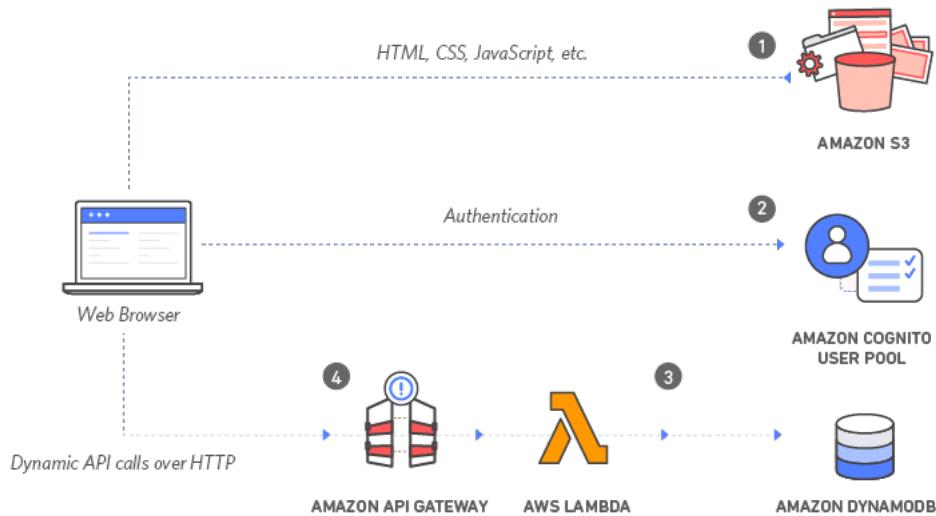


API Gateway - response error codes

A gateway response is identified by a response type defined by API Gateway. The response consists of an HTTP status code, a set of additional headers that are specified by parameter mappings, and a payload that is generated by a non-VTL (Apache Velocity Template Language) mapping template.



You can set up a gateway response for a supported response type at the API level. Whenever API Gateway returns a response of the type, the header mappings and payload mapping templates defined in the gateway response are applied to return the mapped results to the API caller.

The following are the Gateway response types which are associated with the HTTP 504 error in API Gateway:

INTEGRATION_FAILURE - The gateway response for an integration failed error. If the response type is unspecified, this response defaults to the DEFAULT_5XX type.

INTEGRATION_TIMEOUT - The gateway response for an integration timed out error. If the response type is unspecified, this response defaults to the DEFAULT_5XX type.

For the integration timeout, the range is from 50 milliseconds to 29 seconds for all integration types, including Lambda, Lambda proxy, HTTP, HTTP proxy, and AWS integrations.

HTTP 504

In this scenario, there is an intermittent issue where the users are getting HTTP 504 errors in the online auction. This means the Lambda function is working fine at times but there are instances when it throws an error.

Based on this analysis, the most likely cause of the issue is the **INTEGRATION_TIMEOUT** error since you will only get an **INTEGRATION_FAILURE** error if your AWS Lambda integration does not work at all in the first place.

Hence, the correct answer is *the underlying Lambda function has been running for more than 29 seconds which causes the API Gateway request to timeout.*

HTTP 502 or 429

Because a large number of incoming requests will most likely produce an HTTP 502 or 429 error but not a 504 error.

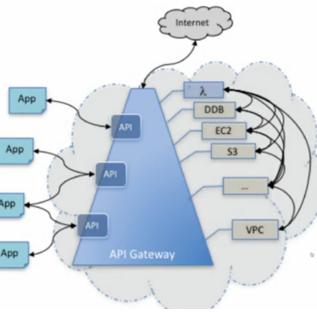
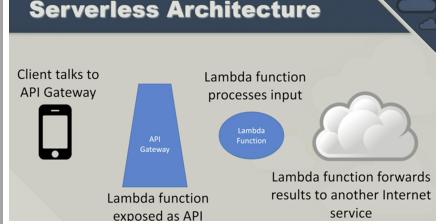
If executing the function would cause you to exceed a concurrency limit at either the account level (*ConcurrentInvocationLimitExceeded*) or function level

(*ReservedFunctionConcurrentInvocationLimitExceeded*), Lambda may return

a *TooManyRequestsException* as a response.

For functions with a long timeout, your client might be disconnected during synchronous invocation while it waits for a response and returns an HTTP 504 error.

The API Gateway allows applications to connect to code in the cloud. The code can be Lambda functions or other code, but the API Gateway provides a sort of wrapper around the code for external access.

API Gateway	API Gateway	Serverless Architecture
<ul style="list-style-type: none">• API management in the cloud<ul style="list-style-type: none">- Create- Publish- Maintain- Monitor- Secure	<ul style="list-style-type: none">• APIs can interact with many targets<ul style="list-style-type: none">- AWS services- Other web services- Data stored in AWS	<ul style="list-style-type: none">• Moves data in and out of the cloud without instances• Process functions without instances• Two primary services<ul style="list-style-type: none">- Lambda- API Gateway
	<p>Cross Origin Resource Sharing (CORS)</p> <ul style="list-style-type: none">• Can be enabled for the API gateway• Allows receipt of requests from other domains<ul style="list-style-type: none">- Default is internal domain requests only	

- The API Gateway provides Application Programming Interface (API) management in the cloud
- You can create, publish, maintain, monitor, and secure APIs with the API Gateway
- API Gateway APIs interact with AWS services, external web services, and data stored in AWS