Configure Provisioned Concurrency in Lambda

- 1 Introduction — Concurrency
  - 1 The number of instances of your Lambda function that are running at the same time to handle multiple requests
  - 2 Example — If 10 users hit your Lambda at once, AWS will spin up 10 instances of your function — one for each request
  - 3 Types of Concurrency in Lambda
    - 1 On-Demand Concurrency
    - 2 Provisioned Concurrency
- 2 On-Demand Concurrency
  - 1 Introduction
    - 1 Automatically spins up environments when requests arrive.
    - 2 Default mode
  - 2 What Happens in On-Demand?
    - 1 A request comes → AWS looks for an existing warm instance
    - 2 If no instance is available → AWS spins up a new one (this causes a delay)
    - 3 After the function is ready → it handles the request
    - 4 If another request comes → AWS may reuse the same instance or create another
  - 3 Cold Start Problem in On-Demand Concurrency
    - 1 A Cold Start occurs when
      - 1 AWS has to create a new environment for a function that hasn't run recently
      - 2 When your function scales suddenly and needs new instances
      - 3 Cold start time — ~100ms to several seconds
    - 2 Real-life problem
      - 1 My Lambda is working fine... but the first request takes time, and after that it becomes fast.
      - 2 That's the cold start problem in On-Demand
    - 3 Solution — Provisioned Concurrency
- 3 Provisioned Concurrency
  - 1 Introduction
    - 1 AWS Lambda's Provisioned Concurrency pre-initializes a defined number of execution environments
    - 2 These environments stay active and ready, ensuring low-latency responses
    - 3 It's designed to eliminate cold starts for critical applications
  - 2 Setup Provisioned Concurrency
    - 1 Step 1: Publish a Lambda Version
      - 1 Provisioned Concurrency only works with published versions (not $LATEST)
      - 2 Go to Lambda → Your Function
      - 3 Click "Publish new version"
      - 4 Give an optional description → Click Publish
    - 2 Step 2: (Optional but Recommended) Create an Alias
      - 1 Aliases make it easier to manage versions with Provisioned Concurrency
      - 2 Go to your Lambda function
      - 3 Click "Aliases" → Create alias
      - 4 Alias name: live, Version: 1 (or whatever you published)
    - 3 Step 3: Set Provisioned Concurrency
      - 1 Now, assign provisioned concurrency to that version or alias
      - 2 Go to "Aliases" tab → Choose your alias
      - 3 Click "Concurrency" tab → Enable "Provisioned Concurrency"
      - 4 Set the number (e.g., 5) → Click Save
    - 4 Step 4: Monitor with CloudWatch
      - 1 ProvisionedConcurrencyUtilization
      - 2 Throttles
- 4 SAA-C03 Exam Notes
  - 1 Provisioned Concurrency does not work with $LATEST
  - 2 Make sure your account has enough concurrency quota
  - 3 You pay for the number of provisioned instances per minute, whether used or not