

Developing REST APIs

INTRODUCING APIS · API GATEWAY · CREATING REST APIS · INTEGRATIONS · DEPLOYING & STAGES · ACCESS
CONTROL · MONITORING · OPTIMIZING

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

- 01 API Gateway in one line:** **Amazon API Gateway** is a fully managed, serverless service to **create, publish, maintain, monitor, and secure** APIs at any scale — the *front door* between clients and your backends. It supports **REST, HTTP, and WebSocket** APIs, integrates with AWS services so you write no integration code, and bills only for **API calls received and data transferred out**.
- 02 REST API vs. HTTP API:** the **REST API** type is *advanced mode* — full control of requests and responses, mock endpoints, edge-optimized and private endpoints, resource policies, API keys and usage plans, and client SDK generation. The **HTTP API** type is a lightweight proxy with the **lowest cost (up to 70% savings) and latency**. Default to HTTP unless you need a REST-only feature or must monetize your API.
- 03 Deployments live on stages:** you must deploy an API to a **stage** (REST and WebSocket require it; an HTTP API auto-deploys a default stage). A stage is a named snapshot that appears in the invoke URL, and **stage variables** point one API's stages at different backends — for example, different **Lambda aliases** for dev and prod — with no code change. Caching, throttling, and logging are configured per stage.
- 04 Integrations translate both ways:** a client **method request** becomes an **integration request** for the backend, and the backend's **integration response** is mapped to the **method response**. **Lambda proxy** passes the response through unwrapped (your function must return the required output format); **non-proxy (custom)** lets API Gateway wrap it (adds 200 on success) and transform it with **VTL mapping templates**. Other types: HTTP proxy, AWS-service (first-class), mock, and private (VPC link).
- 05 Secure and authorize:** protect traffic with **resource policies (REST only)**, **client certificates / mutual TLS**, **AWS WAF**, **CORS**, and **throttling** — applied *most-granular-first* down to the **10,000 requests-per-second** Region quota (429 means throttled). Authorize callers with **None (open)**, **IAM**, **JWT (Amazon Cognito or third-party)**, or a **Lambda authorizer**.

02 KEY TERMS

TERM	DEFINITION
REST	Representational State Transfer — an architectural style; the standard way of structuring requests from a client to a server
RESTful API	An API that adheres to REST principles (request/response, stateless); non-RESTful APIs that use HTTP are also valid
Amazon API Gateway	Fully managed, serverless service to create, publish, maintain, monitor, and secure REST, HTTP, and WebSocket APIs — the front door to your backends
Method / route	The HTTP verb plus resource path that maps a request to a backend integration, such as GET / products
Stage	A named snapshot of a deployed API, identified by API ID and stage name, and included in the URL used to invoke the API
Stage variable	A name-value pair set per stage; lets one API point different stages at different backends (such as Lambda aliases) without code changes
Integration request / response	The translated request API Gateway forwards to the backend, and the backend response it maps back to the client's method response
Mapping template	An Apache Velocity Template Language (VTL) template used in non-proxy integrations to add, edit, or delete request or response data

TERM	DEFINITION
{proxy+}	Greedy path variable that lets a single integration absorb any nested path under a resource
Usage plan	Associates API keys with throttling and quota limits to control who calls stages and methods, and at what rate
Authorizer	The mechanism that authenticates and authorizes a request: None, IAM, JWT (Cognito or third-party), or Lambda
Canary release	A deployment that splits a preconfigured percentage of a stage's traffic to a new API version for testing before promotion

03 REST API VS. HTTP API

the exam's core service-choice decision

DIMENSION	REST API	HTTP API
Control & features	Full control; advanced, REST-only features	Proxy functionality only; features added over time
Cost & latency	Standard pricing and latency	Lowest cost (up to 70% savings) and latency
Endpoint types	Edge-optimized, Regional, private	Regional only
Mock endpoints	Yes	No
Resource policies	Yes	Not supported this way
API keys & usage plans	Yes	Not this way
Client SDK generation	Yes (REST only)	No
Choose when	You need advanced features or monetization	Everything else — the default

04 METHODS, ACTIONS, AND ROUTES

different paths → different backends

METHOD	ACTION	EXAMPLE ROUTE → TARGET
GET	Read (all)	GET /products → readFn
GET	Read (one)	GET /products/{product_id} → readFn
PUT	Update / replace	PUT /products/{product_id} → updateFn
POST	Create	POST /products → create an item (lab)
{proxy+}	Catch-all	GET /products/{proxy+} absorbs any nested path → readFn

05 HOW A REQUEST FLOWS THROUGH API GATEWAY

method side ↔ integration side

01 Client sends a **method request** (e.g., GET /products) → **02** API Gateway checks the **authorizer, throttling, and CORS** → **03** It translates it into the **integration request** the backend accepts → **04** Backend runs — Lambda, an AWS service, an HTTP endpoint, or a mock → **05** Backend returns the **integration response** → **06** API Gateway maps it to the **method response** and replies to the client

06 BACKEND INTEGRATION TYPES

know which one a scenario calls for

TYPE	CONNECTS TO	REMEMBER
Lambda proxy	A Lambda function	Response passes through unwrapped; function returns the required output format
Lambda non-proxy (custom)	A Lambda function	API Gateway wraps (200 on success) and transforms with VTL
HTTP proxy	An internet HTTP endpoint	EC2/Beanstalk or a public URL; good for monolith migration

TYPE	CONNECTS TO	REMEMBER
AWS service (first-class)	An AWS service API	e.g., SQS SendMessage, Step Functions; you supply the mapping
Mock	Nothing — returns default data	REST only; prototype front ends; handles the CORS OPTIONS call
Private (VPC link)	VPC resources via ALB/NLB	REST and HTTP; also AWS Cloud Map resources

07 EXAM TRAPS

where the knowledge check gets you

- ✗ “REST API and HTTP API are the same thing.” — No. The **REST API** type is advanced mode (full request/response control, mock endpoints, edge-optimized/private endpoints, resource policies, API keys/usage plans, client SDKs); the **HTTP API** type is a lightweight proxy, up to **70% cheaper** and lower latency. Default to HTTP unless you need a REST-only feature or must monetize.
- ✗ “Every API that uses HTTP is RESTful.” — REST is only an *architectural style*. Non-RESTful APIs that use HTTP are valid and common; the module says “RESTful” mainly to contrast request/response APIs with **WebSockets**.
- ✗ “Client apps can call Lambda or DynamoDB directly over HTTP.” — Lambda and many AWS services are **not publicly reachable via HTTP URIs**. Use API Gateway (or an SDK) to proxy the request and transform the response.
- ✗ “Lambda proxy integration formats the response for you.” — Backwards. In proxy integration the response **passes through unwrapped** — your function must return the required JSON output format. **Non-proxy (custom)** is where API Gateway wraps the response and VTL mapping templates transform it.
- ✗ “A successful Lambda call always returns 200 to the client.” — In non-proxy integration API Gateway treats any successful function call as **200**; to send an error (e.g., 403) you must map it to a standard HTTP error, or embed statusCode in the proxy output format.
- ✗ “Resource policies and edge-optimized endpoints work for any API.” — Both are **REST-only**. HTTP and WebSocket APIs support **Regional** endpoints; only REST APIs offer **edge-optimized** (CloudFront/DDoS) and **private** (VPC-only) endpoints, and HTTP APIs cannot use resource policies this way.
- ✗ “The account quota is checked first.” — Opposite. Throttling is applied **most-granular-first**: method-per-client → client (usage plan) → method/stage → stage → then the **10,000 req/s** account quota last. A 429 means the request was throttled.
- ✗ “Run dev and prod as two separate APIs (or a canary) for least config.” — With one Lambda function and two aliases, the **least-config** answer is **one REST API, two stages (dev, prod), and a stage variable** per stage holding the matching alias.

08 SELF-QUIZ

answers are upside-down below

- Q1** What does REST stand for, and is it a protocol or an architectural style?

Q2 Which API Gateway API type gives full control and advanced features, and which is the lightweight, lower-cost proxy?

Q3 What must an API deployment be attached to before REST or WebSocket clients can call it?

Q4 Which feature lets one API point its dev and prod stages at different Lambda aliases without changing code?

Q5 In a Lambda proxy integration, whose job is it to produce the HTTP response the client understands?
- Q6** Name the three REST API endpoint types.

Q7 Which greedy path variable lets a single integration absorb any nested path under a resource?

Q8 List the four authorizer options for API Gateway.

Q9 What HTTP status code indicates a throttled request, and what is the Region-wide account request quota?

Q10 Sample exam: to run dev and prod off one Lambda function with two aliases using the LEAST configuration — separate APIs, one API with two stages plus a stage variable, or a canary release?

ANSWER KEY (rotate the page)