



## Warm Throughput

1

### Introduction

- 1 Warm throughput means the extra capacity DynamoDB keeps ready so your table can handle sudden increases in read and write operations instantly
- 2 These values are automatically set for all tables and global secondary indexes (GSI) based on how much the table has scaled in the past
- 3 It ensures your table performs smoothly without delays or throttling during high usage

2

### How it works

- 1 Pre-Provisioned Resources
  - 1 When warm throughput is configured
  - 2 DynamoDB reserves additional backend resources (e.g., partitions, storage, compute) in advance
  - 3 These resources are immediately available when a spike in traffic occurs
- 2 Capacity Allocation
  - 1 DynamoDB scales read/write units to the pre-set warm throughput value
  - 2 It ensures that sufficient partitions and processing power are allocated in the background before traffic arrives
  - 3 Default Mode
    - 1 On-Demand
      - 1 Default Reservation DynamoDB reserves 12,000 read and 4000 write request units per second as the warm throughput baseline
      - 2 Charges You pay for the Warm Throughput regardless of actual usage
    - 2 Provisioned Capacity Mode
      - 1 Default Reservation In provisioned mode, DynamoDB reserves the same amount of capacity as you configure for your table
      - 2 Charges You pay for the Warm Throughput regardless of actual usage
  - 4 Increase warm throughput Manually
    - 1 On-Demand
      - 1 Warm throughput in on-demand mode ensures that DynamoDB reserves additional capacity in the backend to handle sudden traffic spikes
      - 2 Even though On-Demand scales automatically, pre-allocating resources helps reduce latency during scaling events
      - 3 Charges
        - 1 You pay a fixed fee for reserving warm throughput, even if you don't use it
        - 2 Along with this, you also pay for the actual reads and writes based on usage in On-Demand mode
    - 2 Provisioned Capacity Mode
      - 1 When you manually increase warm throughput in provisioned mode, you reserve additional capacity (RCUs/WCUs) to handle traffic spikes.
      - 2 Charges
        - 1 When you manually increase warm throughput in provisioned mode, you reserve additional capacity (RCUs/WCUs) to handle traffic spikes.
        - 2 As long as you don't manually increase the warm throughput beyond the default, there are no one-time setup charges
        - 2 You pay for the Warm Throughput regardless of actual usage
  - 3 Traffic Handling
    - 1 When a sudden surge in traffic occurs (e.g., due to a sale or marketing campaign), the pre-warmed capacity is instantly utilized without delays
    - 2 This prevents throttling or scaling lag, as the system does not need to reactively allocate resources
    - 3 Example If you manually set warm throughput to 100 RCUs
      - 1 If you have 50 provisioned RCUs/WCUs and increase to 100 as warm throughput, DynamoDB reserves the full 100 units for your table, meaning you can always utilize up to 100 when needed
      - 2 When traffic spikes to 100 RCUs, the system handles it instantly without needing to scale up reactively
      - 3 Charges
        - 1 If you provision 100 RCUs and set up an additional 10 RCUs for warm throughput, here's how the charges work
        - 2 Provisioned Throughput Charges (100 RCUs) You always pay for the 100 RCUs, regardless of whether they are used or not
        - 3 Warm Throughput Charges (10 RCUs)
          - 1 One-Time Charge When you manually configure an extra 10 RCUs for warm throughput, there is a one-time charge for setting up this additional capacity
          - 2 Usage-Based Cost
            - 1 After configuration, you are only charged for the extra 10 RCUs if they are actively used to handle traffic
            - 2 If the warm throughput is not utilized, you don't pay for it during regular operations

3

### What can happen without warm throughput

- 1 Delays in Scaling
  - 1 DynamoDB doesn't instantly scale up
  - 2 During the time it takes to increase capacity, your app might slow down or show errors
- 2 Throttling If traffic exceeds the current capacity before scaling completes, requests may be rejected, causing failed operations
- 3 Missed Opportunities High-traffic events, like a sale or a big launch, might lead to poor user experience, loss of trust, and missed revenue because your app couldn't handle the surge in traffic effectively
- 4 Example
  - 1 Imagine running a sale, and 10,000 people try to access your app at once
  - 2 Without warm throughput, DynamoDB may not scale up instantly, causing delays, timeouts, or even crashes during the peak load