Configure Cosmos DB for NoSQL database & containers

Serverless

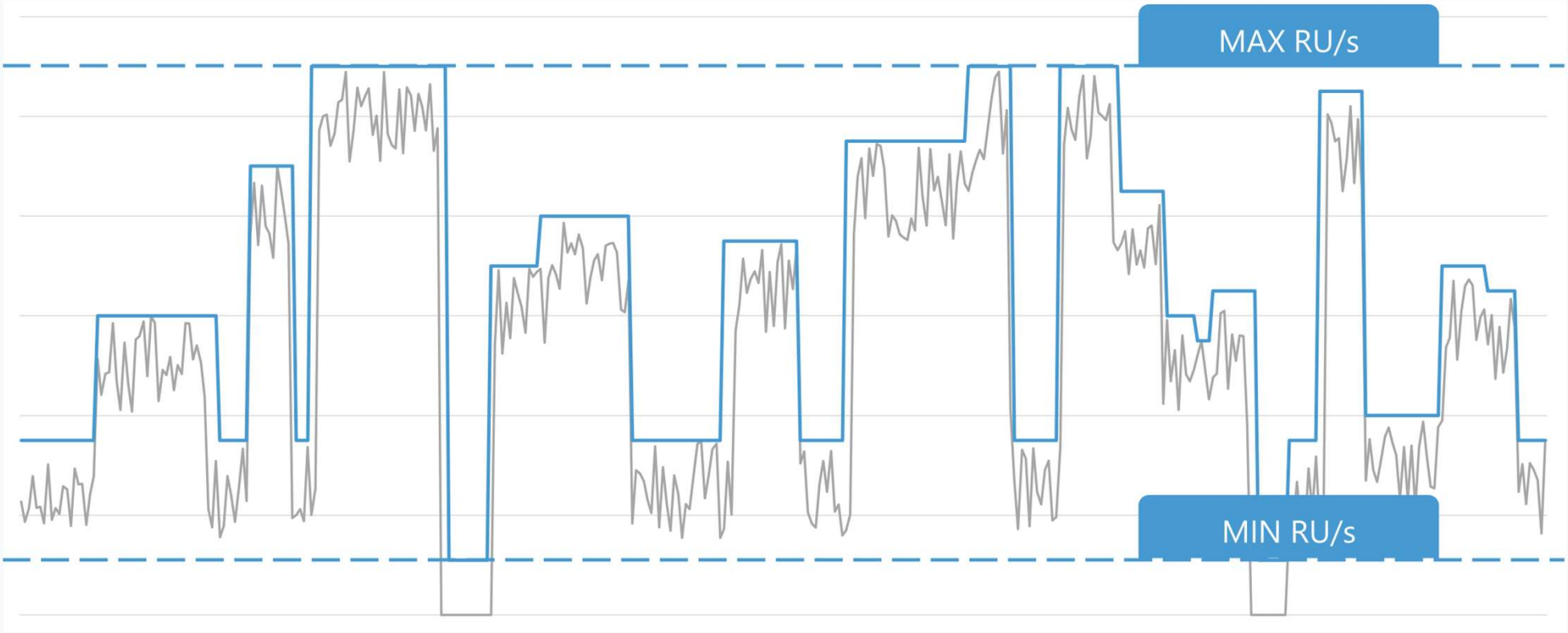
- Consumption-based model where each request consumes request units
- Serverless is great for applications with unpredictable or bursty traffic
- You can use serverless with an application such as:
  - A new application with hard to forecast users loads
  - A new prototype application within your organization
  - Serverless compute integration with a service like Azure Functions
  - Just getting started with Azure Cosmos DB as a new developer
  - Low traffic application that doesn't send or receive numerous data

Compare serverless vs. provisioned throughput

	Provisioned Throughput	Serverless
Workloads	Ideal for workloads with predictable traffic patterns	Can handle workloads that have wildly varying traffic
Request Units	Makes some number of request units available each second to each container for database operations	Doesn't require any planning or automatic provisioning
Global Distribution	Supports distributing your data to an unlimited number of Azure regions.	Can only run in a single Azure region.
Storage Limits	Allows you to store unlimited data in a container.	Only allows up to 50 GB of data in a container.

Autoscale throughput

- Autoscale is great for workloads with variable or unpredictable traffic patterns and can minimize unused capacity that would typically be pre-provisioned.



Compare autoscale vs. standard (manual) throughput

	Standard Throughput	Auto scale Throughput
Workloads	Ideal for workloads with predictable traffic patterns	Can handle workloads that have varying traffic
Request Units	Requires a static number of request units to be assigned ahead of time.	With autoscale, you only set the maximum, and the minimum billed will be 10% of the maximum when there are zero requests.
Scenarios	When your team can accurately predict the amount of throughput your application needs,	If your team cannot predict your throughput needs accurately
Rate-limiting	Will always remain static at the set RU/s that is provisioned. Requests beyond this will be rate-limited	Will scale up to the max RU/s before similarly rate-limiting responses

Migrate between standard and autoscale throughput

- Existing containers can be migrated to and from autoscale
- During the migration process, the system will automatically apply a request unit per second (RU/s) value to the container.

Exercise

- Configure throughput for Azure Cosmos DB SQL API with the Azure portal

Thank You