Amazon DynamoDB

A serverless NoSQL database for scale and performance

Samaneh Utter
Amazon DynamoDB Specialist Solutions Architect
AWS

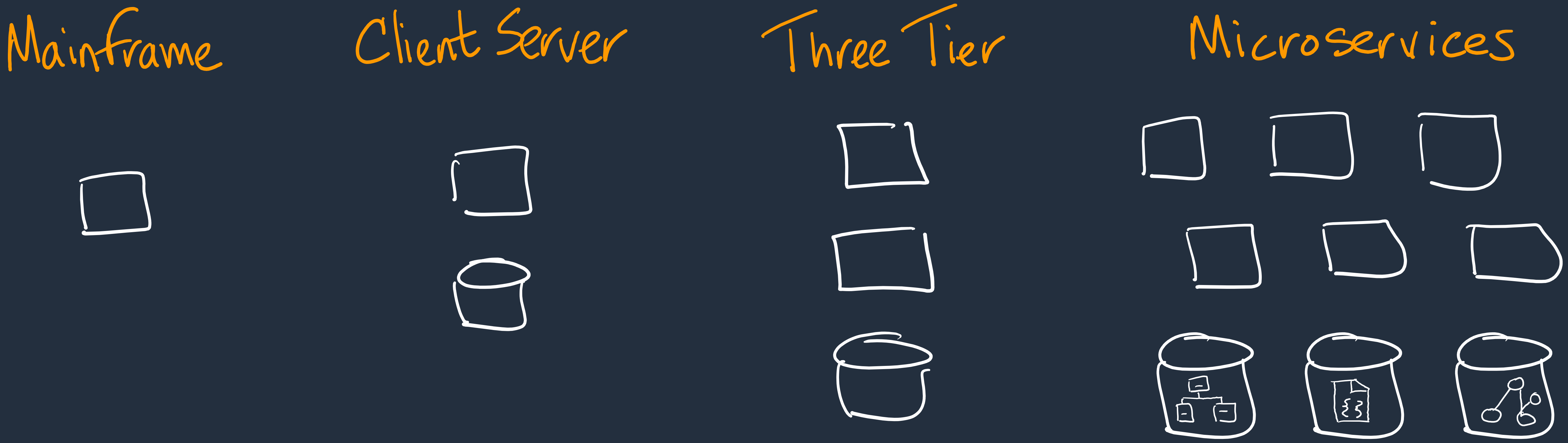
DynamoDB

Performance at scale

No servers to manage

Enterprise ready

Application architecture and patterns have evolved



Characteristics of internet-scale apps

E-commerce

Media streaming

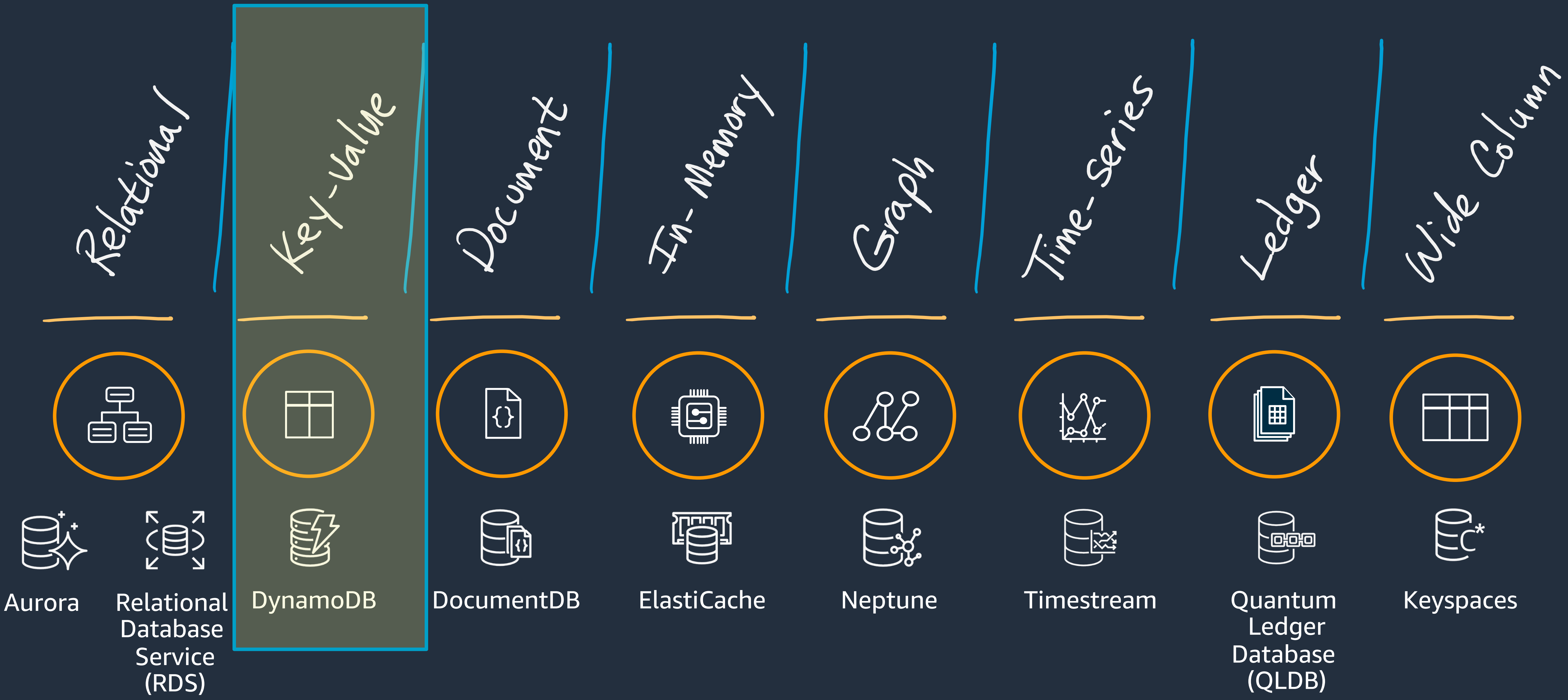
Social media

Online gaming

Shared economy

Users	1 million+
Data volume	TB, PB, EB
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Mobile, IoT, devices
Scale	Up and down
Economics	Pay as you go
Developer access	Instant API access

AWS purpose-built databases



Relational	Key-Value	Document	In-Memory	Graph	Time-series	Ledger	Wide Column
Aurora Relational Database Service (RDS)	DynamoDB	DocumentDB	ElastiCache	Neptune	Timestream	Quantum Ledger Database (QLDB)	Keyspaces

DynamoDB

Performance at scale

- Handles millions of requests per second
- Delivers single-digit-millisecond latency
- Automated global replication
- New advanced streaming with Amazon Kinesis Data Streams for DynamoDB

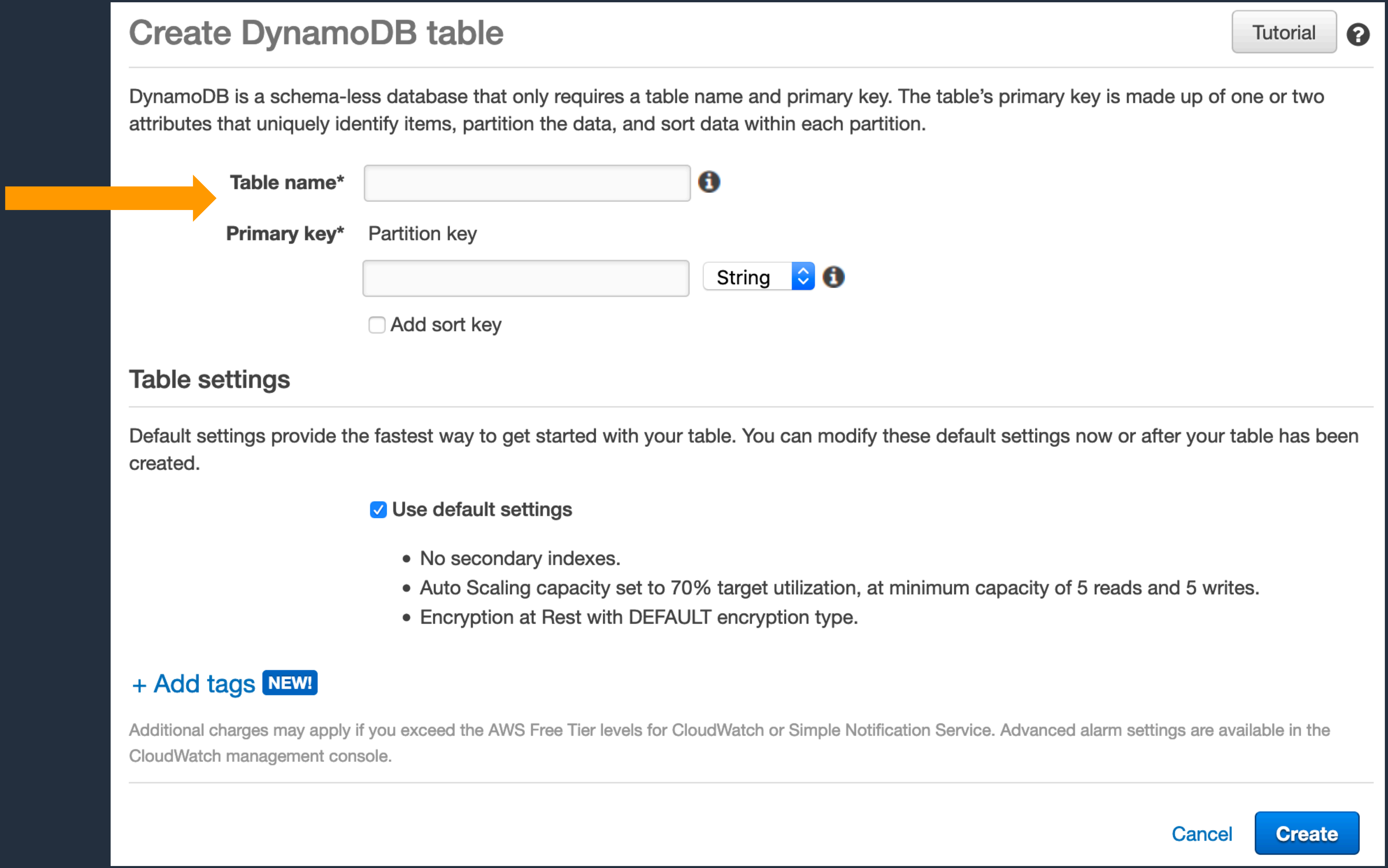
No servers to manage

- Maintenance free
- Auto scaling
- On-demand capacity mode
- Change data capture for integration with AWS Lambda, Amazon Redshift, Amazon Elasticsearch Service
- AWS Glue Elastic Views for DynamoDB

Enterprise ready

- ACID transactions
- Encryption at rest
- Continuous backups (PITR), and on-demand backup and restore
- NoSQL Workbench
- Export table data to S3
- PartiQL (a SQL-compatible query language) support

Core components



Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name*

Primary key* Partition key

String

□ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

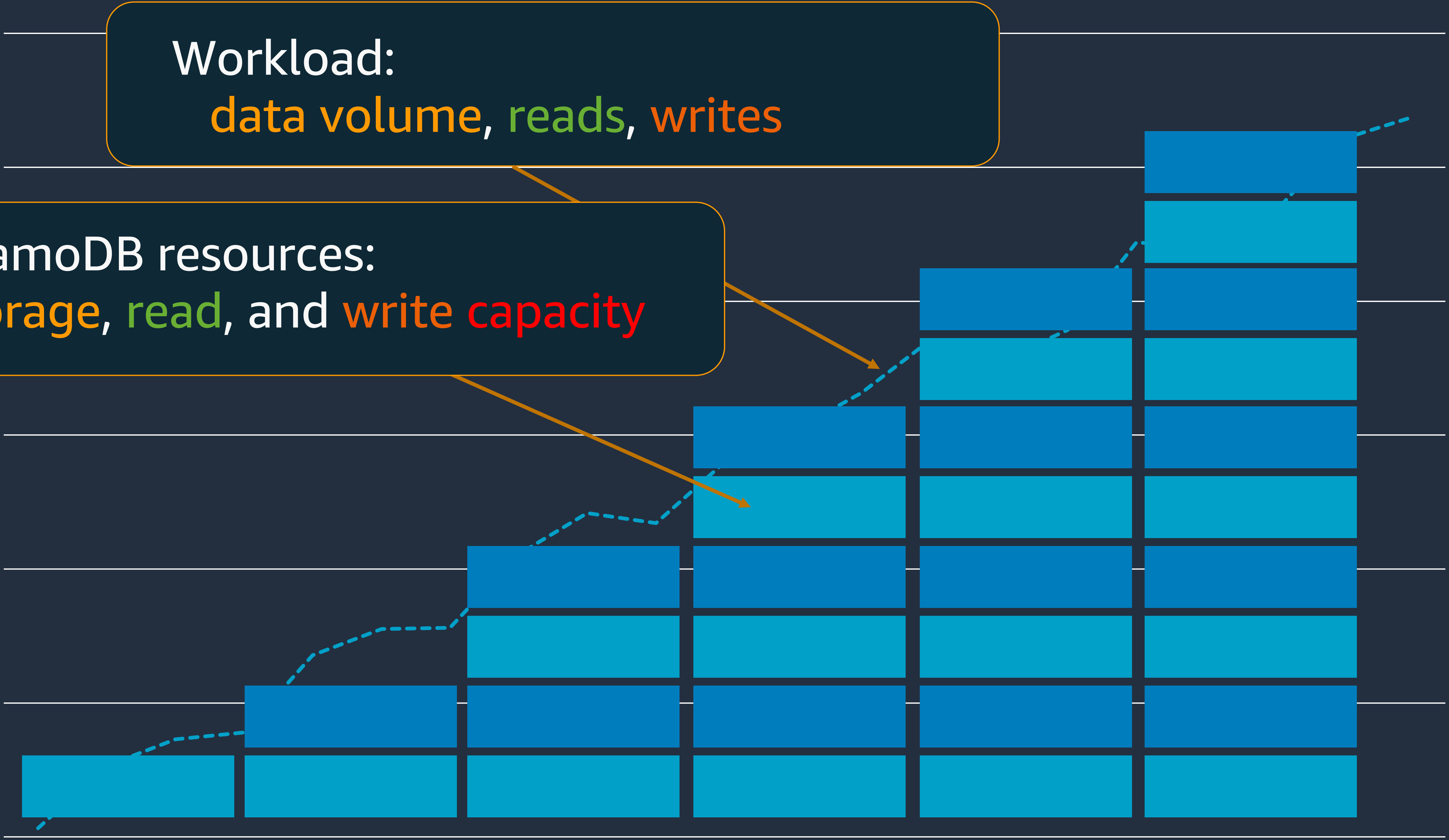
☑ **Use default settings**

- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization, at minimum capacity of 5 reads and 5 writes.
- Encryption at Rest with DEFAULT encryption type.

+ Add tags NEW!

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

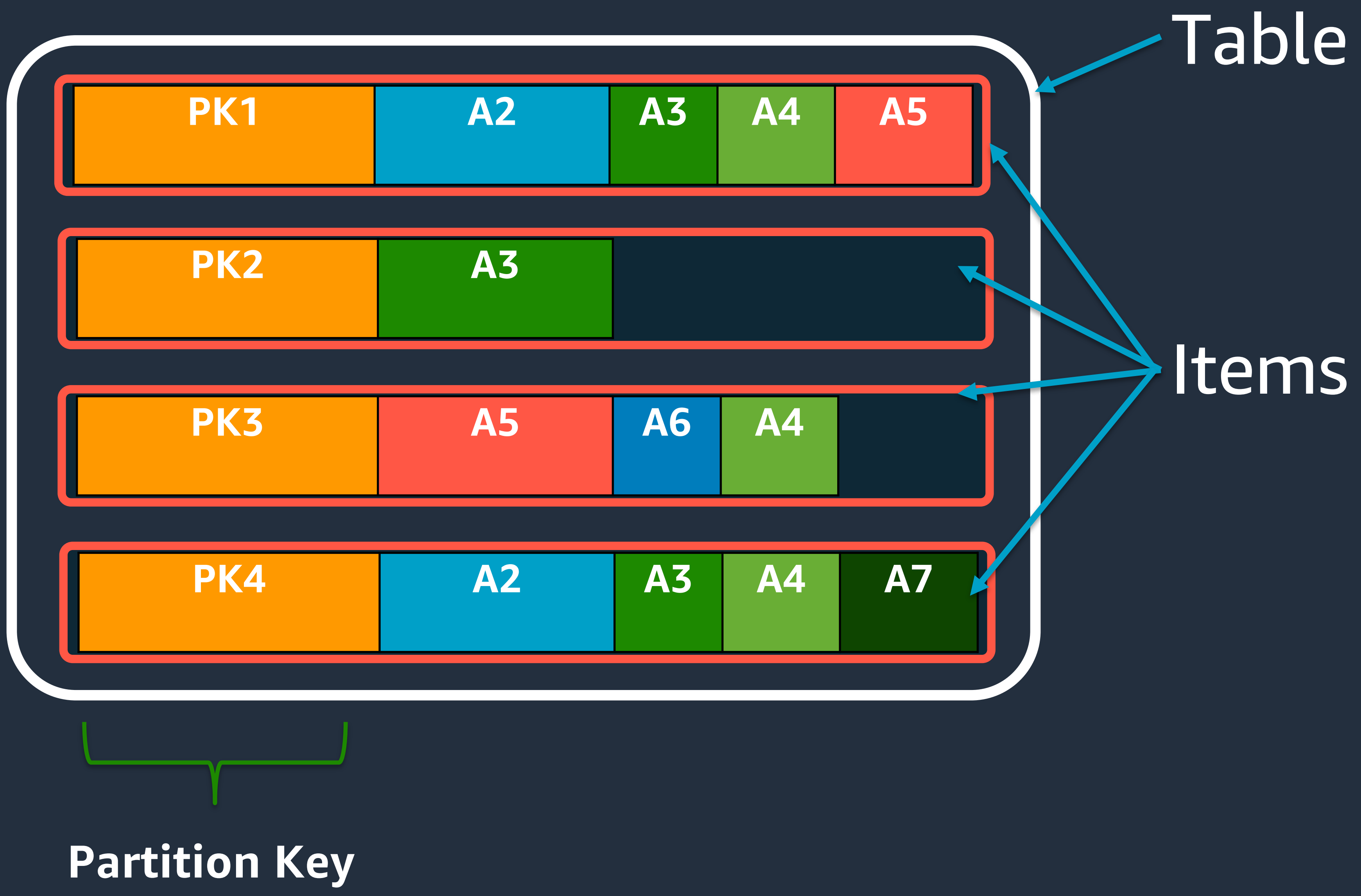
Horizontal scaling with DynamoDB



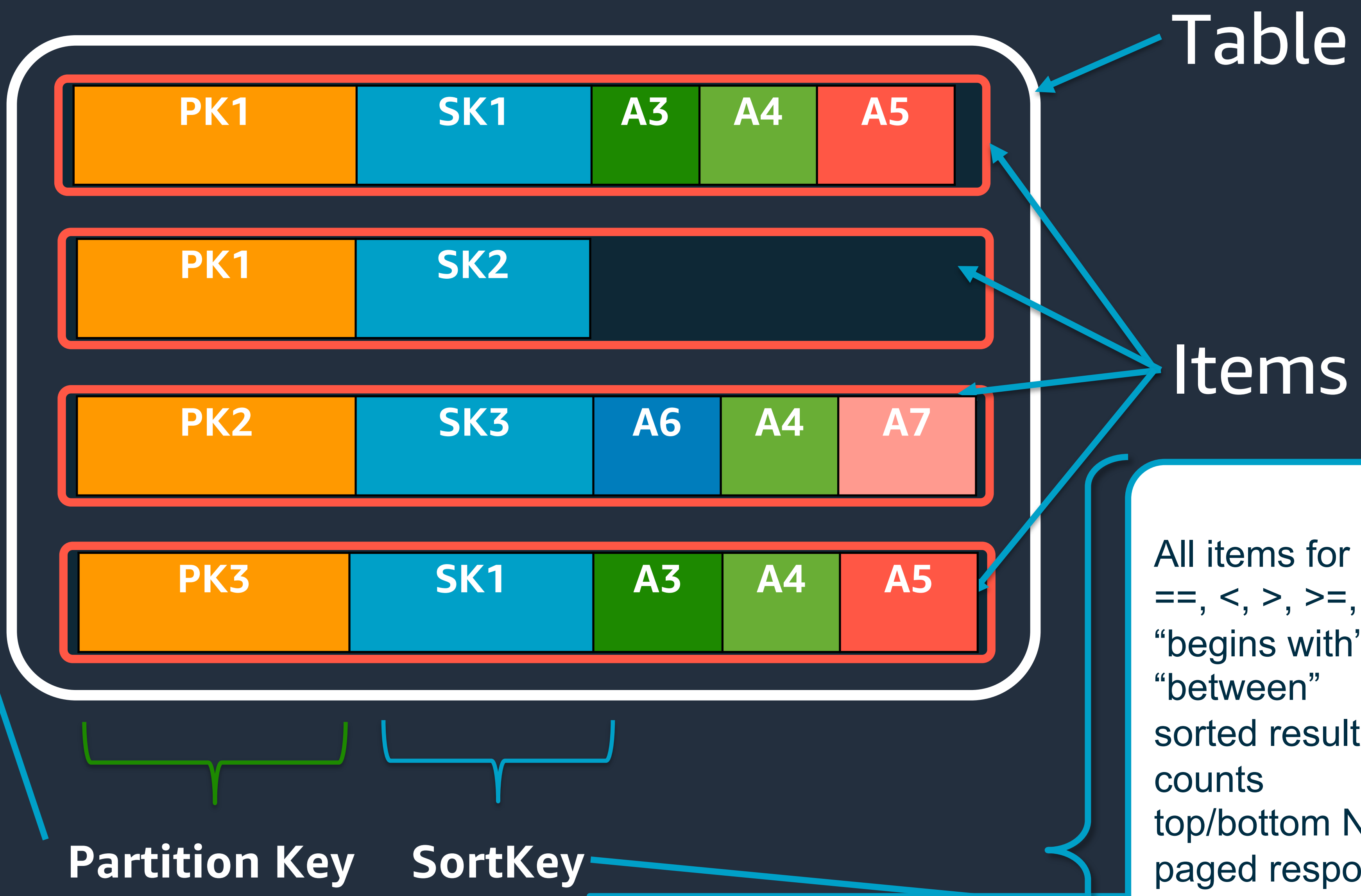
Workload:
data volume, reads, writes

DynamoDB resources:
storage, read, and write capacity

DynamoDB Table



DynamoDB Table



Mandatory
Key-value access pattern
Determines data distribution

Optional
Model 1:N relationships
Enables rich query capabilities

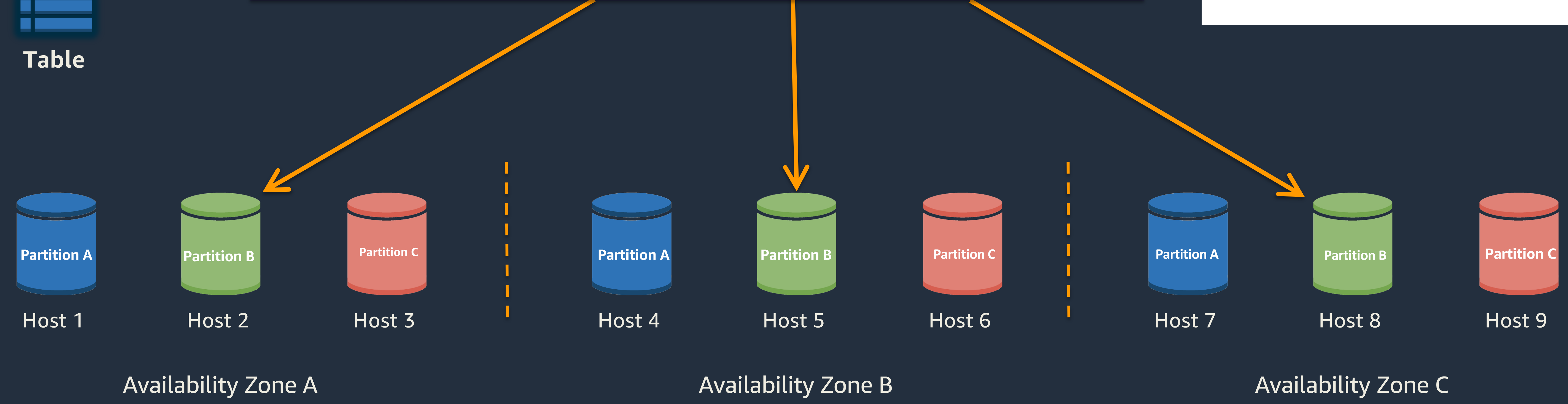
All items for a partition key
==, <, >, >=, <=
“begins with”
“between”
sorted results
counts
top/bottom N values
paged responses

A view from “a different angle”

	CustID	Customer Information
	523422	{ name:“Alex”, city:”London”, …}

Hash(523422) = 0xF355

Hash Value	CustID	Customer Information
0xF355	523422	{ name:“Alex”, city:”London”, …}



Three-way replication

- Data is always replicated to three Availability Zones
- The service runs in three Availability Zones

Local secondary index (LSI)

Added at table creation time.

RCUs/WCUs are consumed from base Table.

Up to 5 LSIs per table.

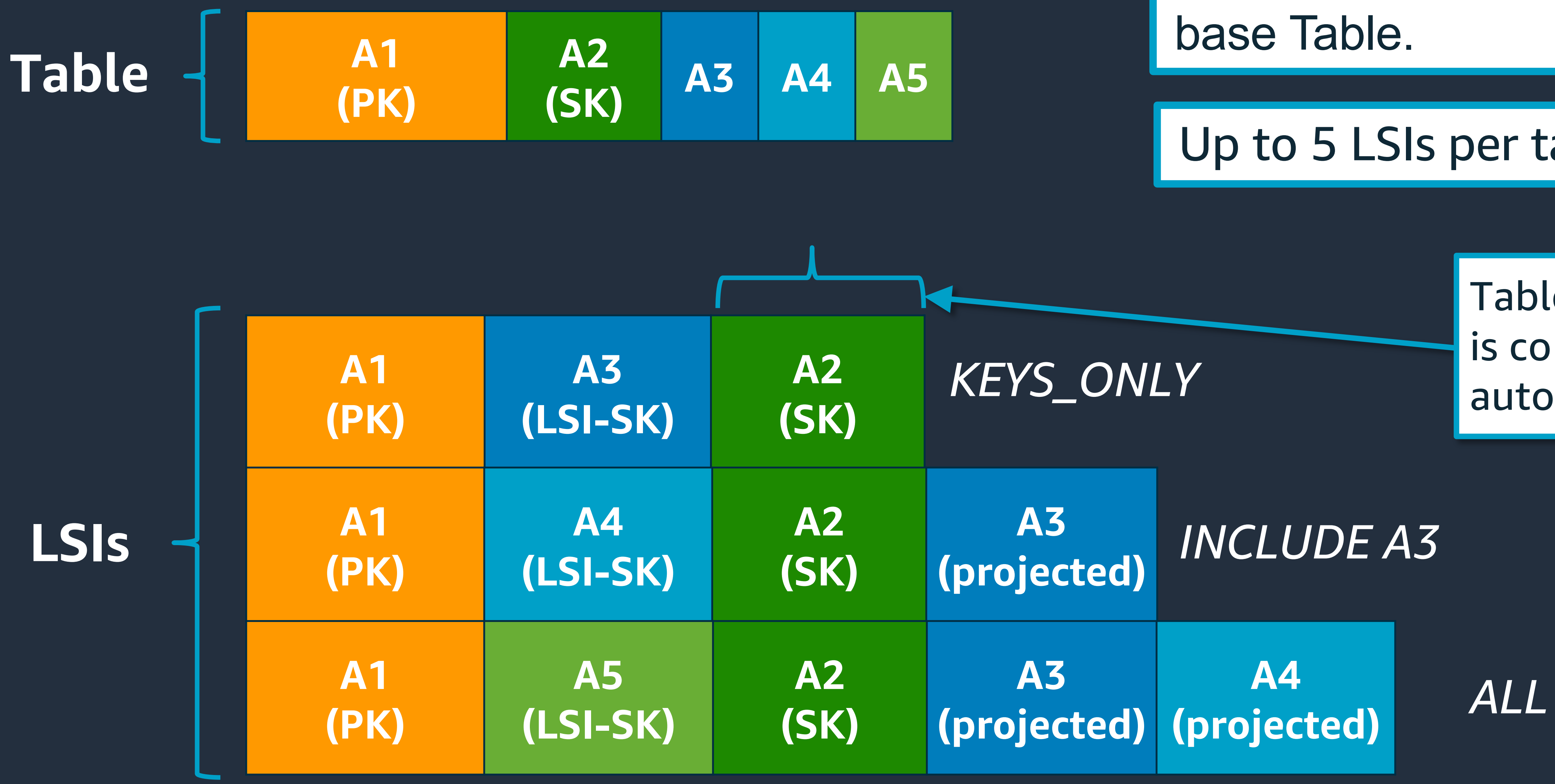


Table: A1 (PK) | A2 (SK) | A3 | A4 | A5

LSIs:

A1 (PK) | A3 (LSI-SK) | A2 (SK) — *KEYS_ONLY*

A1 (PK) | A4 (LSI-SK) | A2 (SK) | A3 (projected) — *INCLUDE A3*

A1 (PK) | A5 (LSI-SK) | A2 (SK) | A3 (projected) | A4 (projected) — *ALL*

Table’s sort key is copied automatically.

Global secondary index (GSI)

Up to 20 GSIs per table.

RCUs/WCUs provisioned separately for GSIs.

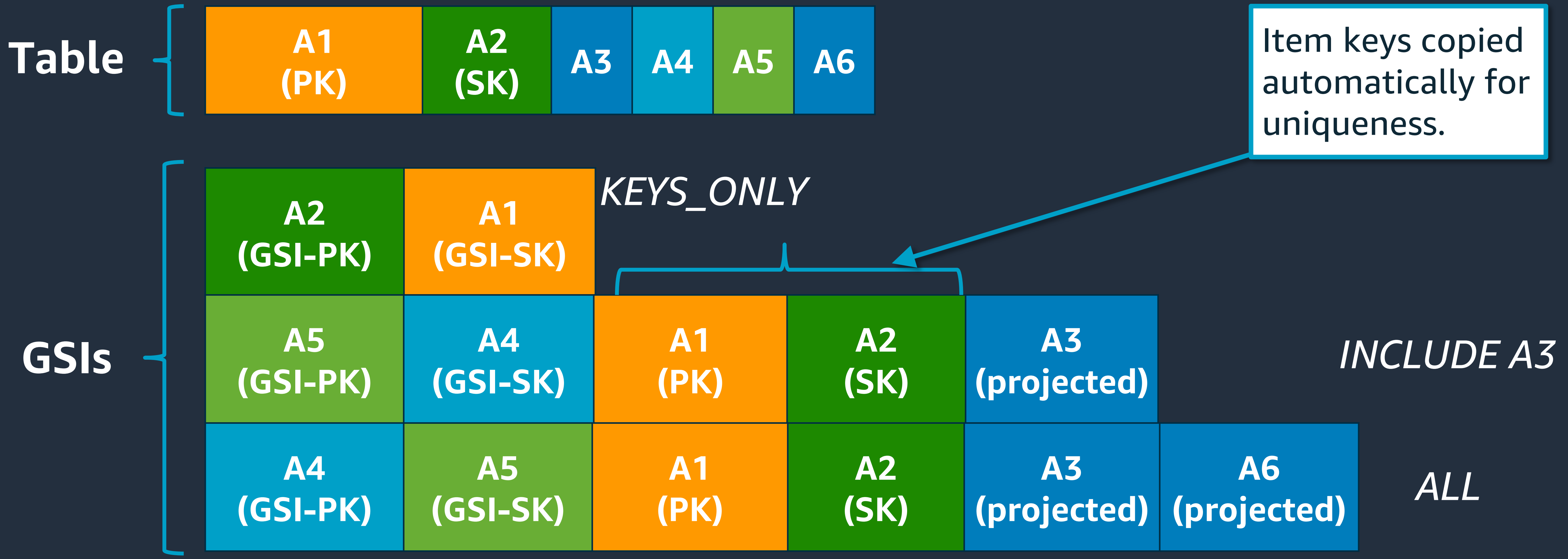
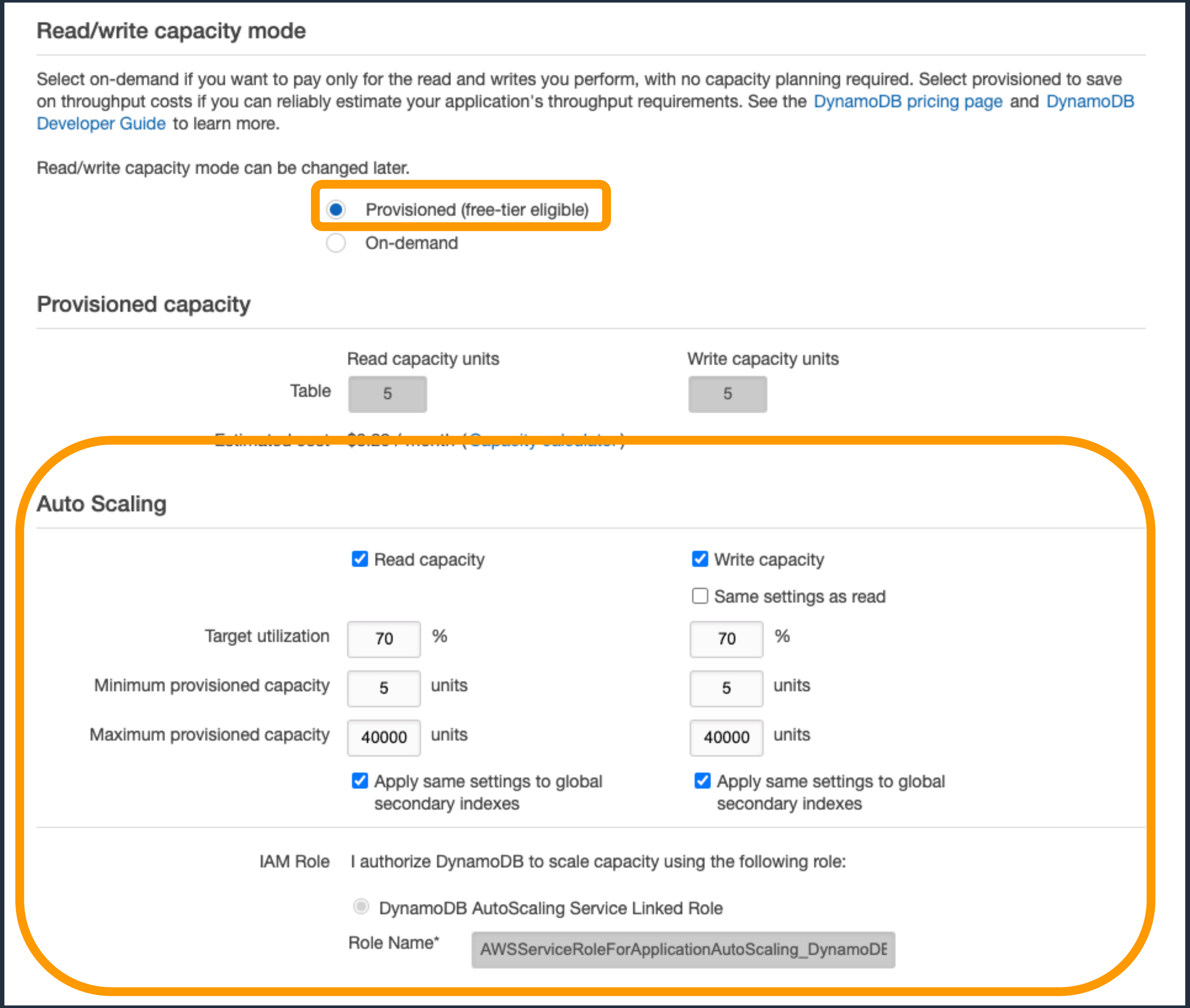


Table	A1 (PK)	A2 (SK)	A3	A4	A5	A6	
GSIs	A2 (GSI-PK)	A1 (GSI-SK)					*KEYS_ONLY*
	A5 (GSI-PK)	A4 (GSI-SK)	A1 (PK)	A2 (SK)	A3 (projected)		*INCLUDE A3*
	A4 (GSI-PK)	A5 (GSI-SK)	A1 (PK)	A2 (SK)	A3 (projected)	A6 (projected)	*ALL*

Item keys copied automatically for uniqueness.

Read/write capacity modes

Provisioned capacity mode



Read/write capacity mode

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the DynamoDB pricing page and DynamoDB Developer Guide to learn more.

Read/write capacity mode can be changed later.

● Provisioned (free-tier eligible)
○ On-demand

Provisioned capacity

	Read capacity units	Write capacity units
Table	5	5

Estimated cost $[illegible] / month (Capacity calculator)

Auto Scaling

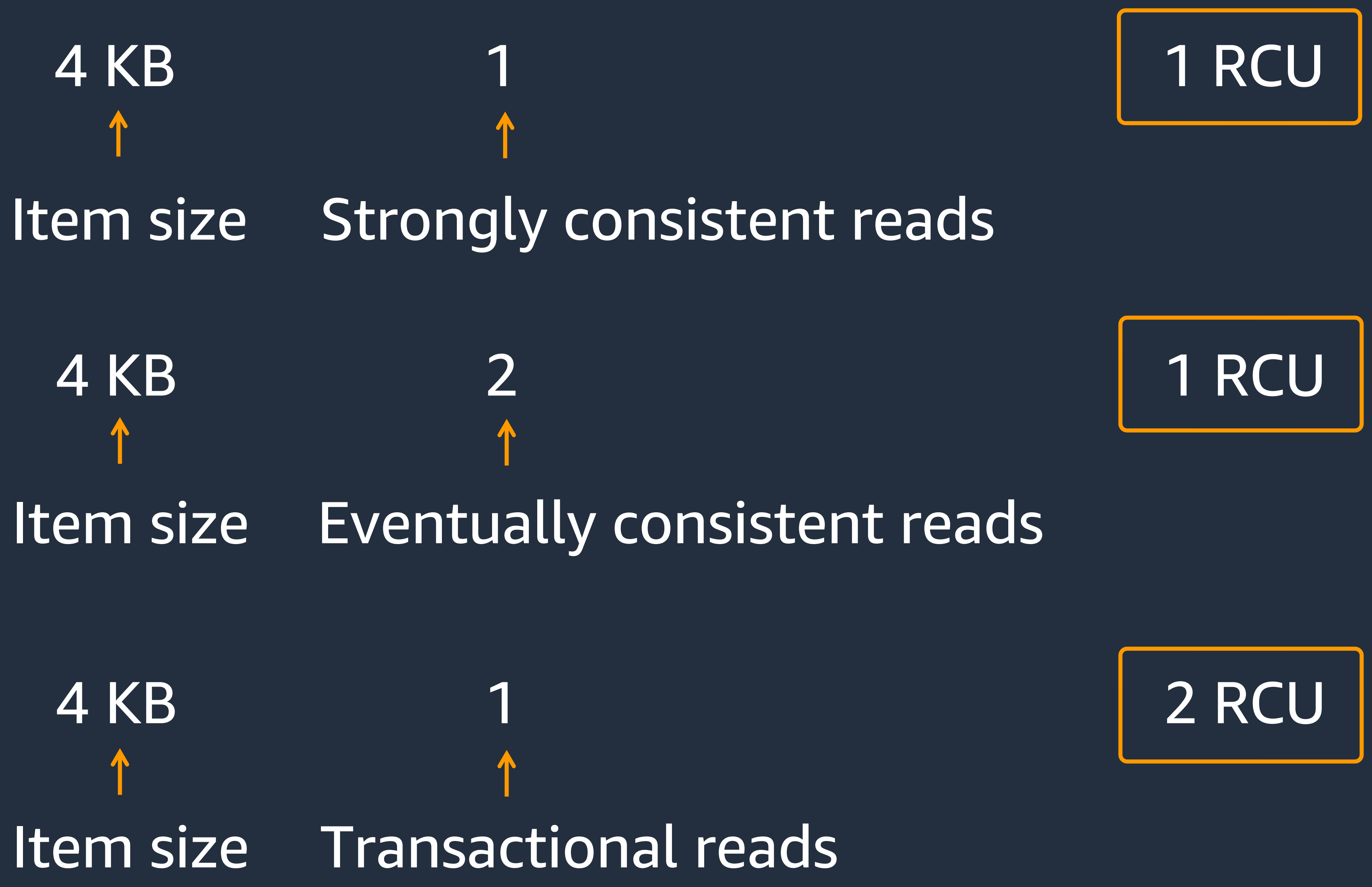
	☑ Read capacity		☑ Write capacity	
			□ Same settings as read	
Target utilization	70	%	70	%
Minimum provisioned capacity	5	units	5	units
Maximum provisioned capacity	40000	units	40000	units
	☑ Apply same settings to global secondary indexes		☑ Apply same settings to global secondary indexes	

IAM Role I authorize DynamoDB to scale capacity using the following role:

○ DynamoDB AutoScaling Service Linked Role

Role Name* AWSServiceRoleForApplicationAutoScaling_DynamoDE

Read capacity unit

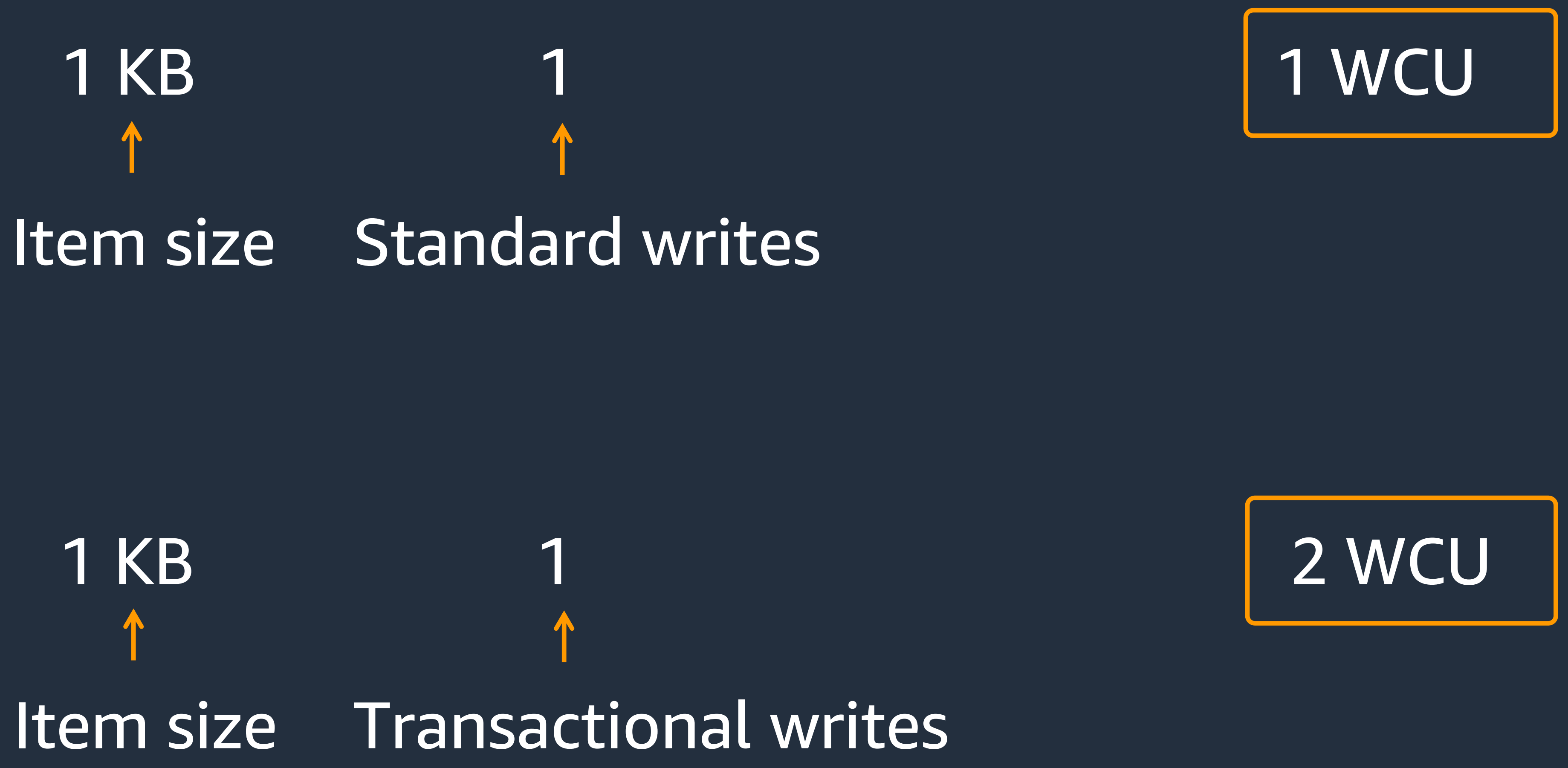


4 KB ↑ Item size — 1 ↑ Strongly consistent reads — 1 RCU

4 KB ↑ Item size — 2 ↑ Eventually consistent reads — 1 RCU

4 KB ↑ Item size — 1 ↑ Transactional reads — 2 RCU

Write capacity unit



1 KB ↑ Item size

1 ↑ Standard writes

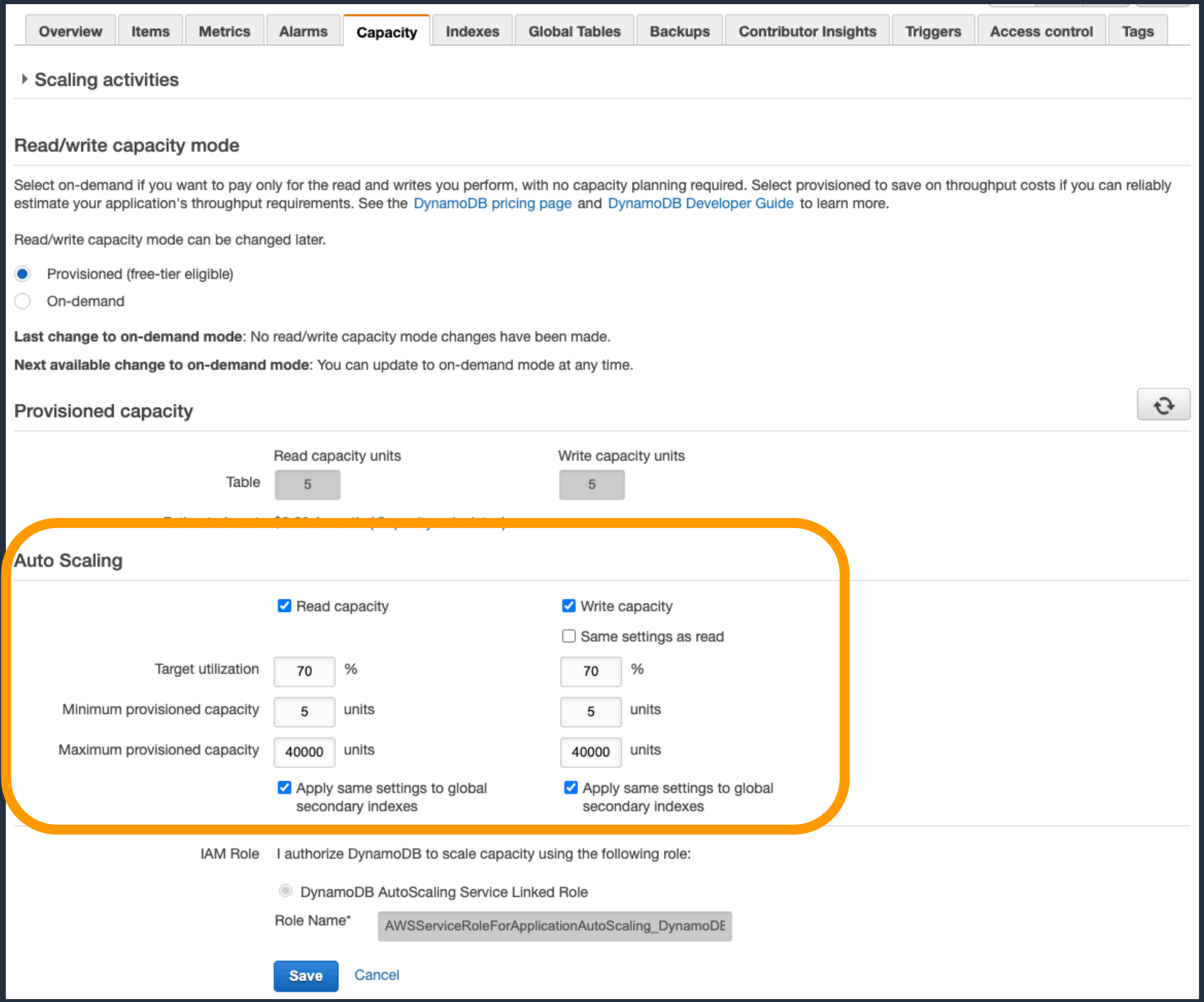
1 WCU

1 KB ↑ Item size

1 ↑ Transactional writes

2 WCU

Provisioned capacity mode: auto scaling



▸ Scaling activities

Read/write capacity mode

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the DynamoDB pricing page and DynamoDB Developer Guide to learn more.

Read/write capacity mode can be changed later.

● Provisioned (free-tier eligible)

○ On-demand

Last change to on-demand mode: No read/write capacity mode changes have been made.

Next available change to on-demand mode: You can update to on-demand mode at any time.

Provisioned capacity

	Read capacity units	Write capacity units
Table	5	5

Auto Scaling

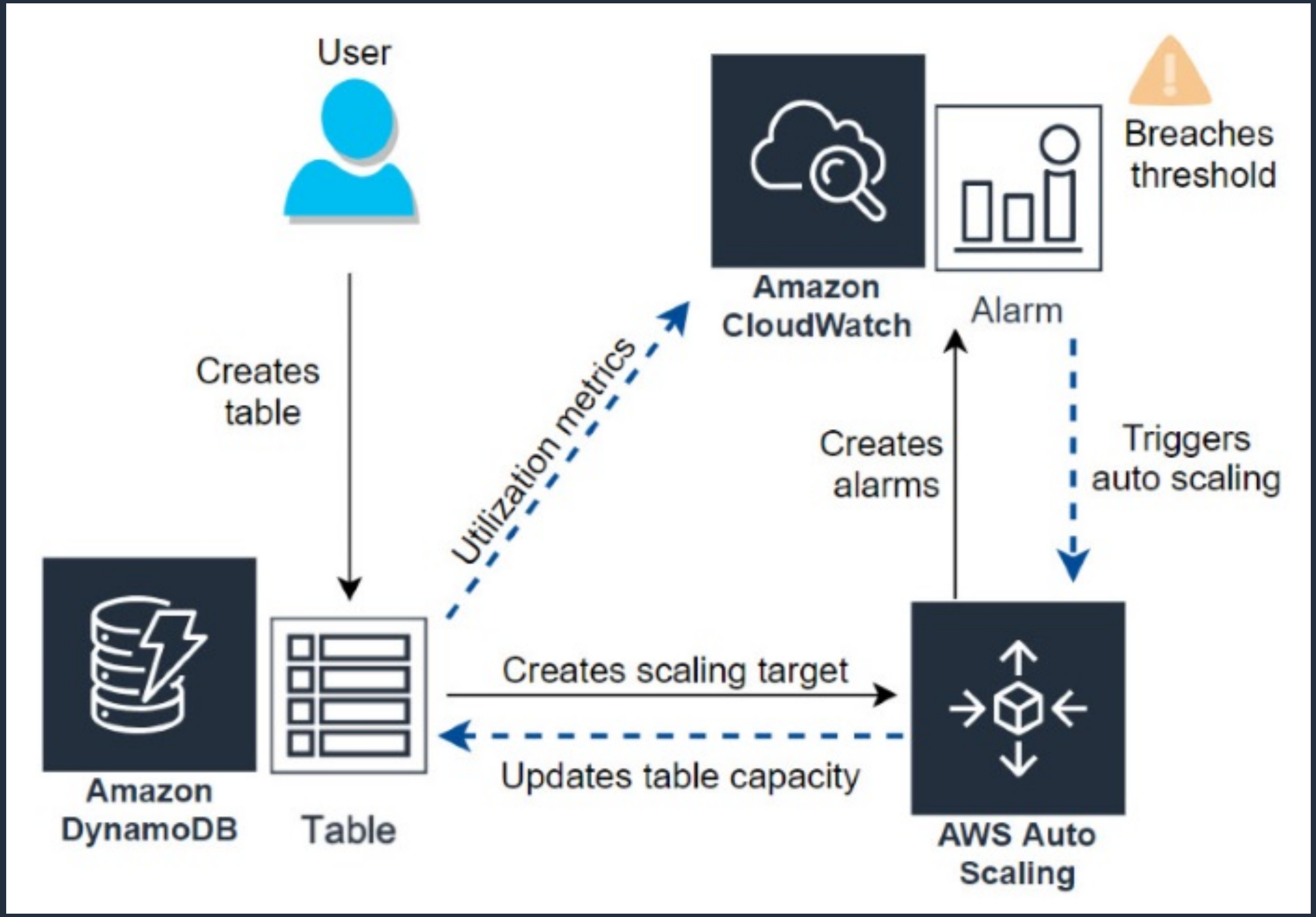
	☑ Read capacity	☑ Write capacity ☐ Same settings as read
Target utilization	70 %	70 %
Minimum provisioned capacity	5 units	5 units
Maximum provisioned capacity	40000 units	40000 units
	☑ Apply same settings to global secondary indexes	☑ Apply same settings to global secondary indexes

IAM Role I authorize DynamoDB to scale capacity using the following role:

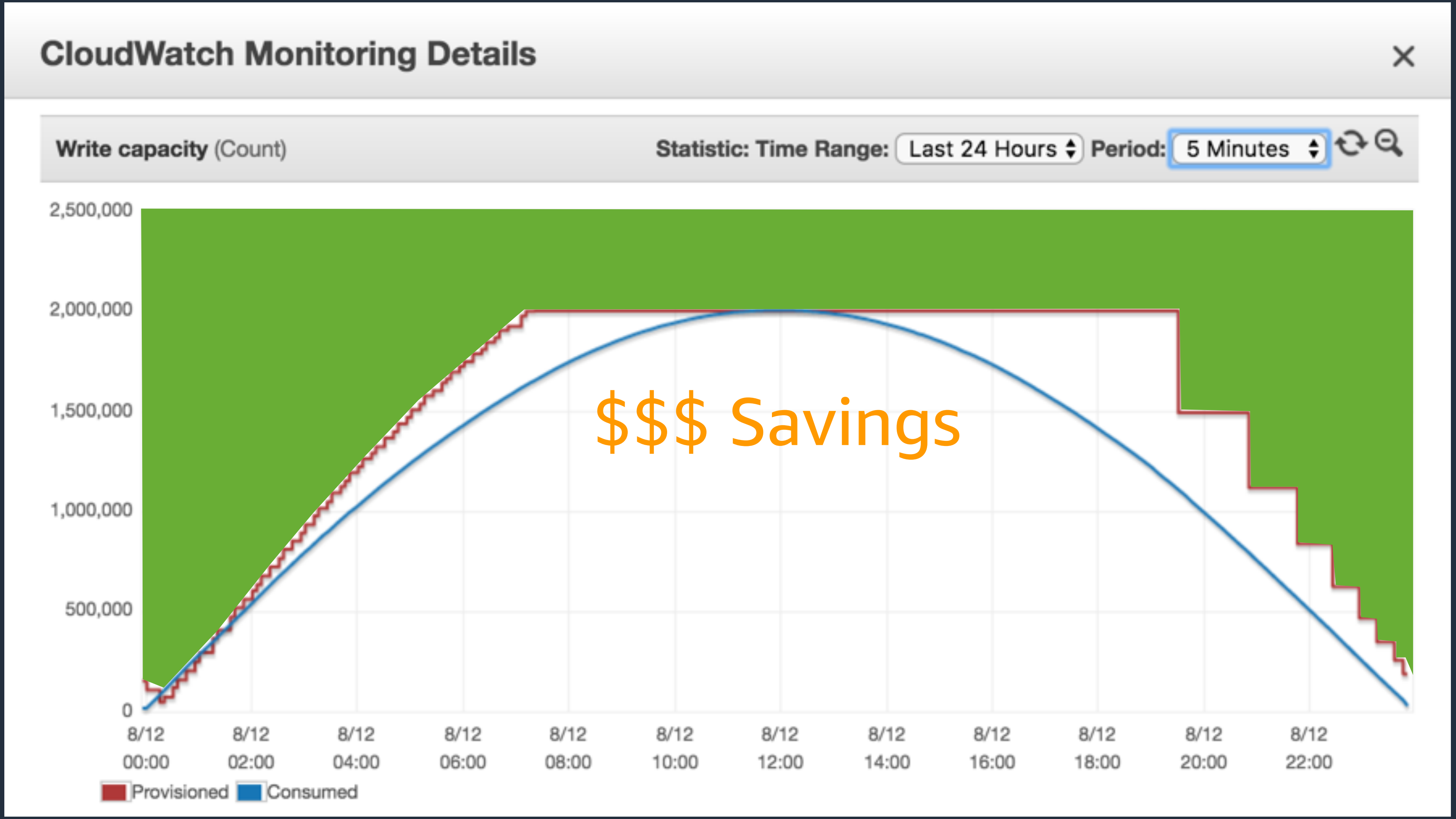
● DynamoDB AutoScaling Service Linked Role

Role Name* AWSServiceRoleForApplicationAutoScaling_DynamoDE

Provisioned capacity mode: auto scaling



Provisioned capacity mode: auto scaling, maintains performance



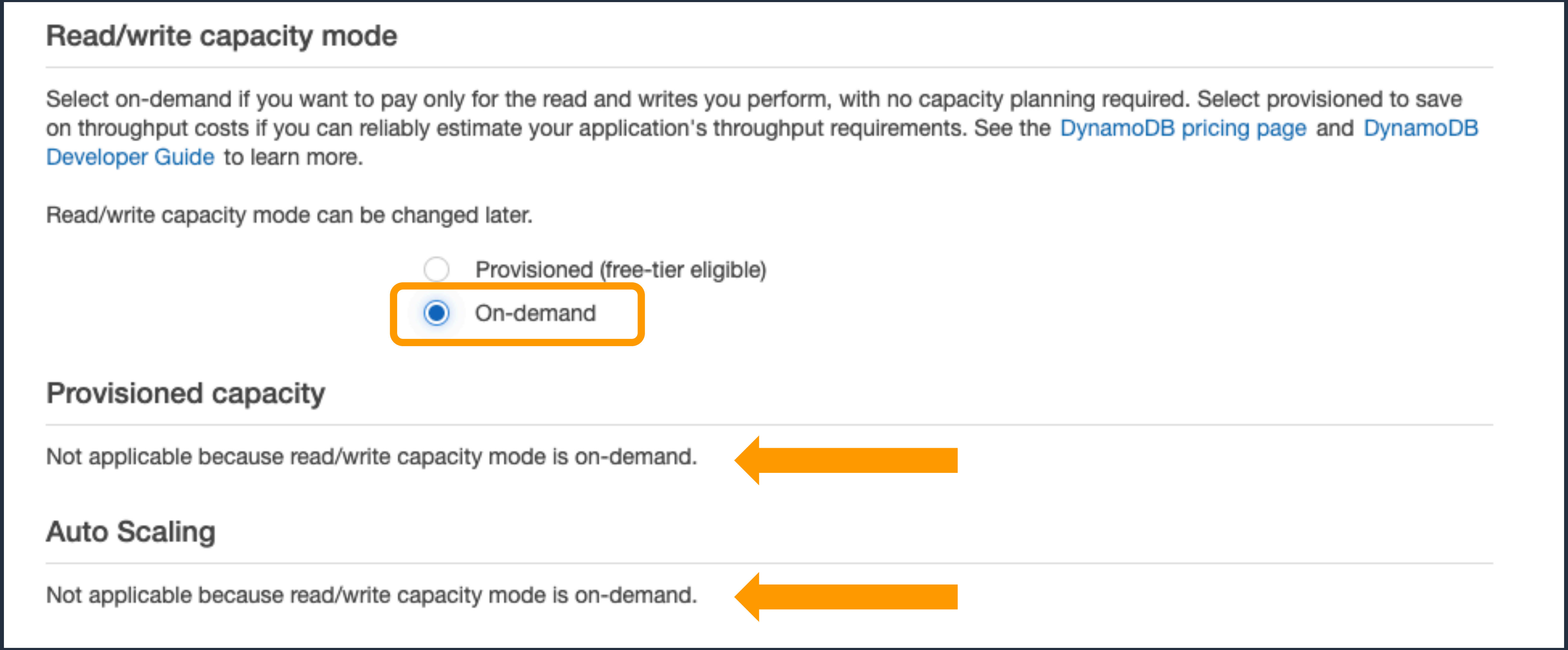
Automated scaling policies

Scales up when you need it

Scales down when you don't

Scheduled auto scaling

On-demand capacity mode



Read/write capacity mode

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the DynamoDB pricing page and DynamoDB Developer Guide to learn more.

Read/write capacity mode can be changed later.

○ Provisioned (free-tier eligible)

● On-demand

Provisioned capacity

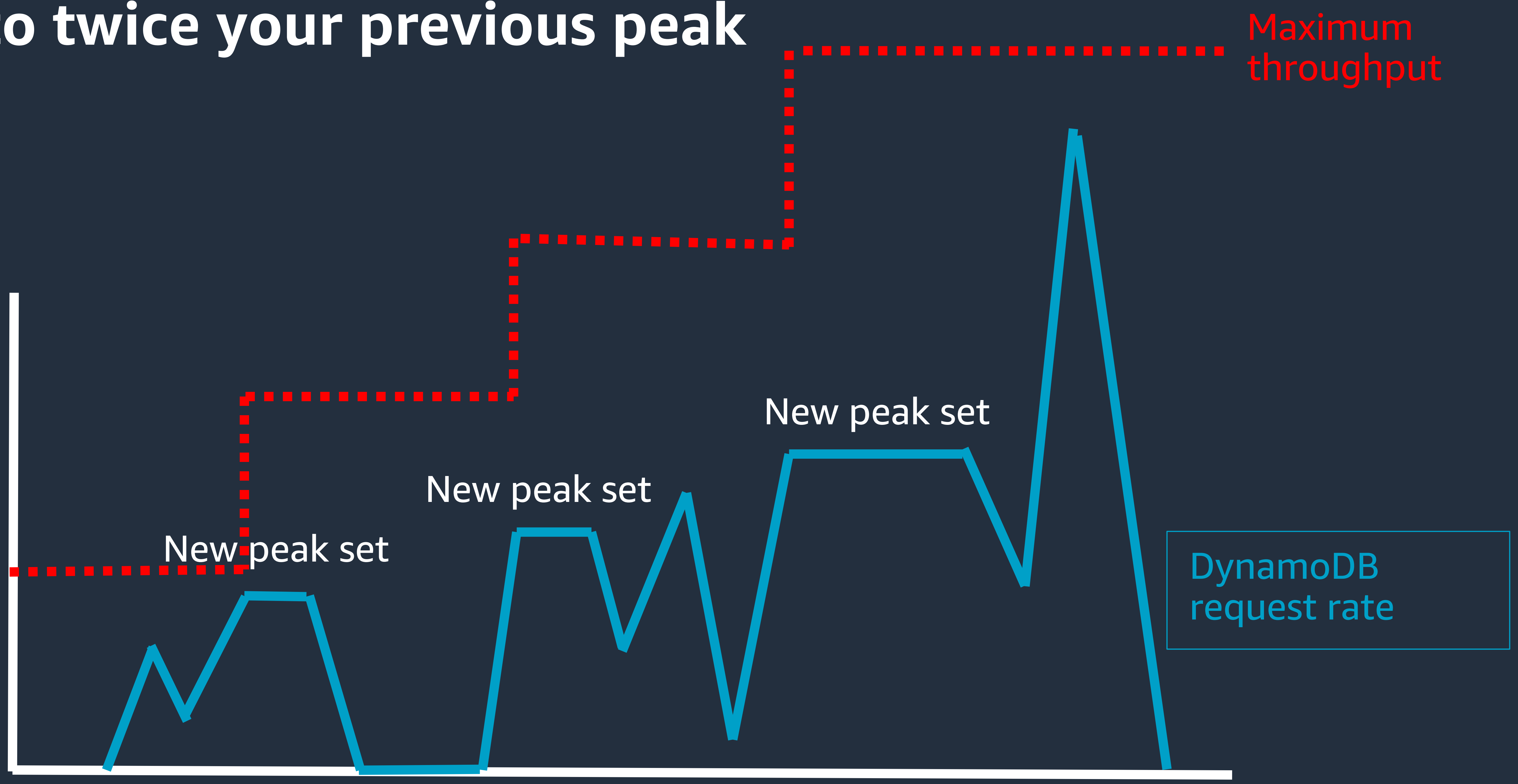
Not applicable because read/write capacity mode is on-demand.

Auto Scaling

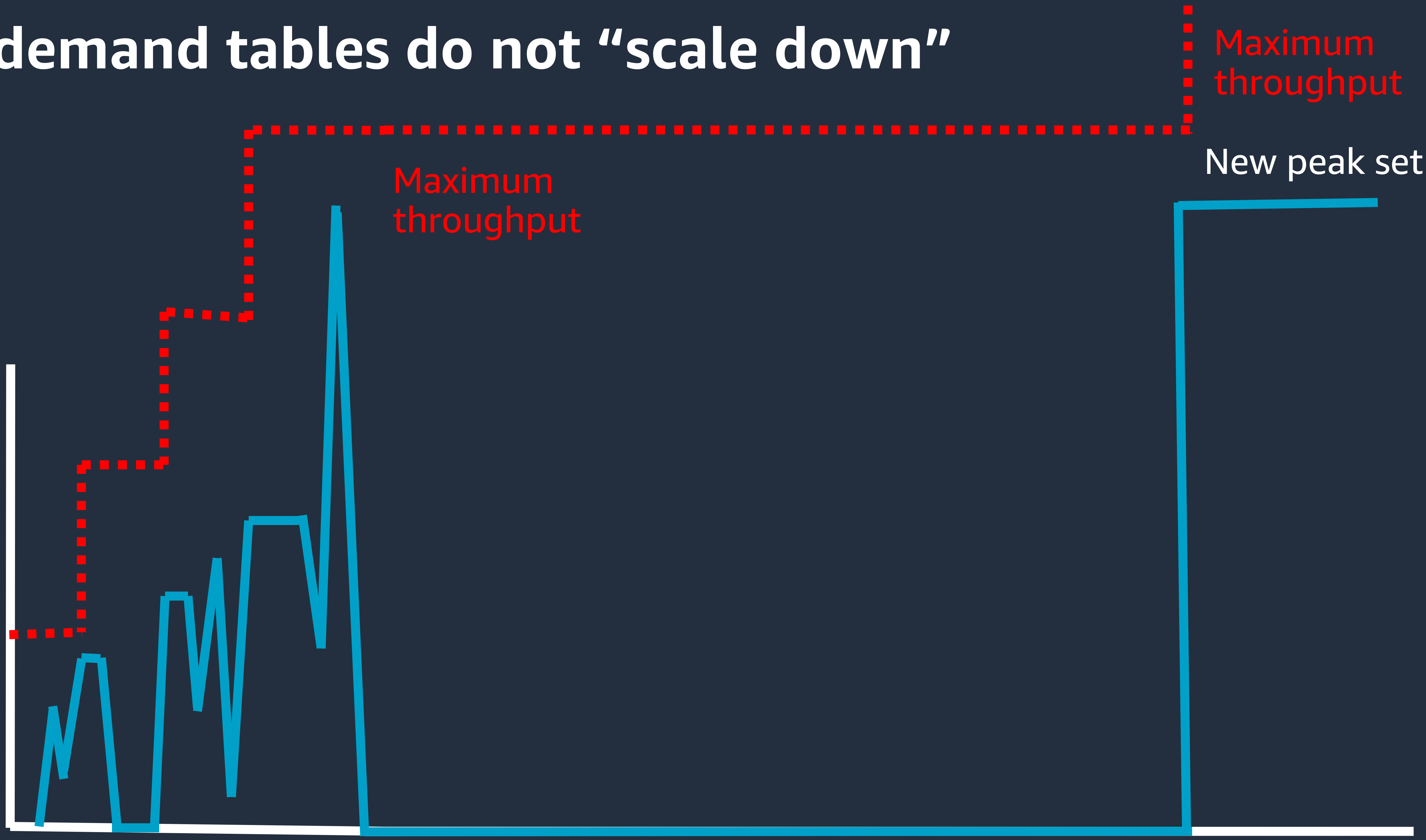
Not applicable because read/write capacity mode is on-demand.

Pay per request: use nothing, pay nothing

Maximum Scaling Capabilities: Up to twice your previous peak



On-demand tables do not "scale down"



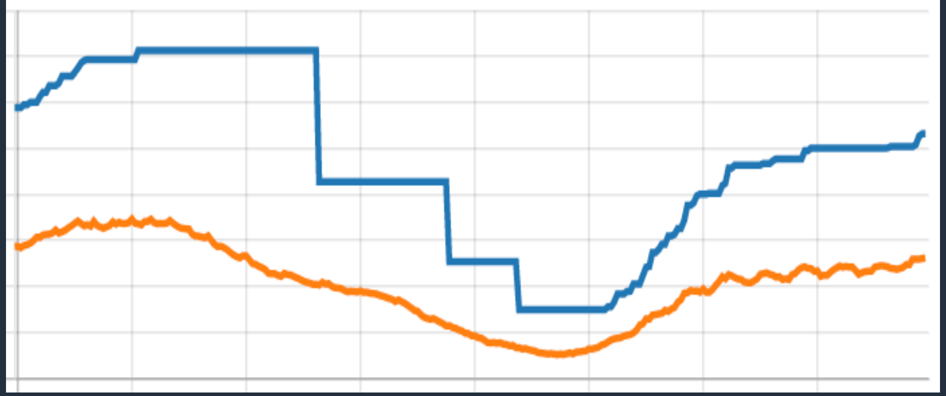
Use provisioned mode

Steady workloads

Gradual ramps

Events with known traffic

Ongoing monitoring



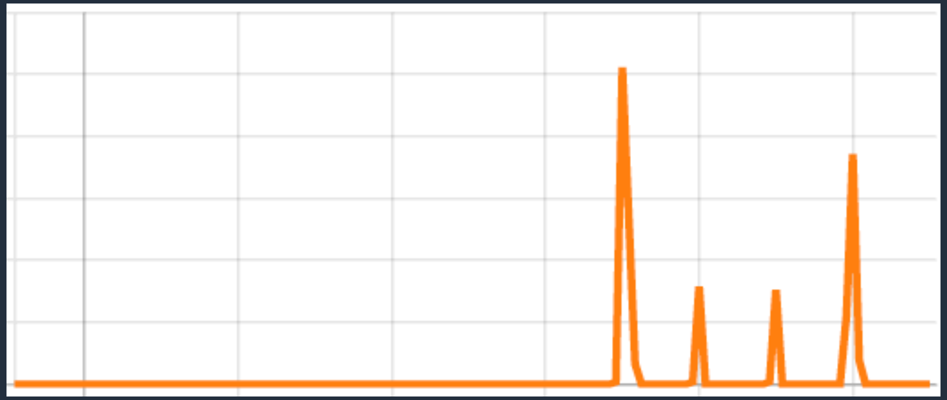
Use on-demand mode

Unpredictable workloads

Frequently idle workloads

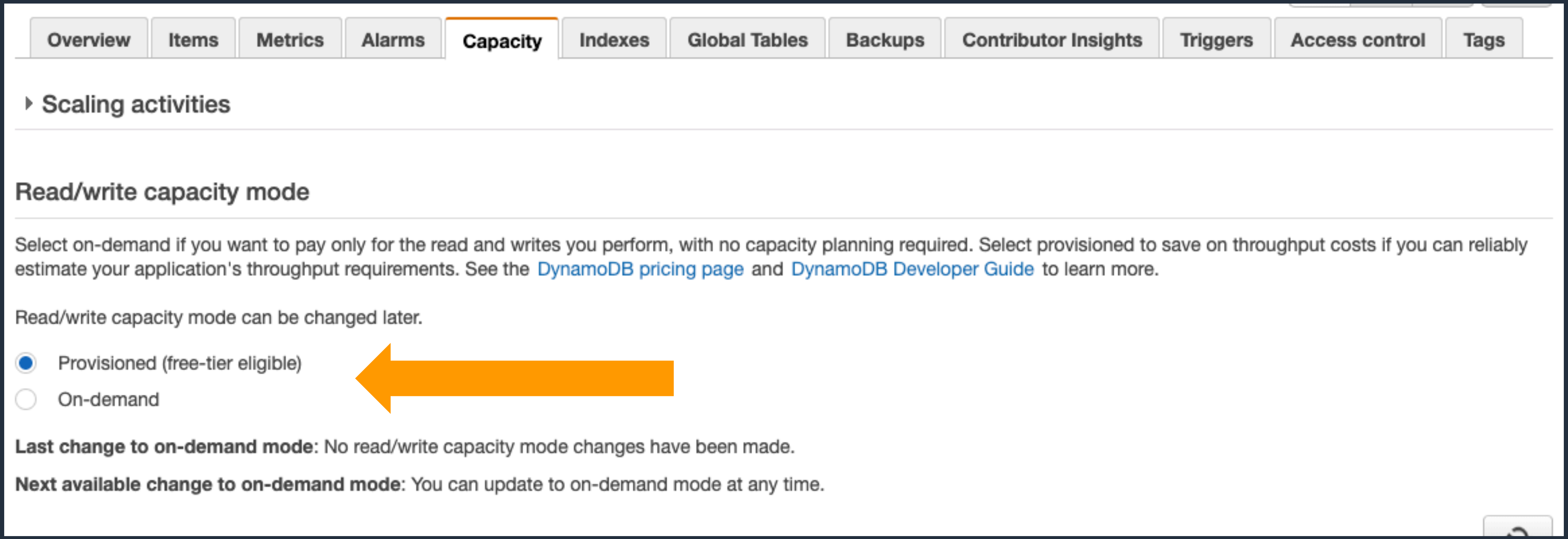
Events with unknown traffic

“Set it and forget it”



Consider your tolerance for operational overhead and overprovisioning

Changing capacity mode



Read/write capacity mode

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the DynamoDB pricing page and DynamoDB Developer Guide to learn more.

Read/write capacity mode can be changed later.

● Provisioned (free-tier eligible)

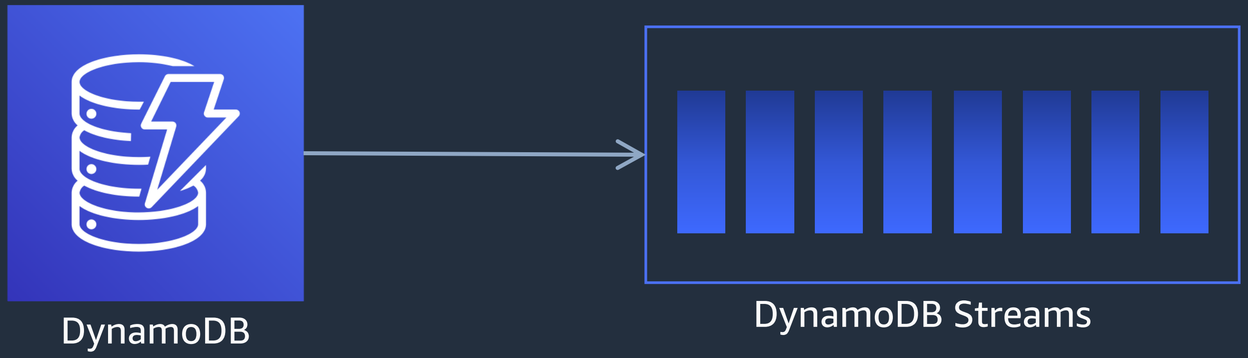
○ On-demand

Last change to on-demand mode: No read/write capacity mode changes have been made.

Next available change to on-demand mode: You can update to on-demand mode at any time.

Change data capture with DynamoDB

DynamoDB Streams



DynamoDB Streams

- ✓ Stream of item changes
- ✓ Exactly once, guaranteed delivery
- ✓ Strictly-ordered by key
- ✓ Durable, scalable
- ✓ Fully-managed
- ✓ 24-hour data retention
- ✓ Sub-second latency
- ✓ Event source for Lambda

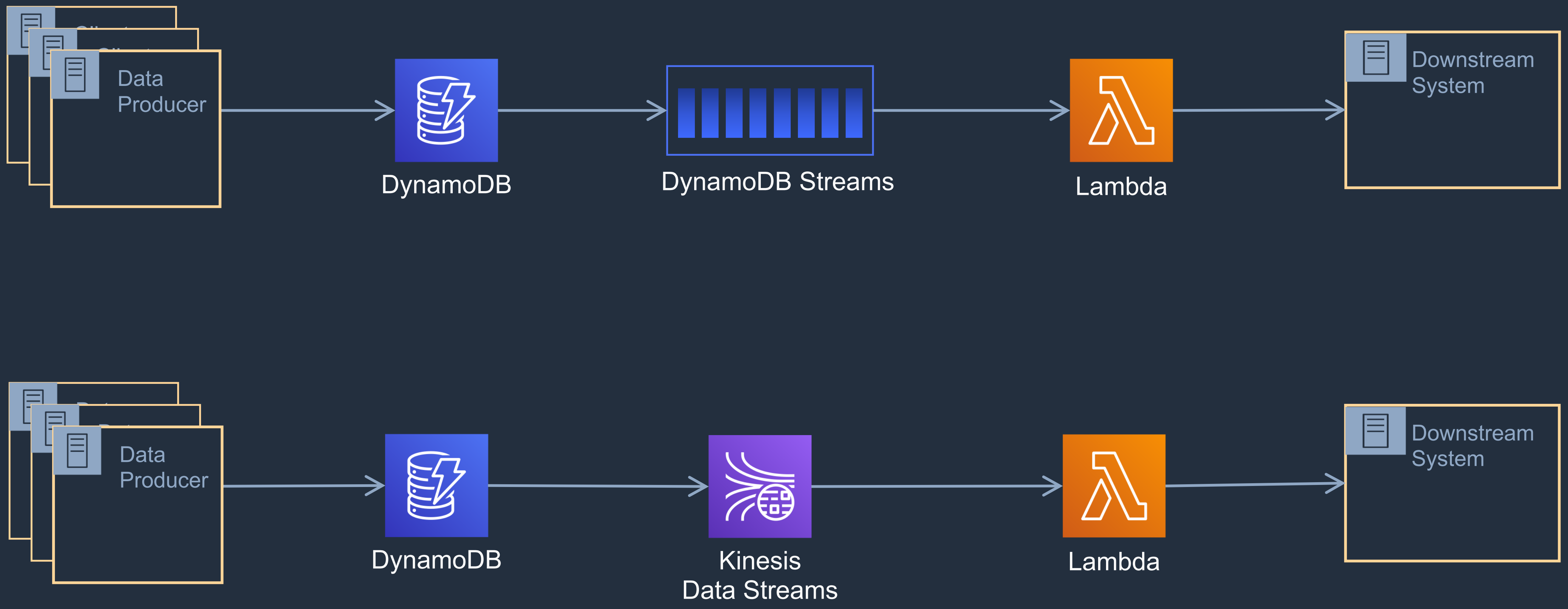
Amazon Kinesis Data Streams for DynamoDB

- Capture item-level changes in your DynamoDB tables as a Kinesis data stream.

- Build real-time applications by using Amazon Kinesis Data Analytics, and stream processing frameworks such as Apache Spark or your code by using the latest versions of the Kinesis Client Library.

- Use Amazon Kinesis Data Firehose to transform and send streaming data to destinations such as Amazon Elasticsearch Service, Amazon Redshift, Amazon S3, and Splunk.

Item-level change data capture



Global tables

Global tables provide apps with multi-Region replication

Build high-performance, globally distributed applications

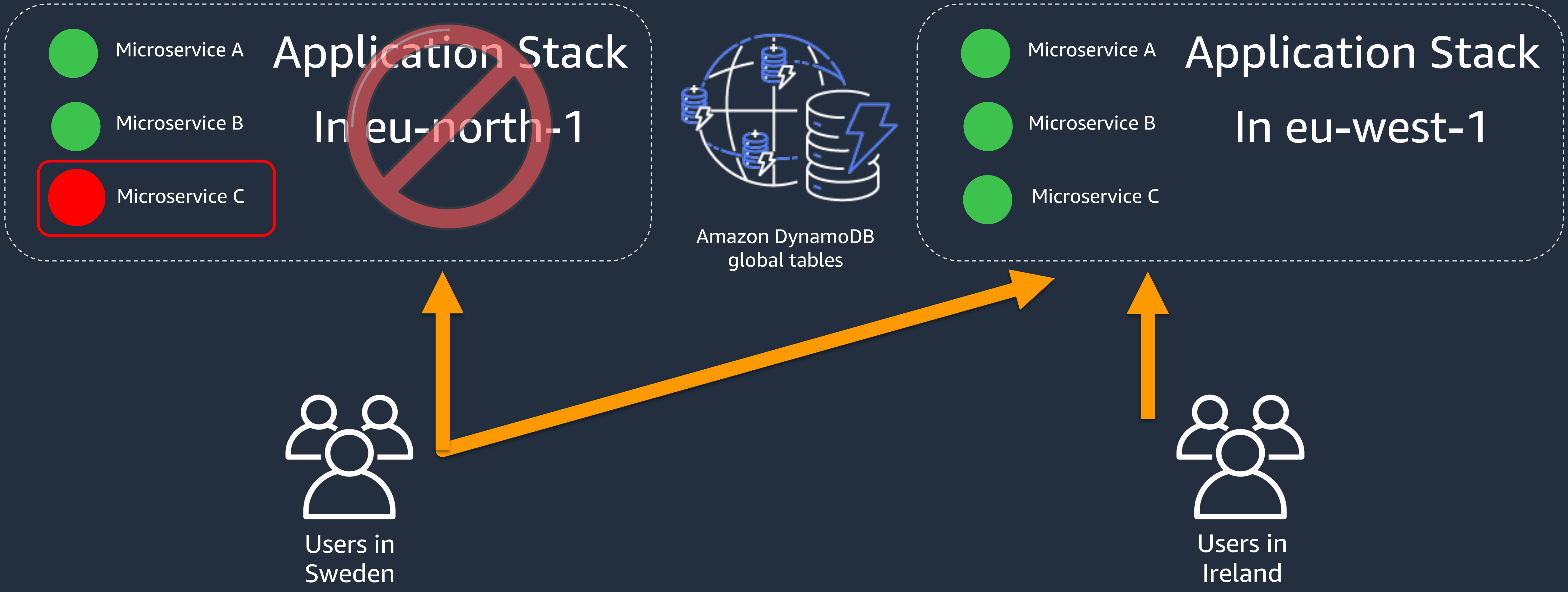
Low-latency reads and writes to locally available tables

Multi-Region redundancy and resiliency and 99.999% availability

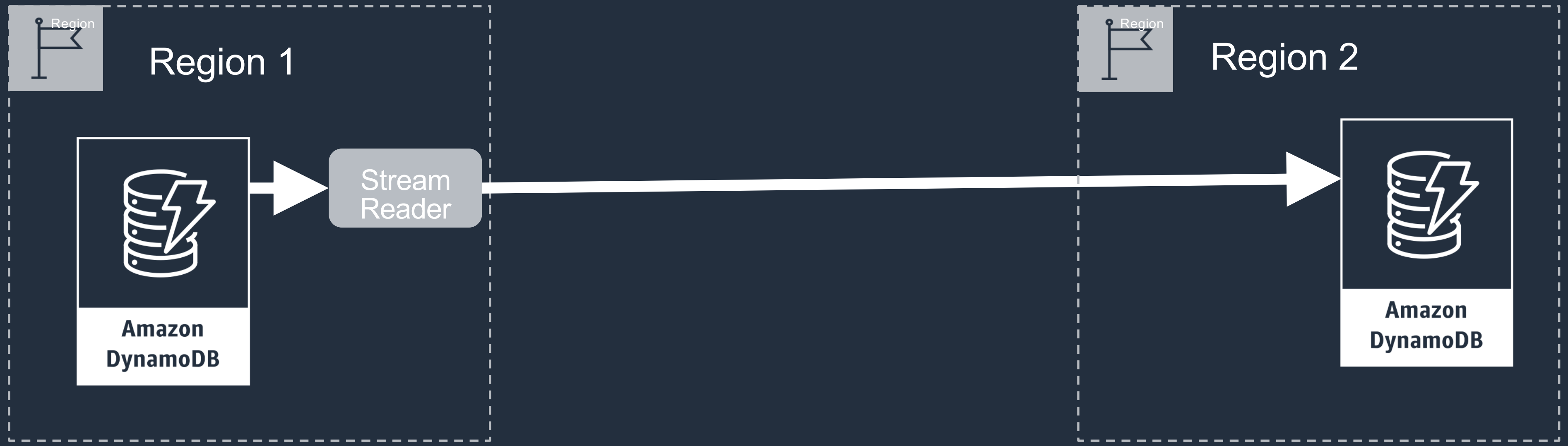
Multi-active writes from any Region

Easy to set up and no application rewrites required

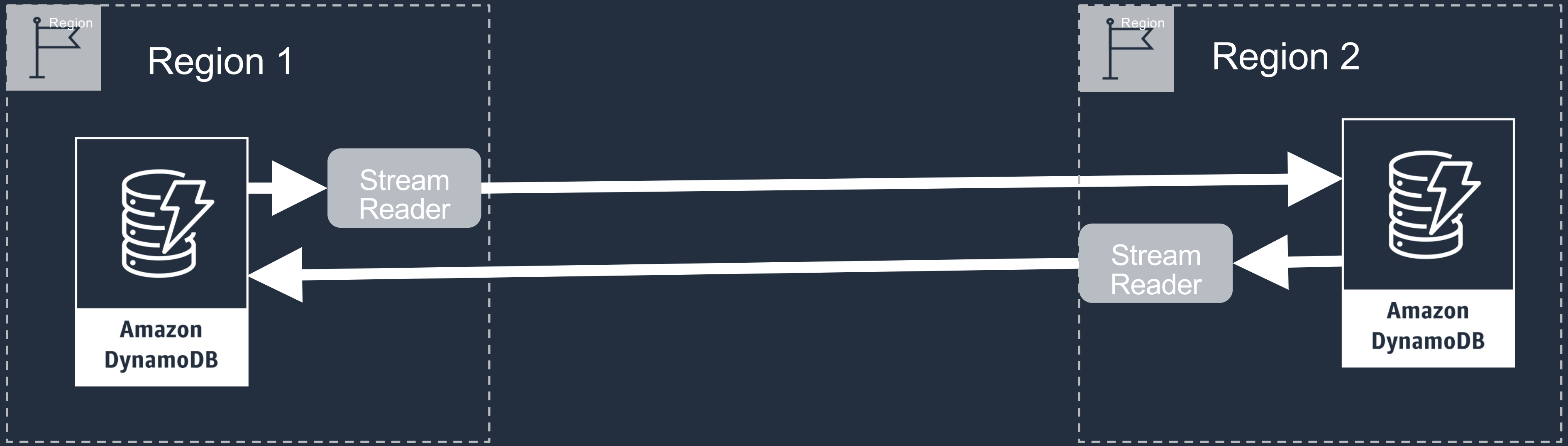
Global tables: architecture beyond the data



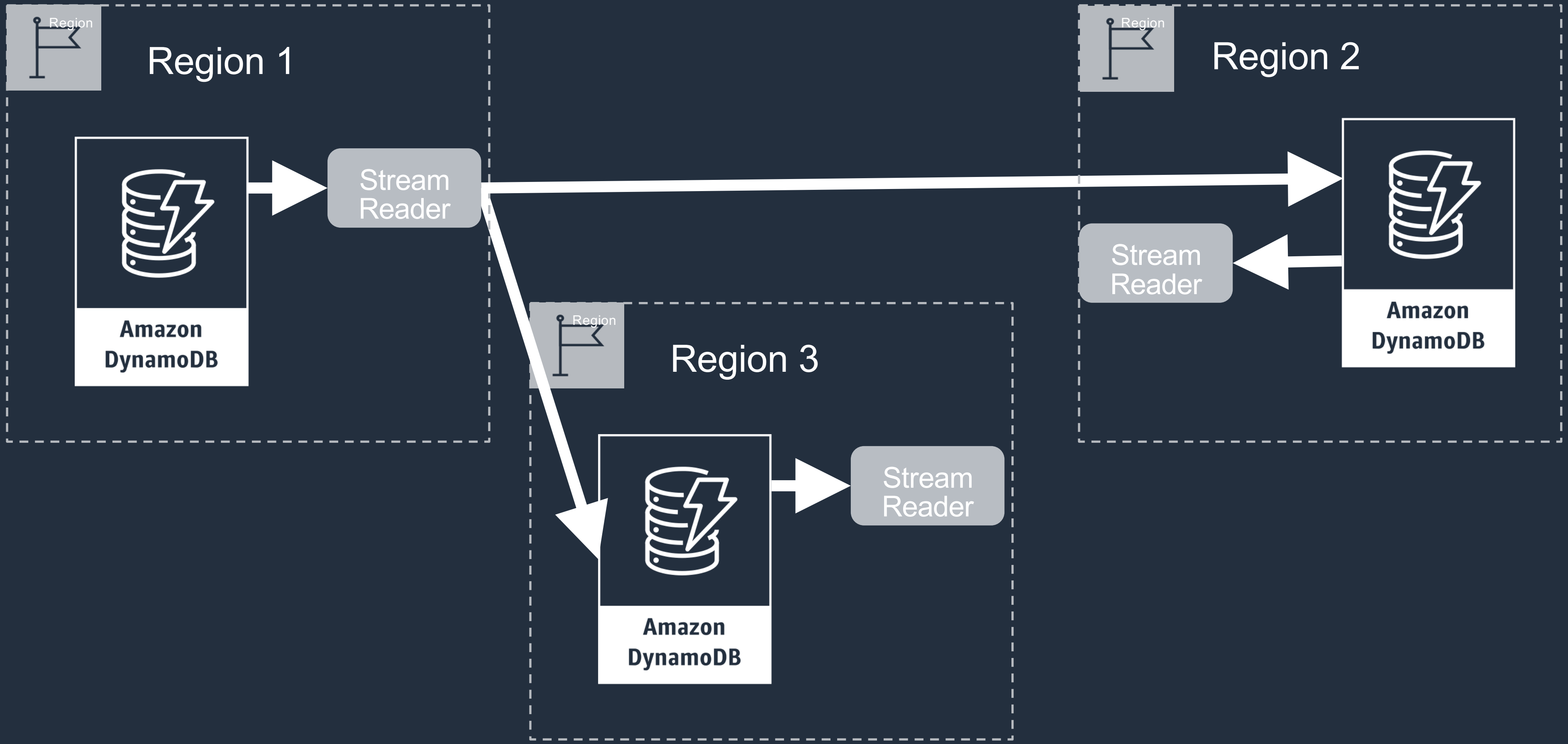
Global tables: high level architecture



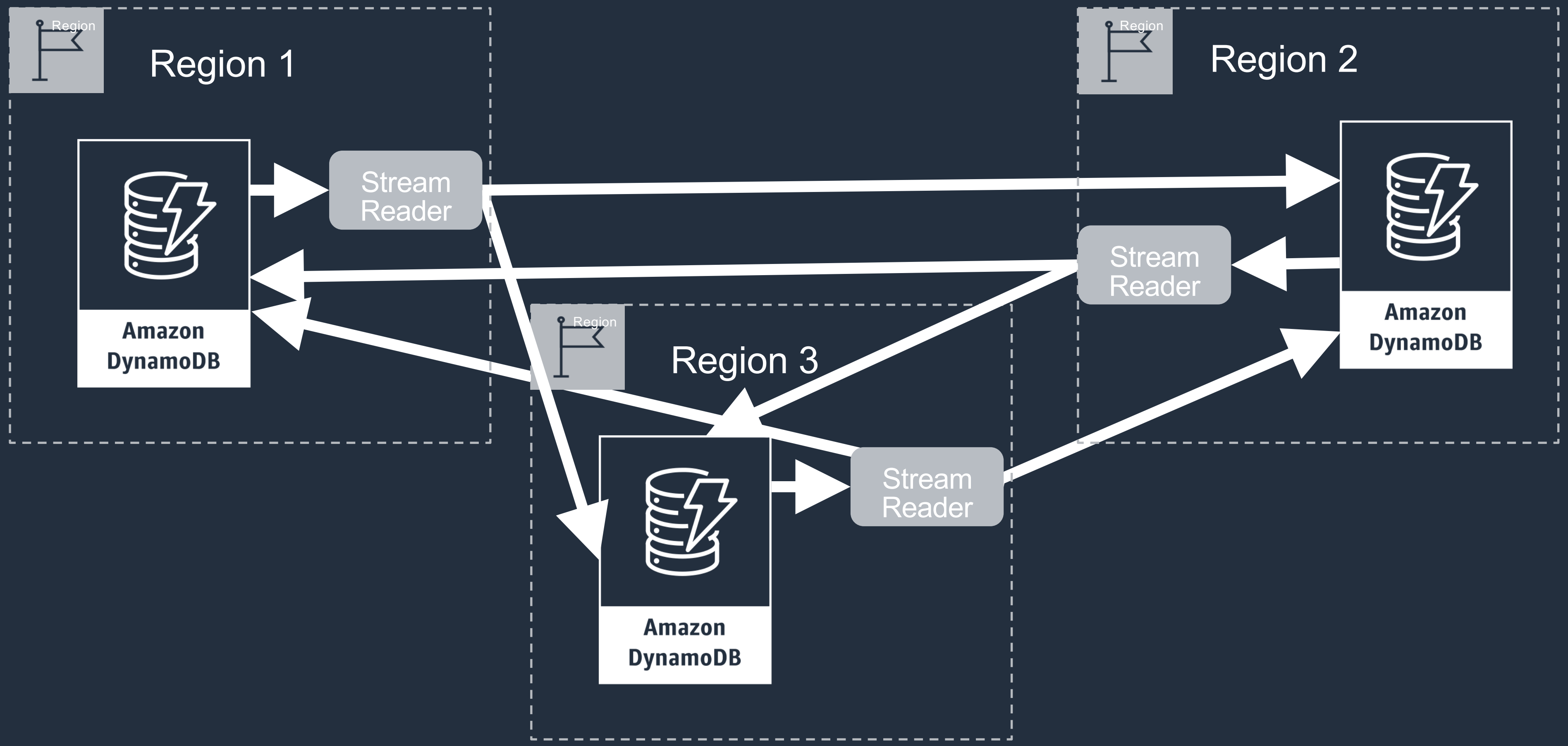
Global tables: multi-active replication



Global tables: multi-Region replication

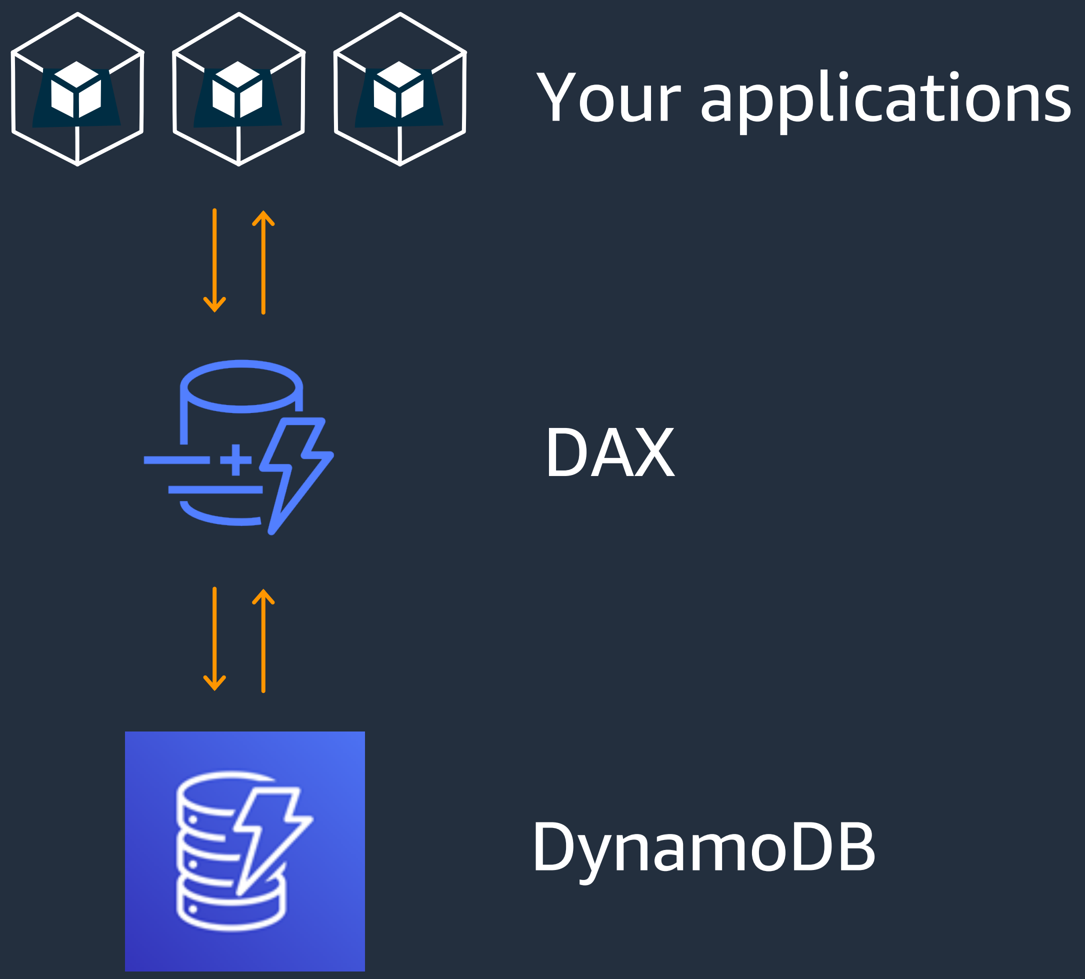


Global tables: multi-Region multi-active replication



DynamoDB Accelerator (DAX)

DAX: read cache



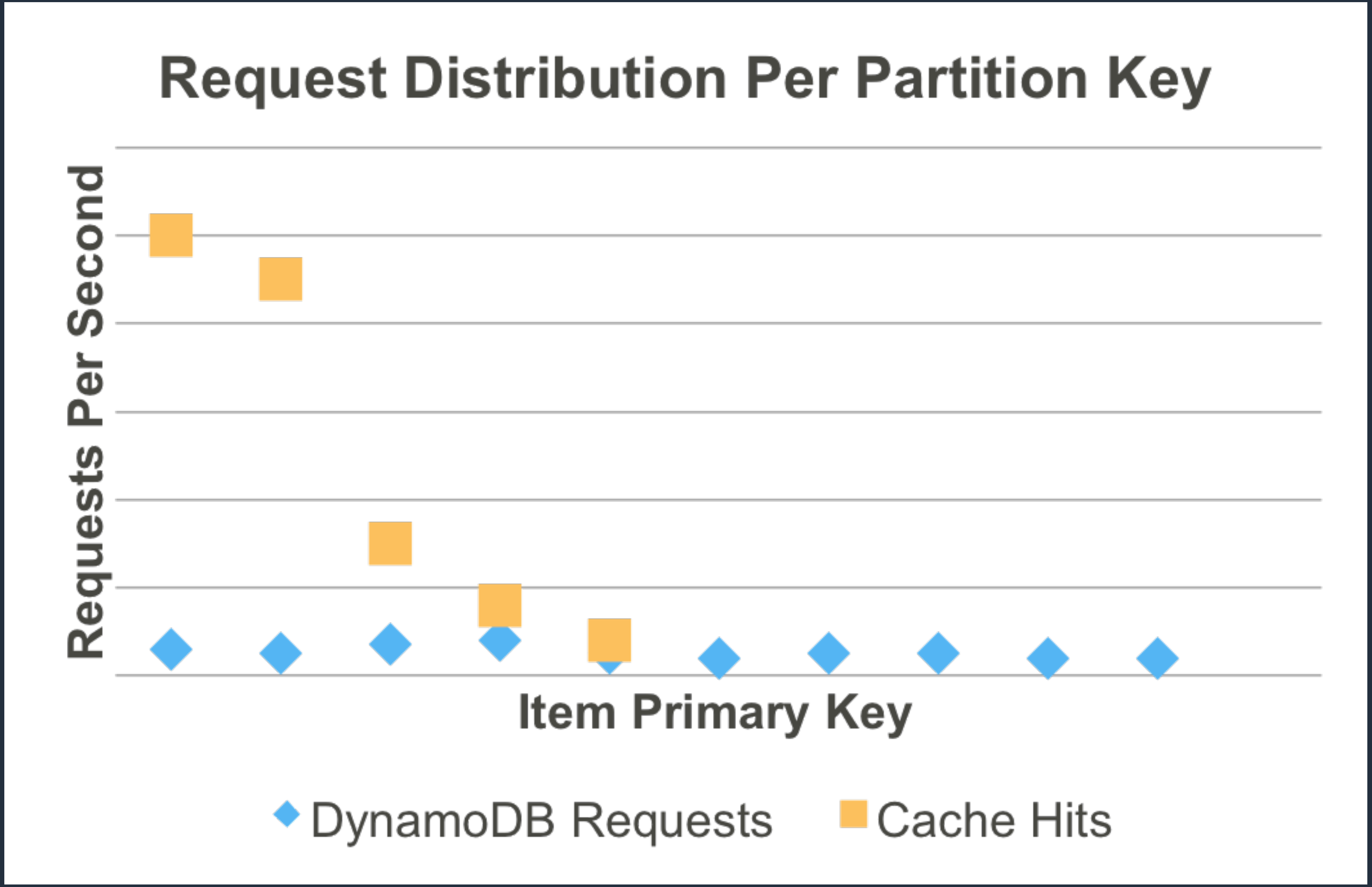
Fully managed,
highly available cache
for DynamoDB

Even faster—microsecond latency

Scales to millions of read requests per second

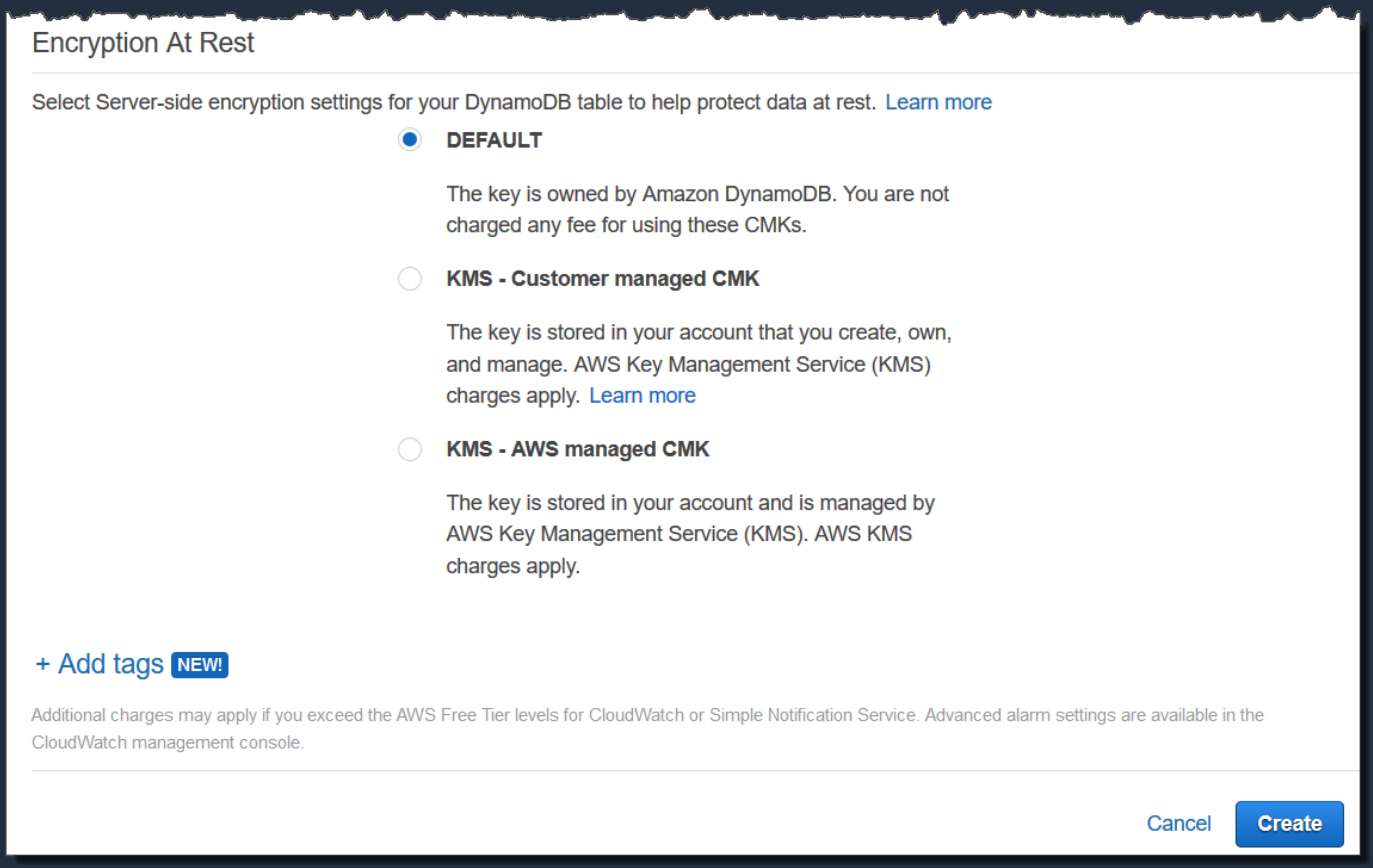
API compatible

DAX: reduced reads from DynamoDB



DynamoDB is enterprise ready

Security



Encryption At Rest

Select Server-side encryption settings for your DynamoDB table to help protect data at rest. Learn more

● DEFAULT

The key is owned by Amazon DynamoDB. You are not charged any fee for using these CMKs.

○ KMS - Customer managed CMK

The key is stored in your account that you create, own, and manage. AWS Key Management Service (KMS) charges apply. Learn more

○ KMS - AWS managed CMK

The key is stored in your account and is managed by AWS Key Management Service (KMS). AWS KMS charges apply.

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Fully integrated with AWS Identity and Access Management (IAM)

Access DynamoDB via secure Amazon VPC endpoints

All tables encrypted in transit, at rest by default

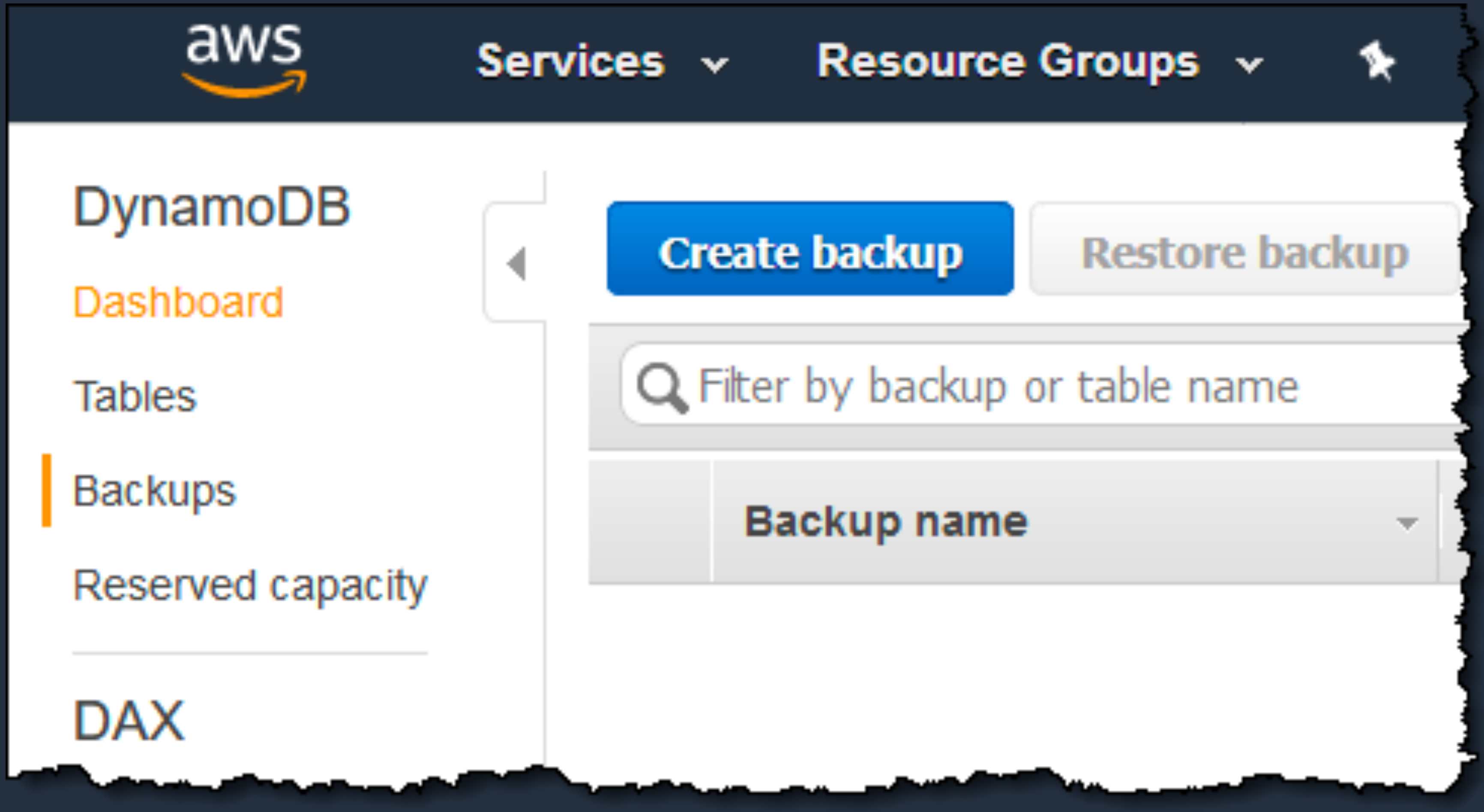
Native, server-side support for ACID transactions

Simplify application code with ACID guarantees

Run transactions for large-scale workloads

Accelerate legacy migrations

Backup and restore

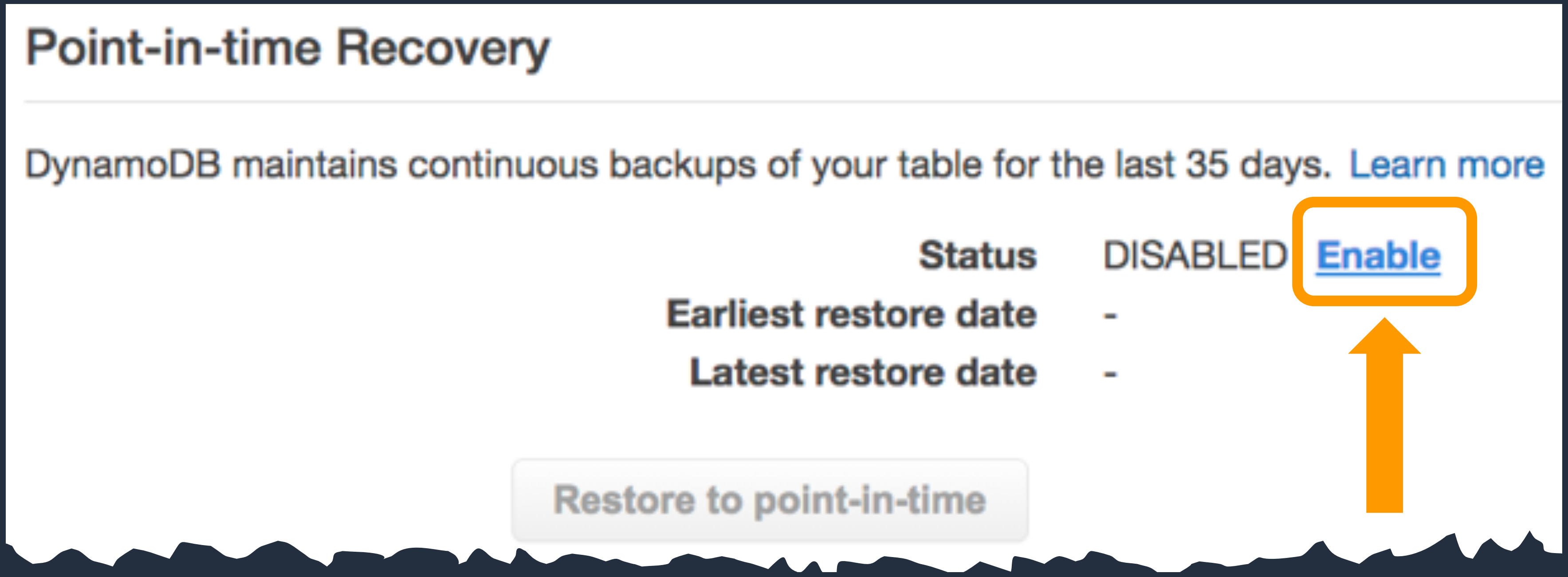


On-demand backups for long-term data archiving and compliance

Continuous backups for point-in-time recovery (PITR)

Back up PBs of data instantly with no performance impact

Point-in-time recovery



Point-in-time Recovery

DynamoDB maintains continuous backups of your table for the last 35 days. Learn more

Status	DISABLED Enable
Earliest restore date	-
Latest restore date	-

35-day rolling window of protection

Restore to a new table with second granularity

No impact to ongoing operations

Covers table deletion and corruption risks

Export DynamoDB data to S3 for analysis and insights

Extract actionable insights

Export DynamoDB table data to your data lake in Amazon S3, and use other AWS services to analyze data and highlight key takeaways.

Integrate with backups

To export, select a DynamoDB table that has point-in-time recovery (PITR) enabled, specify any point in the last 35 days, and choose the target Amazon S3 bucket. The output data formats supported are DynamoDB JSON and Amazon Ion.

Work across Regions

Export data to S3 across AWS Regions and accounts to help comply with regulatory requirements, and to develop a disaster recovery and business continuity plan.

No impact on performance

Does not consume table capacity, and has zero impact on performance and availability. All DynamoDB data added to your Amazon S3 data lake is easily discoverable, encrypted at rest and in transit, and retained in your S3 bucket until you delete it.

PartiQL supported on DynamoDB for easier queries

Easier queries

You can now use PartiQL (a SQL-compatible query language) to query, insert, update, and delete table data in the DynamoDB console.

Consistent performance

With PartiQL, DynamoDB continues to provide consistent, single-digit-millisecond latency at any scale. You can expect the same availability, latency, and performance when performing DynamoDB operations.

Improved productivity

Because PartiQL is supported for all data-plane operations, developers can use a familiar, structured query language to perform these operations.

Supported across AWS Regions

PartiQL for DynamoDB is now supported in 23 AWS Regions.

CloudWatch Contributor Insights for DynamoDB

CloudWatch Contributor Insights for DynamoDB

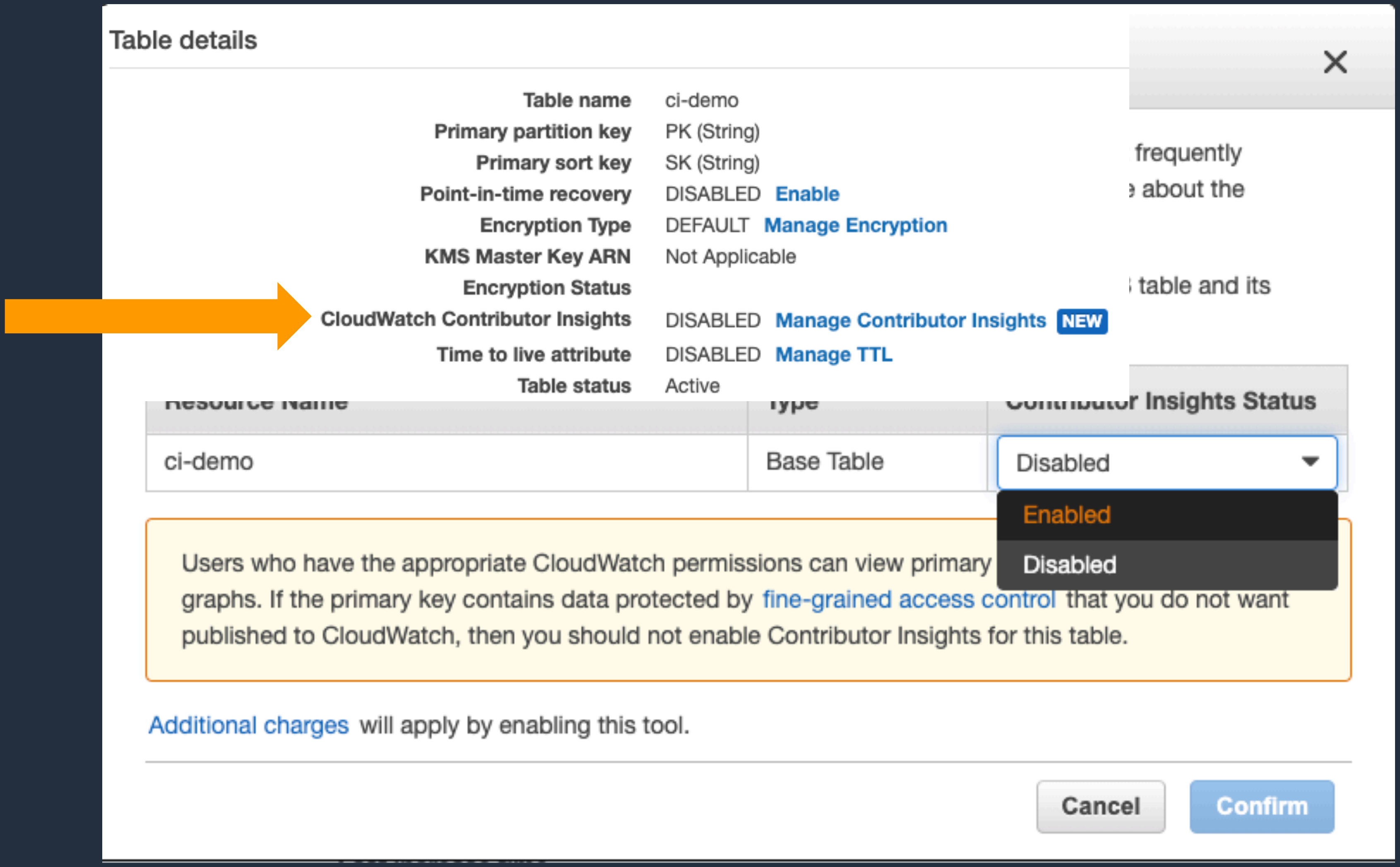


Table details

Table name	ci-demo
Primary partition key	PK (String)
Primary sort key	SK (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	DEFAULT Manage Encryption
KMS Master Key ARN	Not Applicable
Encryption Status	
CloudWatch Contributor Insights	DISABLED Manage Contributor Insights NEW
Time to live attribute	DISABLED Manage TTL
Table status	Active

frequently

about the

table and its

Resource Name	Type	Contributor Insights Status
ci-demo	Base Table	Disabled

Users who have the appropriate CloudWatch permissions can view primary graphs. If the primary key contains data protected by fine-grained access control that you do not want published to CloudWatch, then you should not enable Contributor Insights for this table.

Additional charges will apply by enabling this tool.

CloudWatch Contributor Insights for DynamoDB

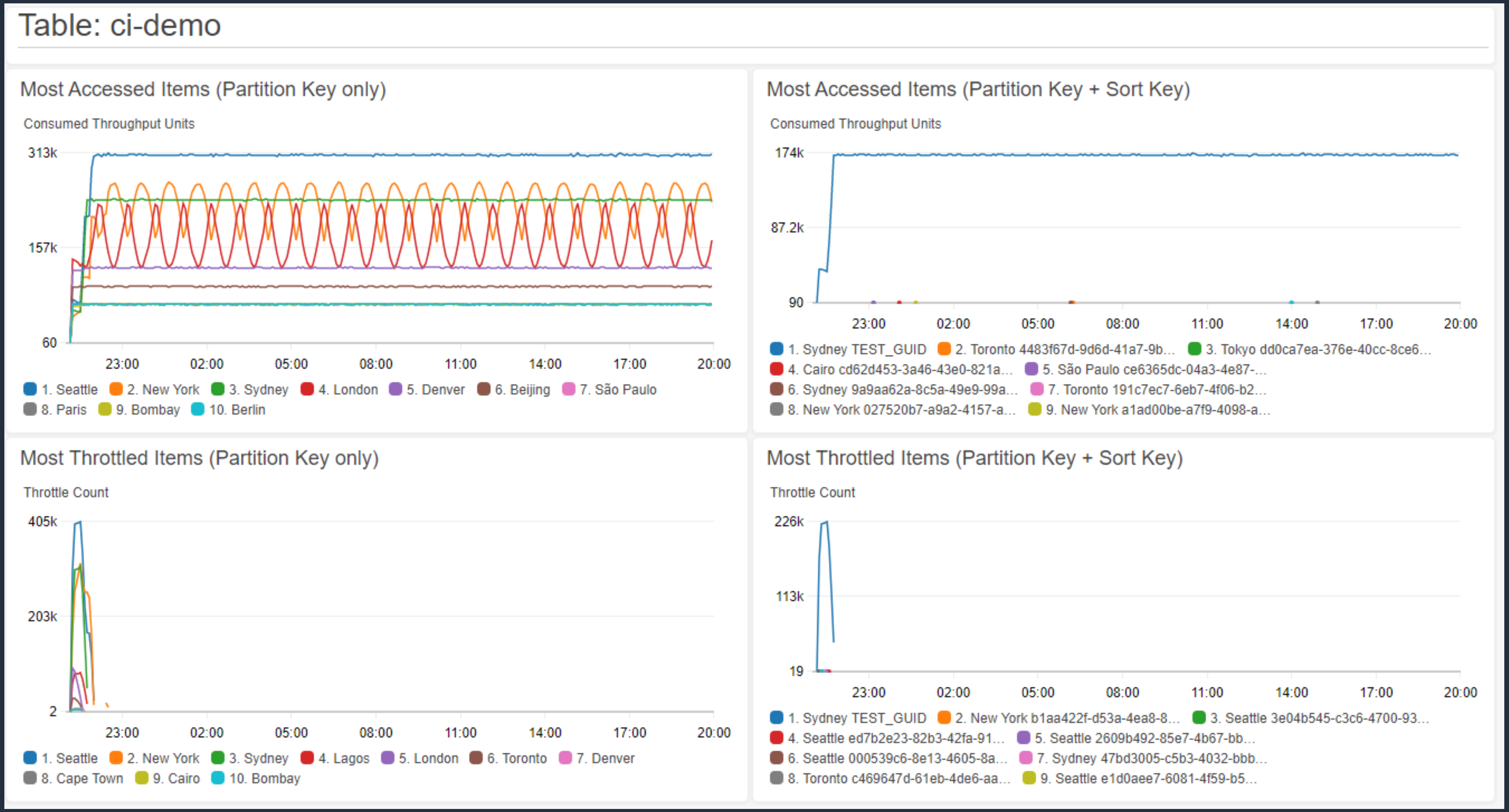


Table: ci-demo

Most Accessed Items (Partition Key only)

Consumed Throughput Units
313k
157k
60
23:00
02:00
05:00
08:00
11:00
14:00
17:00
20:00
1. Seattle
2. New York
3. Sydney
4. London
5. Denver
6. Beijing
7. São Paulo
8. Paris
9. Bombay
10. Berlin

Most Accessed Items (Partition Key + Sort Key)

Consumed Throughput Units
174k
87.2k
90
23:00
02:00
05:00
08:00
11:00
14:00
17:00
20:00
1. Sydney TEST_GUID
2. Toronto 4483f67d-9d6d-41a7-9b...
3. Tokyo dd0ca7ea-376e-40cc-8ce6...
4. Cairo cd62d453-3a46-43e0-821a...
5. São Paulo ce6365dc-04a3-4e87-...
6. Sydney 9a9aa62a-8c5a-49e9-99a...
7. Toronto 191c7ec7-6eb7-4f06-b2...
8. New York 027520b7-a9a2-4157-a...
9. New York a1ad00be-a7f9-4098-a...

Most Throttled Items (Partition Key only)

Throttle Count
405k
203k
2
23:00
02:00
05:00
08:00
11:00
14:00
17:00
20:00
1. Seattle
2. New York
3. Sydney
4. Lagos
5. London
6. Toronto
7. Denver
8. Cape Town
9. Cairo
10. Bombay

Most Throttled Items (Partition Key + Sort Key)

Throttle Count
226k
113k
19
23:00
02:00
05:00
08:00
11:00
14:00
17:00
20:00
1. Sydney TEST_GUID
2. New York b1aa422f-d53a-4ea8-8...
3. Seattle 3e04b545-c3c6-4700-93...
4. Seattle ed7b2e23-82b3-42fa-91...
5. Seattle 2609b492-85e7-4b67-bb...
6. Seattle 000539c6-8e13-4605-8a...
7. Sydney 47bd3005-c5b3-4032-bbb...
8. Toronto c469647d-61eb-4de6-aa...
9. Seattle e1d0aee7-6081-4f59-b5...

NoSQL Workbench

NoSQL Workbench

Data modeler

Visualizer

Operation builder

A **client-side application** that helps you build scalable, high-performance data models

Simplifies query development and testing

A rich GUI-based tool that **helps you visualize data models** and **perform DynamoDB operations**

Available for **Windows, macOS, and Linux**

NoSQL Workbench

To download NoSQL Workbench:

Step-by-step data modeling examples:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.html

https://github.com/aws-samples/amazon-dynamodb-design-patterns

Thank you!

Visit DynamoDB.com

Follow @DynamoDB on 🐦