



## Read Consistency in DynamoDB

### 1 Read consistency

- 1 As DynamoDB uses a distributed storage system, the way it reads and writes data is different from traditional databases like RDS that use regular storage
- 2 In distributed storage, data is spread across multiple servers and partitions, which makes managing reads and writes more complex
- 3 To handle this, DynamoDB uses two key concepts
  - 1 Read Consistency
  - 2 Write Consistency

- 1 Read consistency in DynamoDB decides how accurate and up-to-date the data is when you read it from a distributed system
- 2 Since DynamoDB replicates data across multiple nodes, it offers different levels of consistency to balance between accuracy, speed, and cost

- 1 Introduction
- 3 There are three types of read consistency in DynamoDB
  - 1 Eventually Consistent Reads
  - 2 Strongly Consistent Reads
  - 3 Transactional Reads

- 1 Data Accuracy May return slightly outdated data from Replica Nodes
- 2 Read Scope Single item per operation
- 3 Speed Faster, as it retrieves data from the nearest replica
- 4 Use Case Applications where slight delays in updates are acceptable (e.g., social media, catalogs)
- 5 Consistency Guarantee Eventually consistent across replicas
  - 1 Less compute power
- 6 Compute Power Needed
  - 2 Less compute power because it reads from Replica Nodes, avoiding the need for the Leader Node
  - 3 Replica Node Reads: Less compute power as replicas are optimized for read performance and don't handle consistency validations

- 1 Data Accuracy Always returns the latest data from the Leader Node
- 2 Read Scope Single item per operation
- 3 Speed Slower, as it ensures the most recent data
- 4 Use Case Critical applications needing accurate data immediately (e.g., banking, inventory systems)
- 5 Consistency Guarantee Guaranteed to be the latest
  - 1 Higher Compute Power
- 6 Compute Power Needed
  - 2 Higher compute power because the data must be fetched from the Leader Node, which is the most accurate source
  - 3 Leader Node Reads: More compute power because the Leader Node must ensure the latest state, balancing ongoing writes and read requests

- 1 Data Accuracy Ensures all items reflect the same consistent state
- 2 Read Scope Multiple items in a single operation
- 3 Speed Slightly slower due to transaction overhead
- 4 Use Case Multi-item consistency (e.g., order details and inventory together)
- 5 Consistency Guarantee Strong consistency across multiple items
  - 1 Most compute power
- 6 Compute Power Needed
  - 2 Most compute power because it must coordinate and validate consistency across multiple items or tables
  - 3 Transactional Reads need the most compute power because they ensure all items are consistent together

- 1 Introduction
  - 1 RCU stands for Read Capacity Unit, which measures how much data your DynamoDB table can read per second
  - 2 The type of read consistency you choose determines how your allocated RCUs will be used when performing read operations on your DynamoDB table
  - 2 RCUs ensure that your application has the necessary resources to handle its read operations efficiently
    - 1 Replica Node Reads (Eventually Consistent) consume fewer RCUs as they skip additional consistency checks
    - 2 Leader Node Reads (Strongly Consistent) consume more RCUs because they require more compute power to ensure accuracy
    - 3 Transactional Reads consume the most RCUs as they ensure consistency across multiple items
- 3 What is RCU?