

@devopschallengehub



---

## Lambda Miscellaneous Interview Questions

### What is the difference between synchronous and asynchronous invocation?

- **Synchronous invocation:** the caller waits for the function result (example: API Gateway → Lambda). Caller receives the function response or an error.
- **Asynchronous invocation:** the caller receives an immediate acceptance and Lambda processes the event later; Lambda handles retries and can route failures to a **Dead-Letter Queue (DLQ)** or failure destination. Asynchronous flows are useful for decoupling and smoothing bursts.

Synchronous: **Web applications** A web app calls a backend Lambda to validate login credentials or process form data.

### S3 → Lambda Asynchronous

- When a file is uploaded to S3, an event triggers Lambda to process it (e.g., create a thumbnail, extract metadata).

---

### Explain Lambda execution environment / cold start vs warm start.

Lambda runs functions inside short-lived execution environments (containers).

- **Cold start:** happens when Lambda creates a new execution environment — OS + runtime + function initialization occurs and adds latency. Languages with heavy runtimes (Java/.NET) and large deployment packages typically see longer cold starts. VPC attachments (ENI creation) can also increase cold-start time.
- **Warm start:** reusing an already-initialized environment is faster because initialization was already done.
- **Mitigation:** reduce package size, use lighter runtimes, avoid unnecessary VPC attachments, or use **provisioned concurrency** to keep environments initialized.

---

### What are Lambda limits you should be aware of?

Key limits to design around:

- **Timeout:** max execution time (currently 15 minutes).
- **Memory:** configurable per function (affects CPU and cost).

- **Deployment package:** limits for zipped packages and container image sizes (container images allow larger bundles).
- **Ephemeral storage:** /tmp storage available to function (commonly ~512 MB by default — verify for your account/region).
- Also watch concurrency limits (account and reserved), package unzipped size, and environment variable size limits.

---

#### How do you pass configuration or secrets to a Lambda?

- **Environment variables:** simple config values (can be encrypted with KMS).
  - **AWS Secrets Manager:** store and rotate secrets; Lambda fetches at runtime (or caches securely).
  - **AWS Systems Manager (SSM) Parameter Store:** store secure strings and configuration parameters.
  - Always follow least-privilege IAM so the function role can only read required secrets/parameters.
- 

---

#### What is a Lambda layer? When do you use it?

A **Lambda layer** is a way to package and share common libraries, binaries, or runtime extensions separately from your function code.

- **When to use:** share dependencies across multiple functions, reduce each function's deployment package size, or include native libraries compiled for the Lambda runtime.
  - Layers promote reuse and faster deployments but manage versioning carefully.
- 


---

#### How do you debug a Lambda function?

- **CloudWatch Logs:** primary place to inspect stdout/stderr and structured logs.
  - **CloudWatch Logs Insights:** query logs at scale.
  - **AWS X-Ray:** distributed tracing for latency hotspots and downstream calls.
  - **Local debugging / SAM CLI:** run and step-through functions locally with sample events.
  - **Best practices:** structured logging, log levels, correlation IDs, add meaningful metrics and alarms, and include error context for tracing.
- 

#### Which is an example of synchronous Lambda invocation?

- A. S3 bucket upload event → Lambda creates thumbnails.
- B. SNS topic publishes a message → Lambda processes it.
- C. API Gateway request → Lambda validates login credentials.
- D. EventBridge scheduled rule triggers Lambda every 5 minutes.

C. API Gateway request → Lambda validates login credentials. 

---

#### Which is an example of asynchronous Lambda invocation?

- A. Lambda called using AWS CLI with invoke.
- B. API Gateway request → Lambda processes and returns data.

- C. S3 upload triggers Lambda to process the file.
- D. A web app calls Lambda for form validation.

C. S3 upload triggers Lambda to process the file. ✓

---

**What is a Lambda cold start?**

- A. When Lambda is running in low memory mode.
- B. When a new execution environment must be created, including runtime and initialization.
- C. When Lambda reuses an existing execution environment.
- D. When Lambda fails due to timeout.

B. When a new execution environment must be created, including runtime and initialization. ✓

---

**Which of the following increases cold start latency the most?**

- A. Using lighter runtimes like Node.js or Python.
- B. Small deployment packages.
- C. Heavy runtimes (Java/.NET) and VPC networking attachments.
- D. Provisioned concurrency enabled.

C. Heavy runtimes (Java/.NET) and VPC networking attachments. ✓

---

**What is the maximum timeout for a Lambda function?**

- A. 1 minute
- B. 5 minutes
- C. 10 minutes
- D. 15 minutes

D. 15 minutes ✓

---

**Which AWS service is best for securely storing and rotating database credentials for Lambda?**

- A. Environment variables
- B. AWS Secrets Manager
- C. SSM Parameter Store (SecureString)
- D. CloudWatch Logs

B. AWS Secrets Manager ✓

---

**What is the purpose of Lambda Layers?**

- A. To increase Lambda execution time.
- B. To store temporary files for execution.
- C. To share libraries and dependencies across multiple functions.
- D. To enable asynchronous invocation.

C. To share libraries and dependencies across multiple functions. ✓

---

**Which is the primary service to view Lambda logs?**

- A. AWS CloudTrail
- B. AWS CloudWatch Logs
- C. AWS Systems Manager
- D. AWS Trusted Advisor

B. AWS CloudWatch Logs 

---

### How can you reduce Lambda cold start issues?

- A. Use heavier runtimes like Java.
- B. Increase function timeout.
- C. Use provisioned concurrency, smaller packages, and lighter runtimes.
- D. Increase memory allocation only.

C. Use provisioned concurrency, smaller packages, and lighter runtimes. 

---

Normally, when a Lambda function is invoked **after being idle**, AWS has to create a **new execution environment** → this is the **cold start**.

Provisioned Concurrency is a feature that **keeps a pre-defined number of execution environments “warm” and ready to serve requests instantly**.

- You configure a number (e.g., 5, 10, 100).
- AWS keeps that many environments initialized at all times.
- When a request comes, Lambda **reuses one of these pre-warmed environments**, avoiding cold start latency.

### How it works

- You **enable provisioned concurrency on a published version or an alias**.
- Example:
  - `aws lambda put-provisioned-concurrency-config \`
  - `--function-name MyFunction \`
  - `--qualifier PROD \`
  - `--provisioned-concurrent-executions 5`
- AWS ensures **5 execution environments** are always initialized.