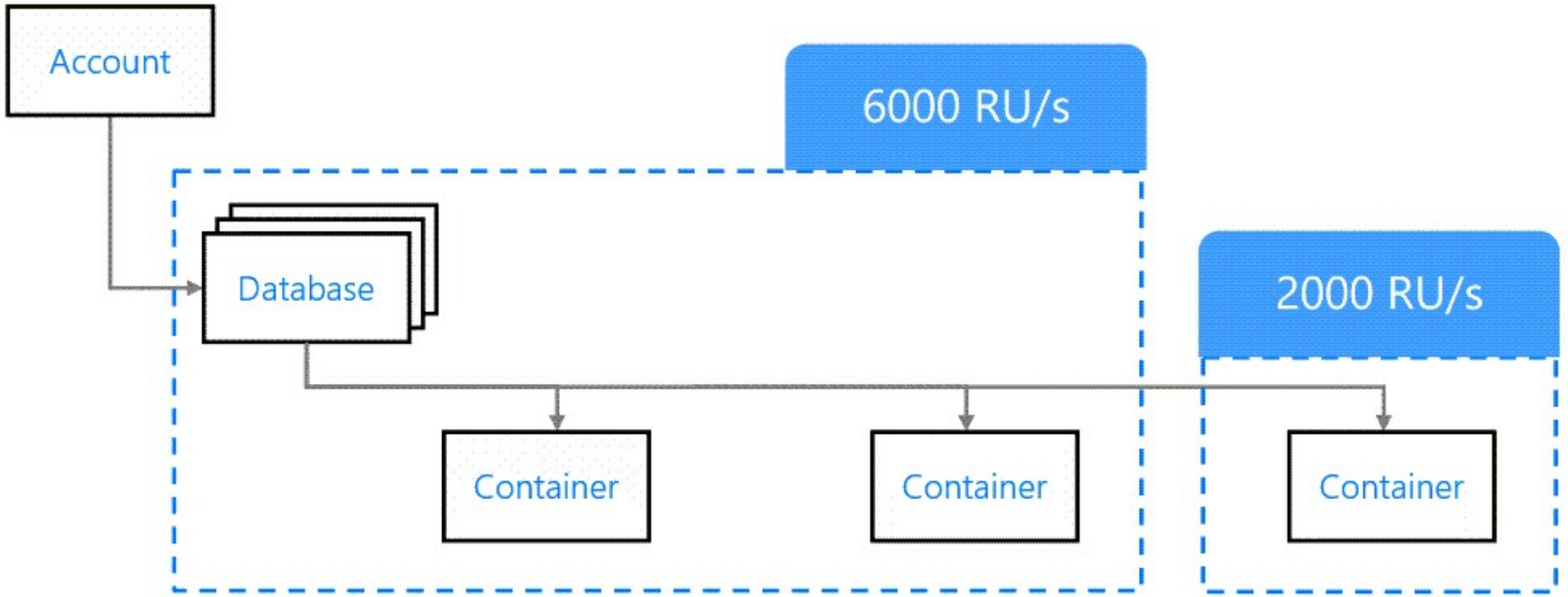
Container-level throughput provisioning



Database-level throughput provisioning



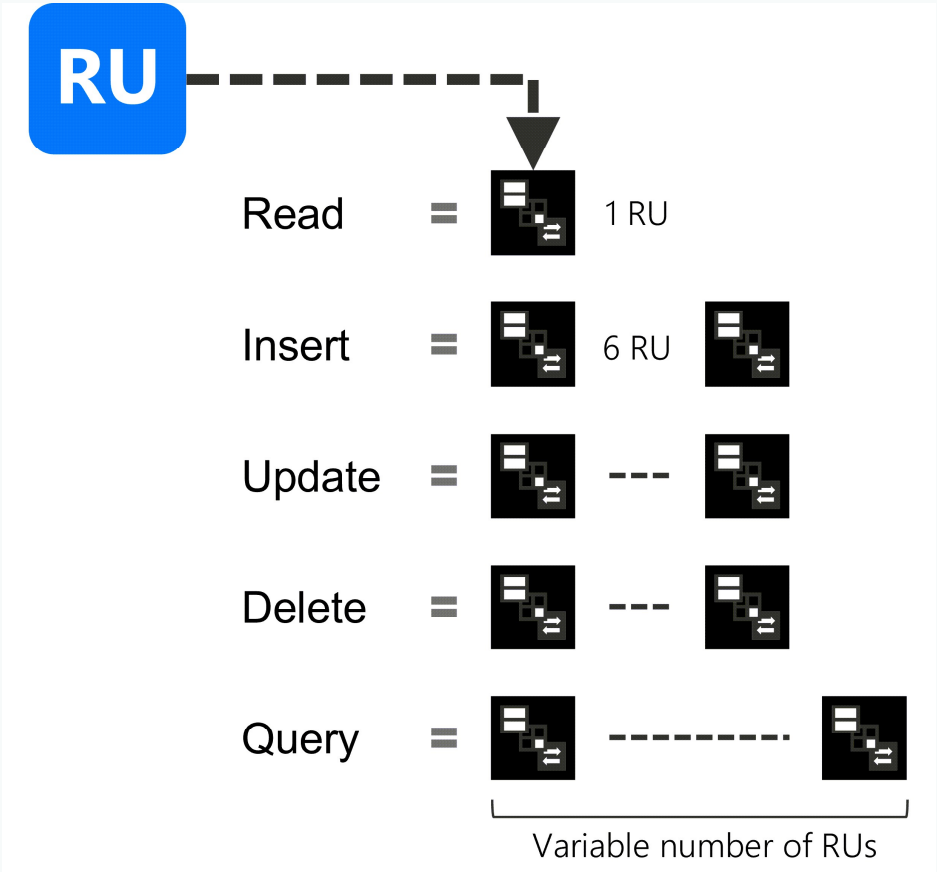
Mixed-throughput provisioning



Evaluate throughput requirements

- Request units are a rate-based currency.
- Every request consumes a fixed number of request units, including but not limited to:
  - Reads
  - Writes
  - Queries
  - Stored procedures

Estimating ad-hoc RU/s consumption



Estimating ad-hoc RU/s consumption

- You can use a spreadsheet application to build a quick table to figure out a rough estimate of your needed request unit capacity
-

Operation type	Number of requests per second	Number of RU per request	RU/s needed
Write Single Document	10,000	10	100,000
Top Query #1	700	100	70,000
Top Query #2	200	100	20,000
Top Query #3	100	100	10,000
Total RU/s			200,000 RU/s

Evaluate data storage requirements

- Use **Azure Cosmos DB Capacity Calculator**
  - https://cosmos.azure.com/capacitycalculator/
- The calculator will inquire about details such as:
  - Total data stored
  - Whether you expect to perform near real-time analytics
  - The anticipated size of documents
  - Point reads per second
  - Queries per second

TTL

- The length of time documents live in the database before being automatically purged.
- TTL value is defined as an integer in seconds.

Configuring TTL on a container

- Configured using the DefaultTimeToLive property of the container's JSON object.

DefaultTimeToLive	Expiration
Does not exist	Items are not automatically expired
`-1`	Items will not expire by default
n	*n* seconds after last modified time

Configuring TTL on a container

- The TTL value for an item is configured by setting the ttl path of the item
- The TTL value for an item will only work if the DefaultTimeToLive property is configured for the parent container
- If the ttl path is configured for the item, it will override the DefaultTimeToLive property of the parent container.

Configuring TTL on a container

Container.DefaultTimeToLive	Item.ttl	Expiration in seconds
`1000`	*null*	`1000`
`1000`	`-1`	*This item will never expire*
`1000`	`2000`	`2000`

Container.DefaultTimeToLive	Item.ttl	Expiration in seconds
null	*null*	*This item will never expire*
null	`-1`	*This item will never expire*
null	`2000`	*TTL is disabled at the container level. This item will never expire.*

Plan for data retention with TTL

- Azure Cosmos DB only charges for storage you directly consume in real time
- Don't have to pre-reserve storage in advance
- In high-write scenarios, TTL values can be used to save on data storage costs in Azure Cosmos DB.
- Consider solutions such to aggregate and migrate data such as:
  - Change feed
  - Azure Data Warehouse
  - Azure Blob Storage
- When designing your solution, plan how long your data will need to be retained in Azure Cosmos DB before being migrated across your entire Azure solution space to minimize storage costs.

Thank You