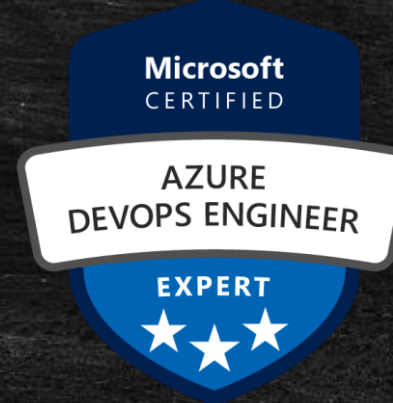
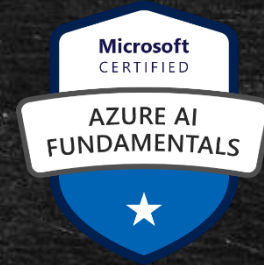
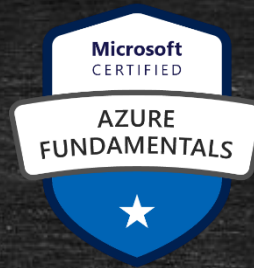


Azure Cosmos DB For NoSQL

Felipe Augusto, Microsoft MVP



Felipe Augusto



Introdução

O Azure Cosmos DB é o banco de dados NoSQL totalmente gerenciado da Microsoft no Azure no formato de PaaS (Platform as a Service). Como um banco de dados NoSQL, o Azure Cosmos DB é horizontalmente escalonável e não relacional.

- Os bancos de dados NoSQL foram projetados para atender a necessidades como:
 - Grandes volumes de dados.
 - Dados com muitas fontes e formas diferentes.
 - Esquemas de dados dinâmicos que armazenam diferentes tipos de dados.
 - Uso de dados em alta velocidade e/ou em tempo real.
- ACID
 - 1. Atomicidade; 2. Consistência; 3. Isolamento; 4. Durabilidade.



	Responsibility	SaaS	PaaS	IaaS	On-prem
Responsibility always retained by the customer	Information and data	Customer	Customer	Customer	Customer
	Devices (Mobile and PCs)	Customer	Customer	Customer	Customer
	Accounts and identities	Customer	Customer	Customer	Customer
Responsibility varies by type	Identity and directory infrastructure	Shared	Shared	Customer	Customer
	Applications	Microsoft	Shared	Customer	Customer
	Network controls	Microsoft	Shared	Customer	Customer
	Operating system	Microsoft	Microsoft	Customer	Customer
Responsibility transfers to cloud provider	Physical hosts	Microsoft	Microsoft	Microsoft	Customer
	Physical network	Microsoft	Microsoft	Microsoft	Customer
	Physical datacenter	Microsoft	Microsoft	Microsoft	Customer



Microsoft



Customer



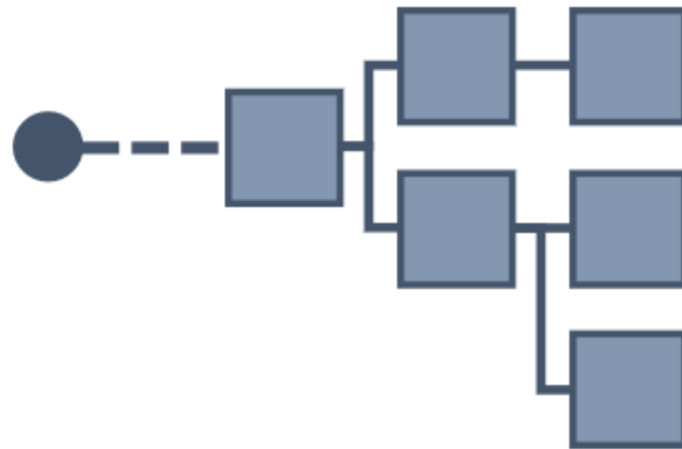
Shared



Key-Value



Documents



Column-Family



Graph



Escalabilidade

- Essas características incluem:

- Um armazenamento de dados não relacional.
- Ser projetado para escalar horizontalmente.
- Não impor um esquema específico.

- São projetados para escalar horizontalmente por meio de fragmentação ou particionamento, assim permitindo manter o alto desempenho, independentemente do tamanho.

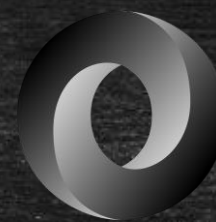
- Verticalmente



- Horizontal



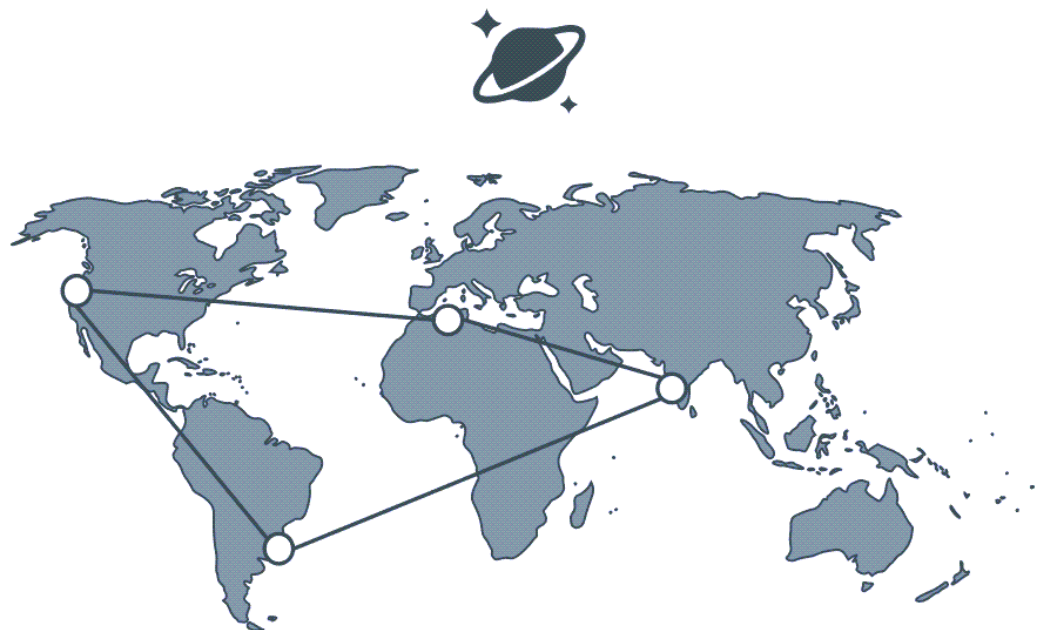
O que é um documento JSON



▪ JavaScript Object Notation (ou JSON) é um formato de dados leve. O JSON foi criado para ser altamente compatível com a notação literal de um objeto na linguagem JavaScript. Muitas estruturas, navegadores e até mesmo bancos de dados têm suporte para JavaScript, o que faz do JSON um formato popular para transmissão e armazenamento de dados.

```
1 {  
2   "name": "Felipe Augusto",  
3   "profession": "Technical Architecture Manager",  
4   "age": 40,  
5   "address": {  
6     "city": "Curitiba",  
7     "postalCode": 80740510,  
8     "Country": "br"  
9   },  
10  "languages": [".NET", "Java", "Node.js", "Rust"],  
11  "socialProfiles": [  
12    {  
13      "name": "LinkedIn",  
14      "link": "https://linkedin.com/in/felipementel"  
15    },  
16    {  
17      "name": "Youtube",  
18      "link": "https://www.youtube.com/@D.E.P.L.O.Y"  
19    }  
20  ]  
21 }
```


O que é o Azure Cosmos DB for NoSQL?



O Azure Cosmos DB for NoSQL é um serviço rápido de banco de dados NoSQL que oferece consultas avançadas sobre diversos dados.

Ele ajuda a fornecer desempenho configurável e confiável, é distribuído globalmente e permite o desenvolvimento rápido.

O modelo de dados de documento divide os dados em entidades de documento individuais. Um documento pode ter qualquer tipo de dados estruturado, mas normalmente o JSON é usado como formato de dados

O Azure Cosmos DB for NoSQL tem algumas vantagens, como:

- Velocidade garantida em qualquer escala, mesmo em momentos de intermitência, com elasticidade instantânea e ilimitada, leituras rápidas e gravações de vários mestres, em qualquer lugar do mundo.
- SDKs para linguagens populares, uma API de NoSQL nativa, bem como APIs para MongoDB, Cassandra e Gremlin, e análise sem ETL (extração, transformação, carregamento).
- Pronta para aplicativos críticos com alta disponibilidade, de 99,999% e segurança de nível empresarial.
- Serverless/em servidor totalmente gerenciado e econômico com dimensionamento automático instantâneo que responde às necessidades do aplicativo.

Para calcular a quantidade de minutos de indisponibilidade permitida em um sistema com 99,999% de disponibilidade (também conhecido como "cinco noves"), podemos seguir os passos abaixo:

1. Existem 525.600 minutos em um ano (365 dias x 24 horas x 60 minutos).
2. A indisponibilidade permitida é 0,001% (ou seja, 100% - 99,999%).

$$\text{Indisponibilidade (em minutos)} = 525.600 \times \frac{0,001}{100}$$

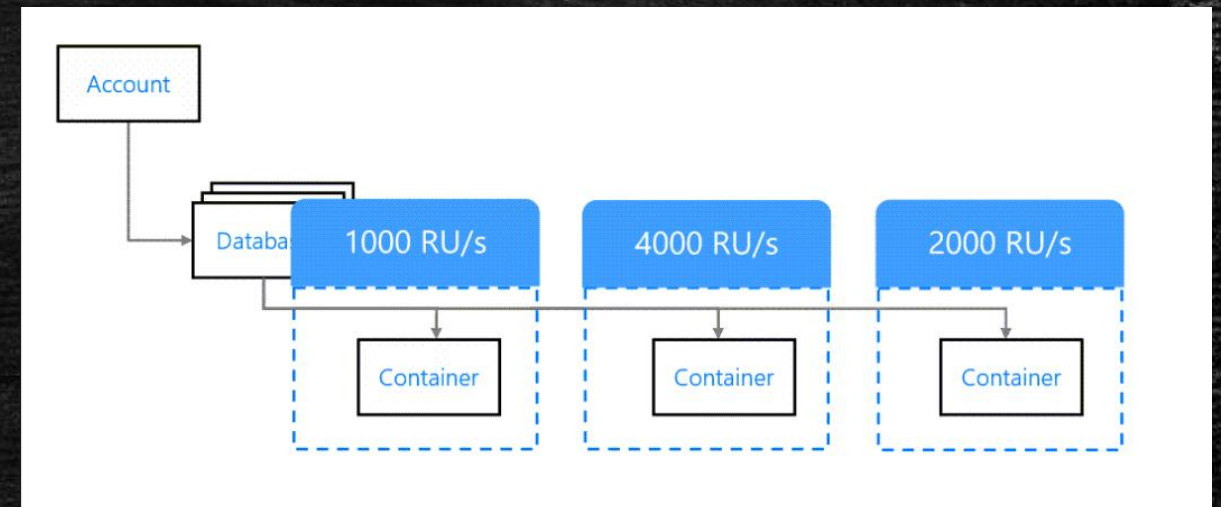
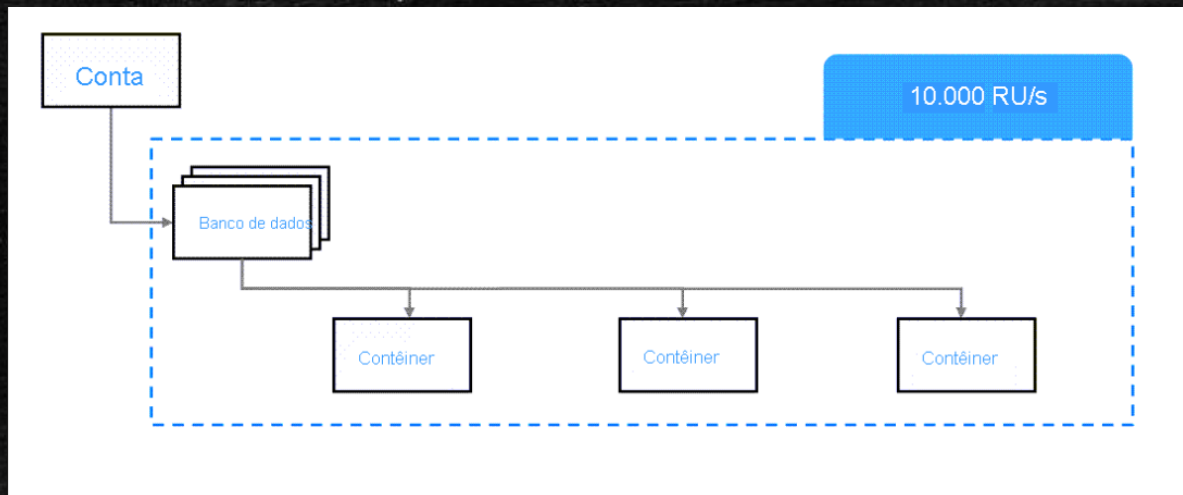
Isso dá aproximadamente **5,26 minutos de indisponibilidade por ano.**

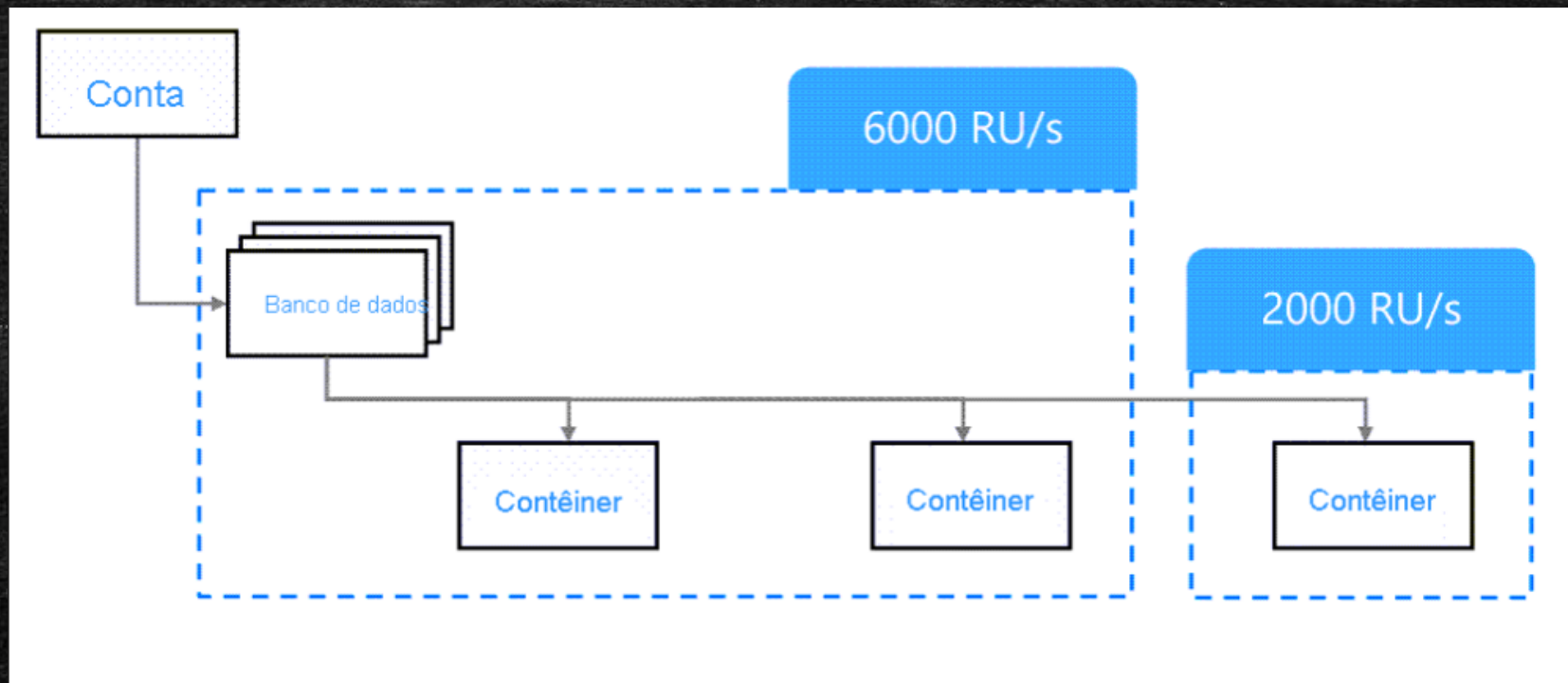
As contas são as unidades fundamentais de distribuição e alta disponibilidade.

Banco de dados é uma unidade lógica de gerenciamento para contêineres no Azure Cosmos DB for NoSQL. Um banco de dados do Azure Cosmos DB gerencia usuários, permissões e contêineres

Contêineres são a unidade fundamental de escalabilidade no Azure Cosmos DB for NoSQL. Normalmente, você provisiona a taxa de transferência no nível de contêiner

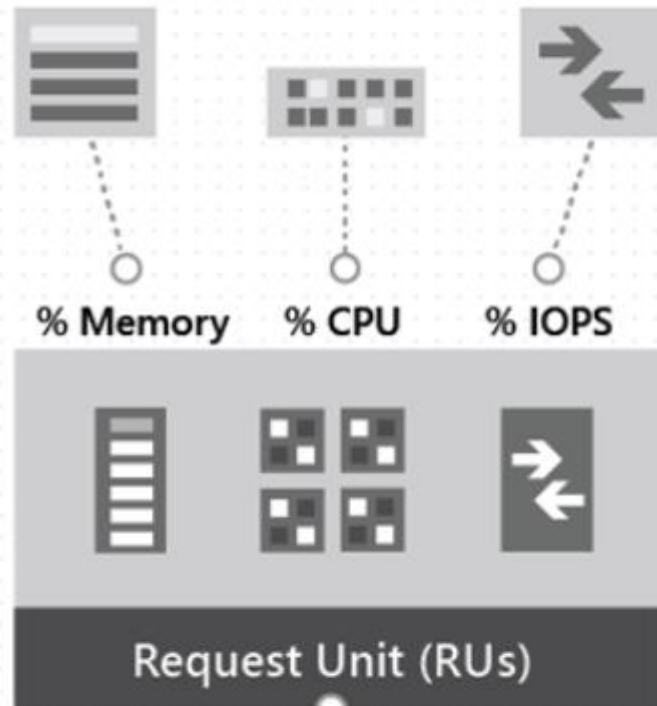
Ao configurar o Azure Cosmos DB, você pode provisionar a taxa de transferência nos níveis do banco de dados, do contêiner ou de ambos.



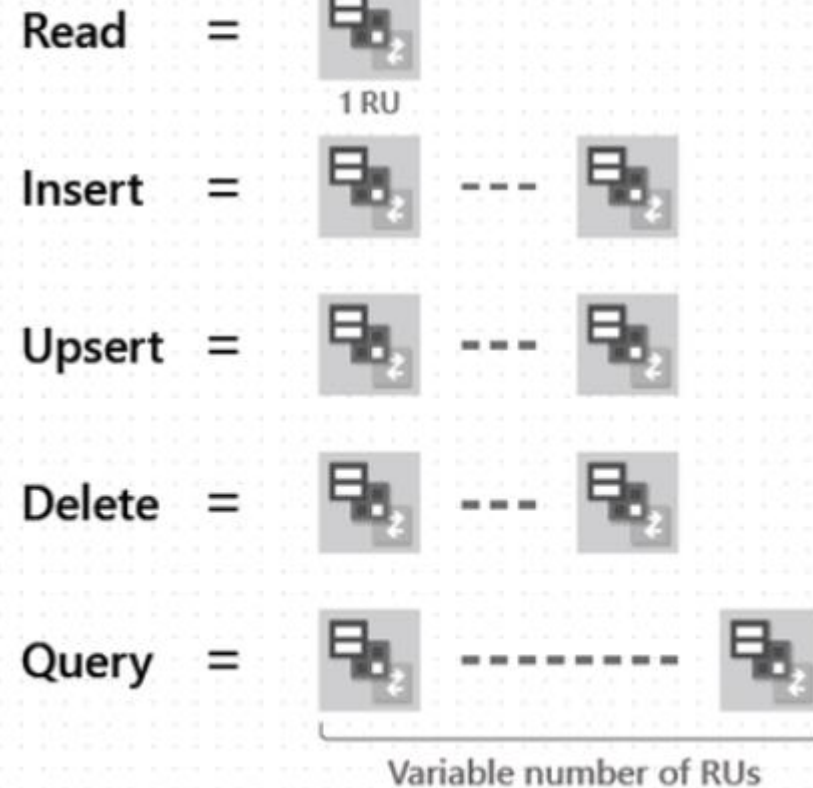


Não é possível provisionar menos de 400 RU/s e elas são provisionadas em incrementos de 100.

Usage is expressed in Request Units



Database operations consume a variable number of RUs



RU

Read =  1 RU

Insert =  6 RU 

Update =  --- 

Delete =  --- 

Query =  

Variable number of RUs

Estimando o consumo de RU/s

Algumas RU/s são normalizadas entre vários métodos de acesso, tornando muitas operações comuns previsíveis. Usando esse conhecimento, você pode fazer algumas estimativas básicas para cargas de trabalho simples. Por exemplo, você pode estimar as RU/s necessárias para operações comuns de banco de dados, como uma RU/s para uma operação de leitura e seis RU/s para uma operação de gravação de um documento de 1 KB em condições ideais

Dica

A medição de RU/s para consultas deve ser feita em escala. Medir consultas em execução em uma só partição física não produzirá dados significativos sobre a taxa de transferência real usada em seu cenário do mundo real depois que ele for implantado e expandido.

Tipo de operação	Número de solicitações por segundo	Número de RUs por solicitação	RU/s necessárias
Gravar um documento	10.000	10	100.000
Consulta principal nº 1	700	100	70.000
Consulta principal nº 2	200	100	20.000
Consulta principal nº 3	100	100	10.000
Total de RU/s			200.000 RU/s

Excesso de requisições

Caso o número de requisições seja maior do que o provisionado, seguindo o padrão de Throughput, será lançado erros por throttling, com o Http status code 429 (Too many requests).

Sugestão de solução: Implementar Cloud Patterns, como Circuit breaker e Retry

TTL

Time to live

Ao projetar uma solução, planeje por quanto tempo seus dados precisarão ser mantidos no Azure Cosmos DB antes de serem migrados por todo o espaço de solução do Azure para minimizar os custos de armazenamento.

DefaultTimeToLive	Validade
<i>Não existe</i>	Os itens não expiram automaticamente
-1	Os itens não expirarão por padrão
<i>n</i>	<i>n</i> segundos após a última modificação

Carga provisionada/Provisionated Throughput

Manual

Auto Scale

As contas sem servidor só podem ser executadas em uma região individual do Azure.

Versão Serverless

A computação sem servidor é ótima para aplicativos com tráfego imprevisível ou intermitente. Você pode usá-la com um aplicativo, como:

- Um novo aplicativo com cargas de usuários de difícil previsão
- Um novo aplicativo de protótipo na sua organização
- Integração da computação sem servidor a um serviço como o Azure Functions
- Apenas começando a usar o Azure Cosmos DB como um novo desenvolvedor
- Aplicativo de tráfego baixo que não envia nem recebe dados em alto volume

As contas sem servidor só podem ser executadas em uma região individual do Azure.

Versão Serverless X Capacidade Provisionada

Capacidade Provisionada é ideal para cenários em que a RU/s provisionada completa é consumida por > 66% de horas por mês.

Serverless será útil se a sua equipe não puder prever as necessidades de taxa de transferência com precisão ou usar o valor máximo de taxa de transferência de < 66% de horas por mês.

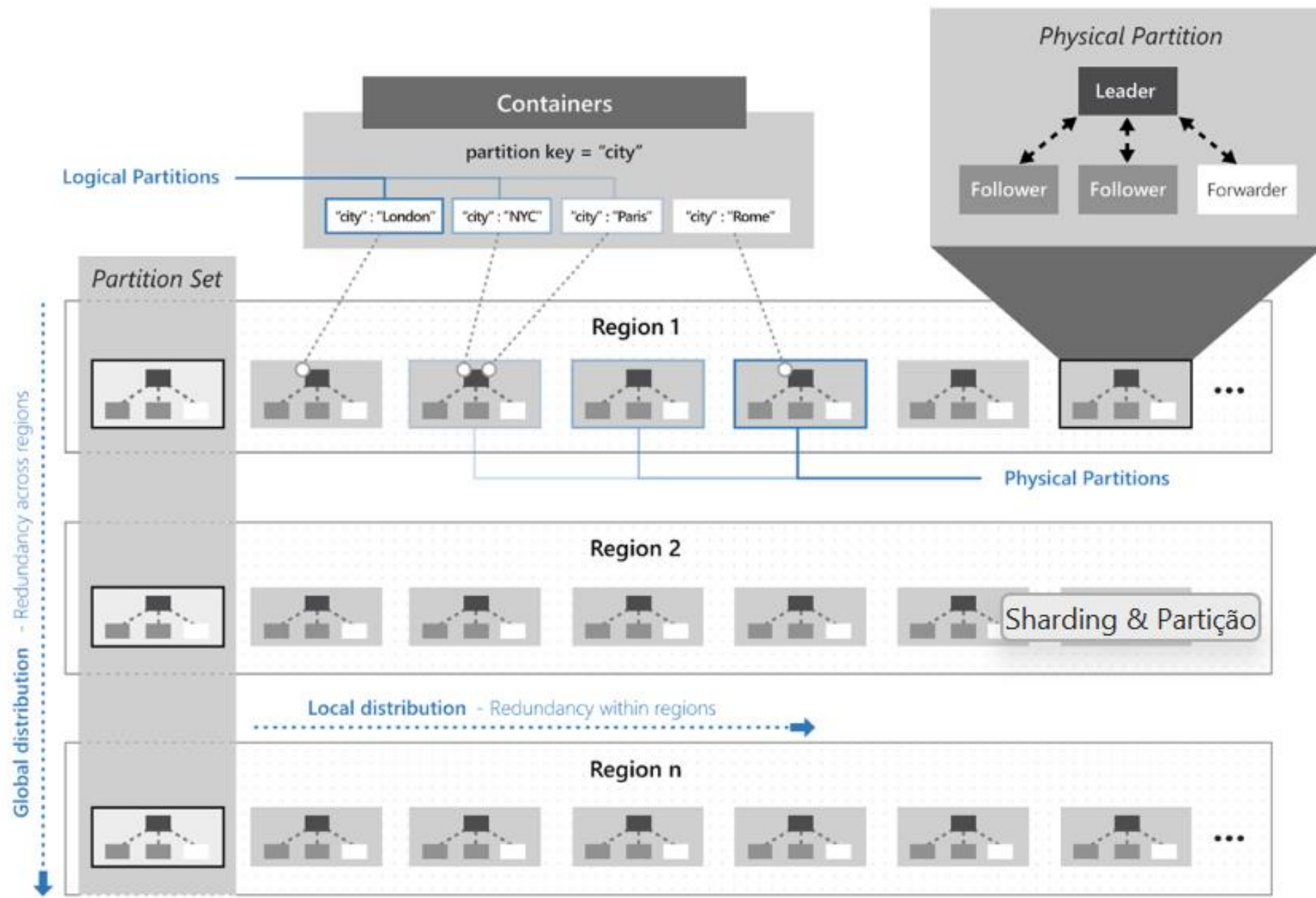
Começando com CosmosDB

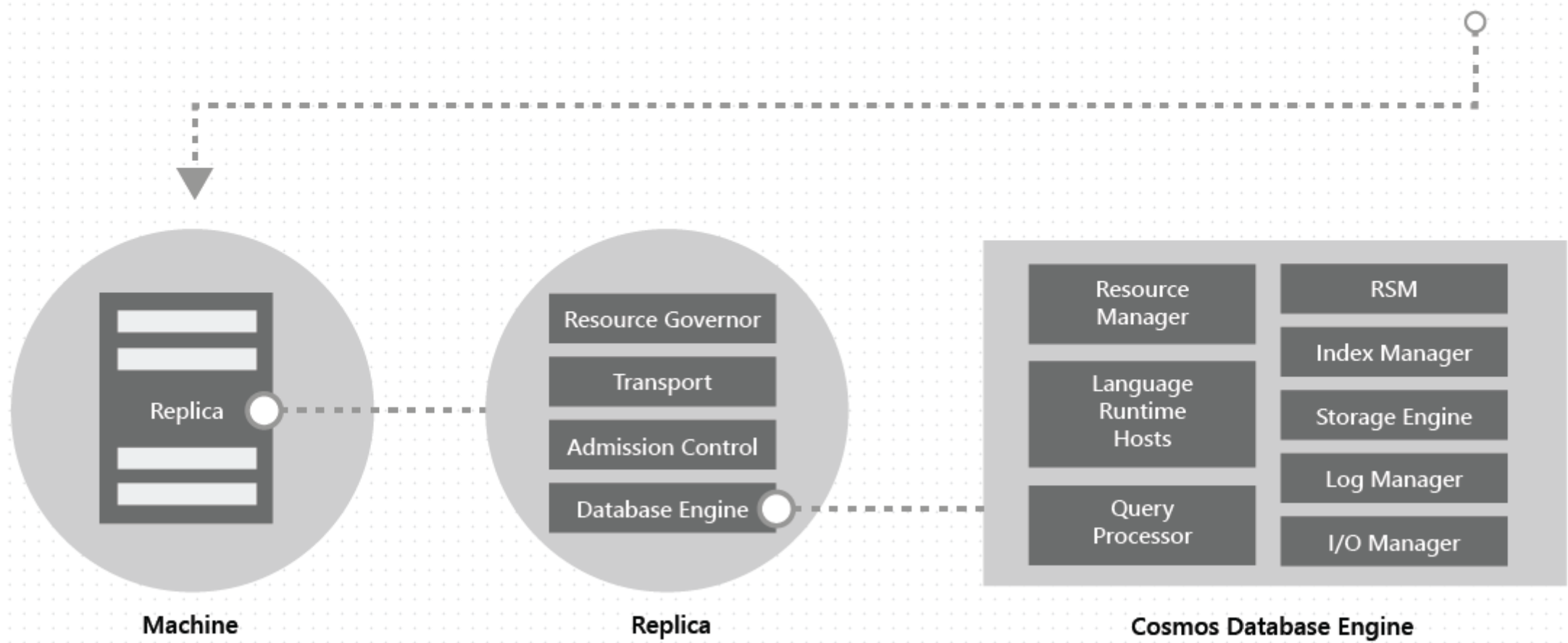
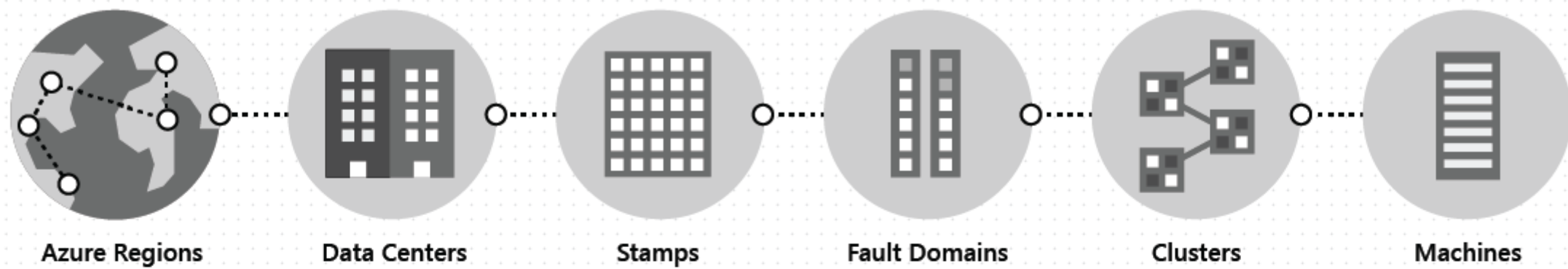
- Container
- Camada gratuita no Azure
- Pay as you go

Dica: Invista seu tempo aprendendo sobre como fazer um (melhor mais performático) particionamento.

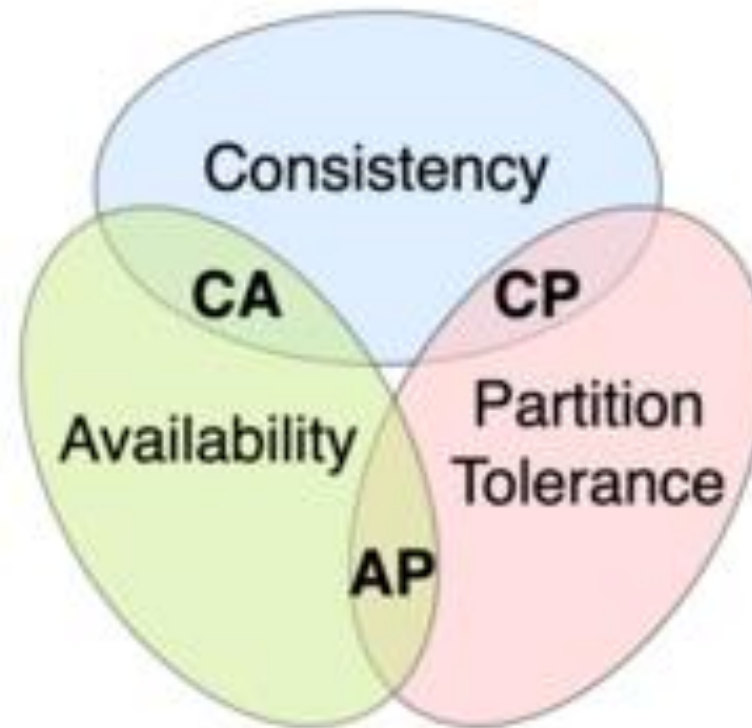
Sharding & Partição

- Lógica (20GB)
 - Se escolher o campo userID como partição e tiver 1000 registros, terão 1000 partições logicas
- Física
 - A quantidade de taxa de transferência provisionada (cada partição física individual pode fornecer uma taxa de transferência de até dez mil unidades de solicitação por segundo
 - O armazenamento de dados total (cada partição física individual pode armazenar até 50 GB de dados).
 - Não há limite para o número total de partições físicas no contêiner.



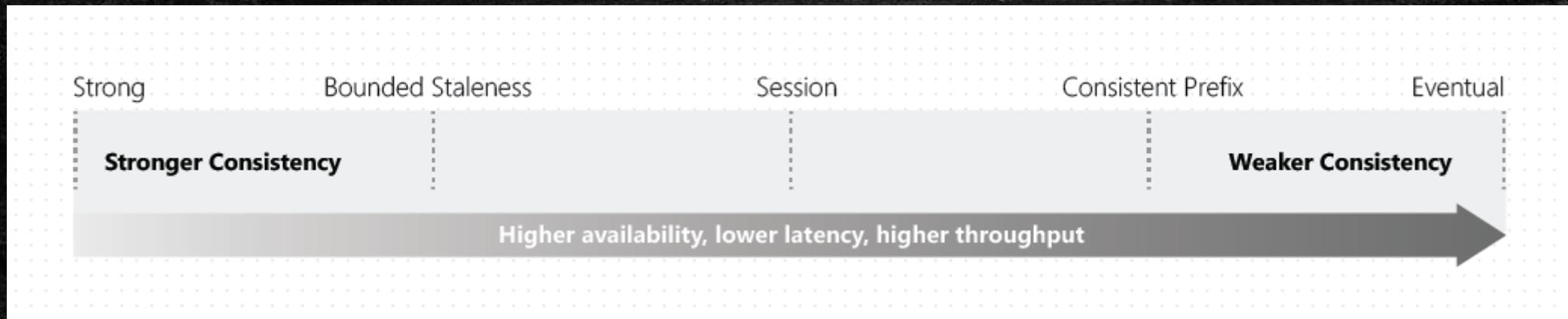


Teorema CAP



Tipos de consistência

- Forte (Strong): Isso significa que as leituras têm a garantia de ver a gravação mais recente.
- Desatualização limitada (Bounded Staleness): Quanto tempo abrimos mão para ter a info em todas as instancias
- Sessão (Session): Cada insert tem um id de sessão e esse precisa ser compartilhado
- Prefixo consistente (Consistent Prefix)
- Eventual (Eventual): Fora de ordem



Tipos de consistência

Nível de coerência	Descrição
Forte	Consistência linear. Os dados são replicados e confirmados em todas as regiões configuradas antes de serem reconhecidos como confirmados e visíveis para todos os clientes.
Desatualização Limitada	Lê gravações atrasadas por um limite configurado no tempo ou nos itens.
Sessão	Em uma sessão específica (instância do SDK), os usuários podem ler as próprias gravações.
Prefixo Coerente	As leituras podem ficar atrás das gravações, mas as leituras nunca aparecerão fora de ordem.
Eventual	As leituras acabarão sendo consistentes com as gravações.

Talk is cheap
Show me the code

<https://devblogs.microsoft.com/cosmosdb/introducing-bulk-support-in-the-net-sdk/>

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/query/pagination>

<https://learn.microsoft.com/pt-br/azure/cosmos-db/partitioning-overview>

<https://learn.microsoft.com/pt-br/azure/cosmos-db/nosql/database-transactions-optimistic-concurrency>

<https://learn.microsoft.com/en-us/azure/cosmos-db/global-dist-under-the-hood>

<https://learn.microsoft.com/en-us/credentials/certifications/azure-cosmos-db-developer-specialty/?practice-assessment-type=certification>

<https://learn.microsoft.com/en-us/azure/cosmos-db/provision-throughput-autoscale>