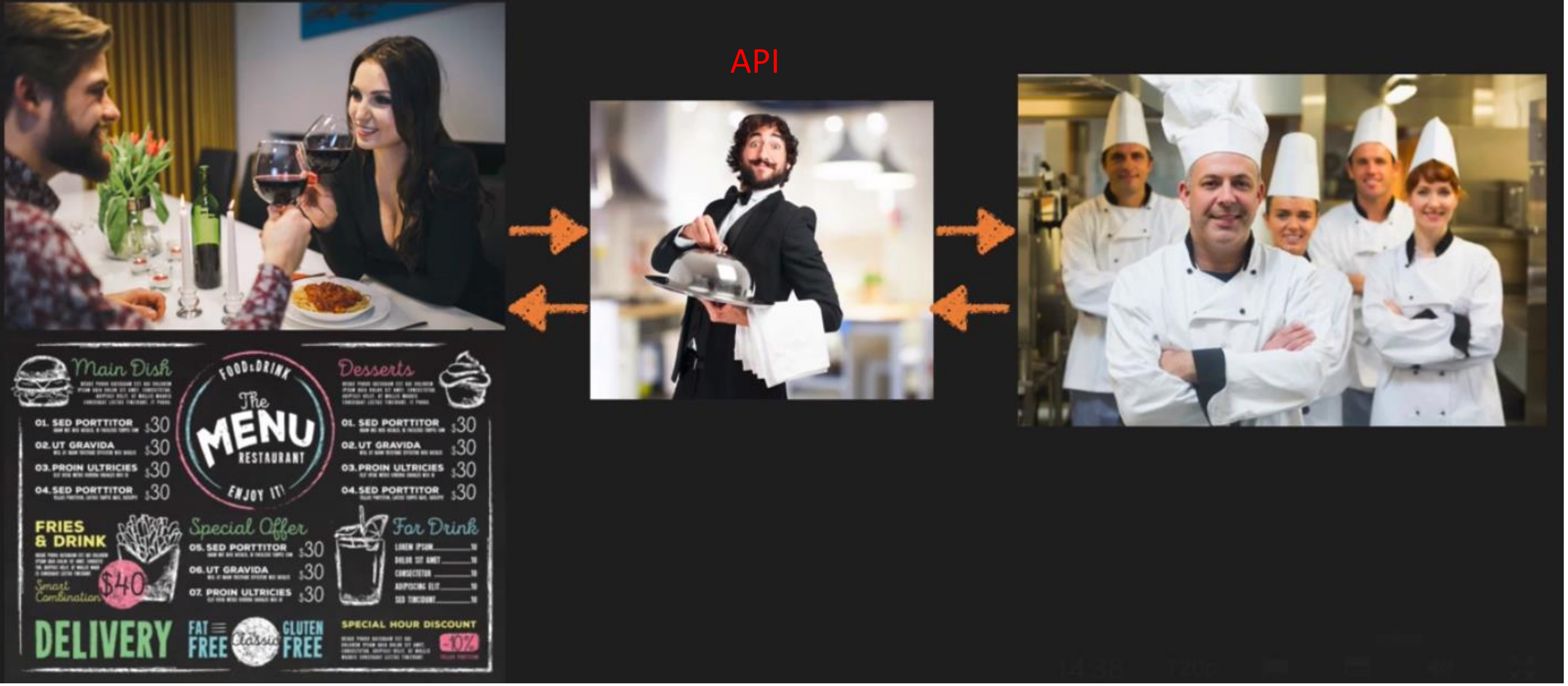
API Gateway 101

What is an API

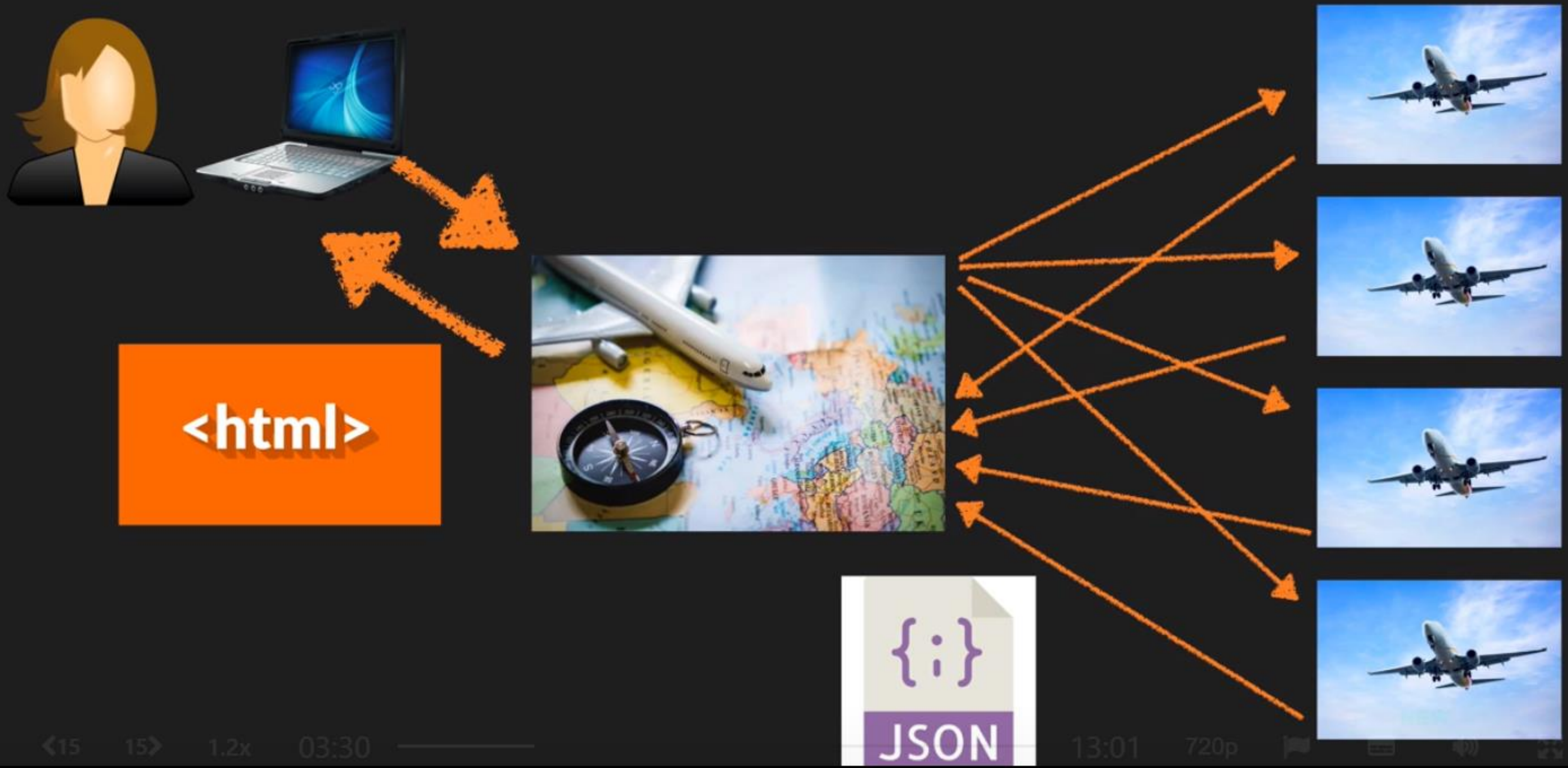
A CLOUD GURU

An **API** is an Application Programming Interface.



Travel Website Example

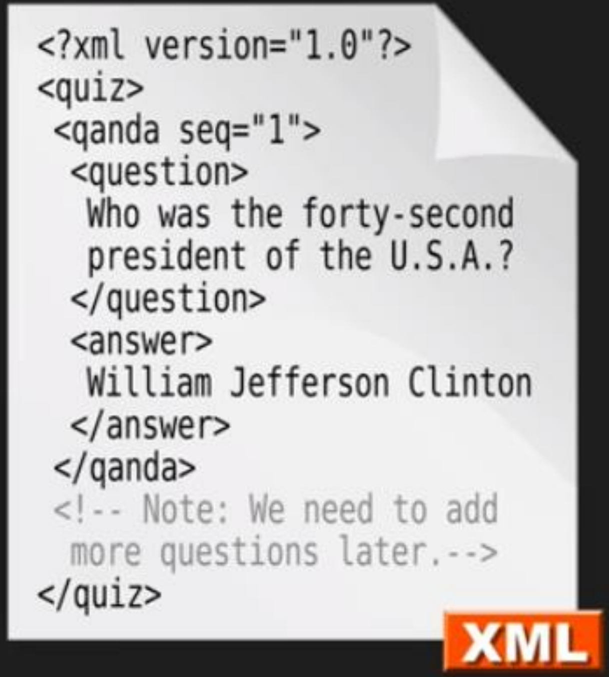
A CLOUD GURU



Types of APIs

- REST APIs (**RE**presentational **S**tate **T**ransfer)
  - Uses JSON
- SOAP APIs (**S**imple **O**bject **A**ccess **P**rotocol)
  - Uses XML

```
{
“_id” : “51262c865ca358946be09d77",
“firstname” : “John”,
“surname” : “Smith”,
}
```



```
<?xml version="1.0"?>
<quiz>
 <qanda seq="1">
  <question>
   Who was the forty-second
   president of the U.S.A.?
  </question>
  <answer>
   William Jefferson Clinton
  </answer>
 </qanda>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

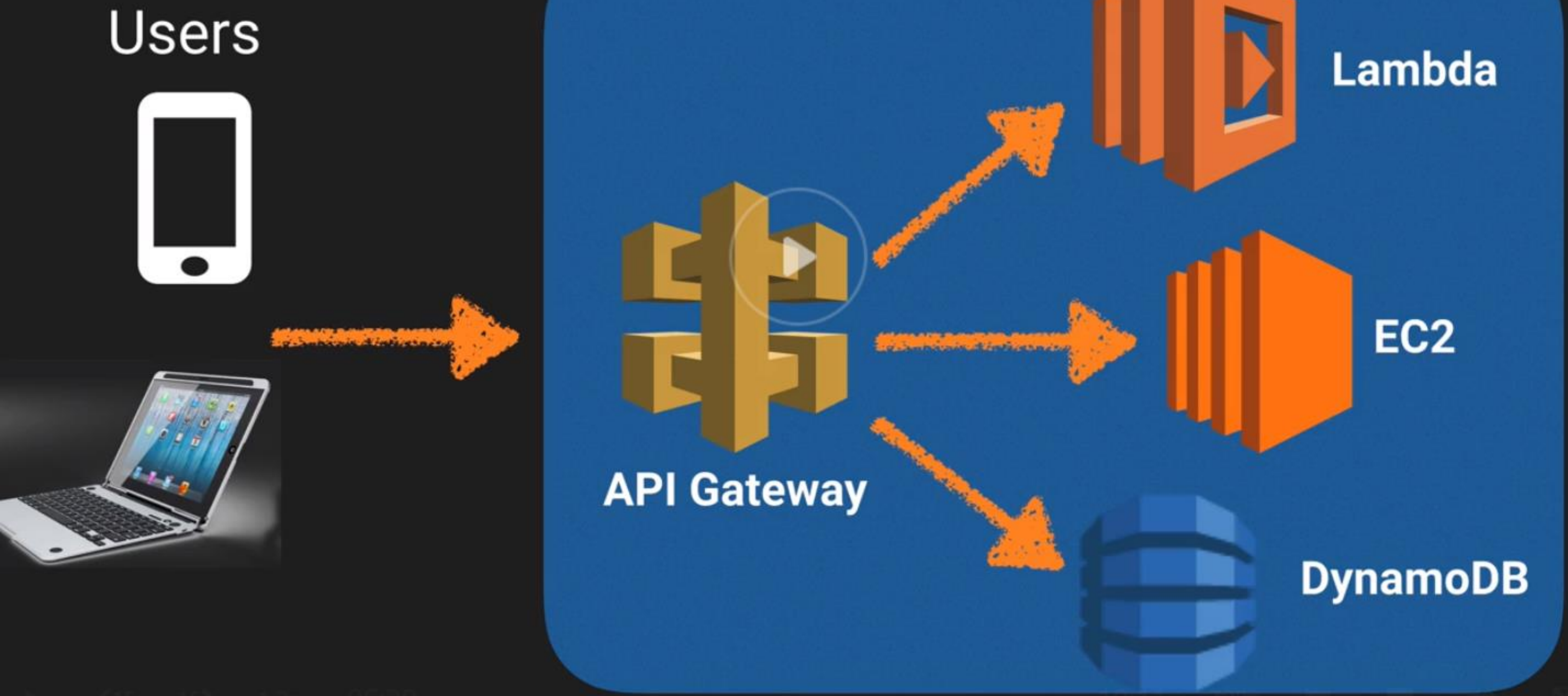
What is API Gateway?

A CLOUD GURU

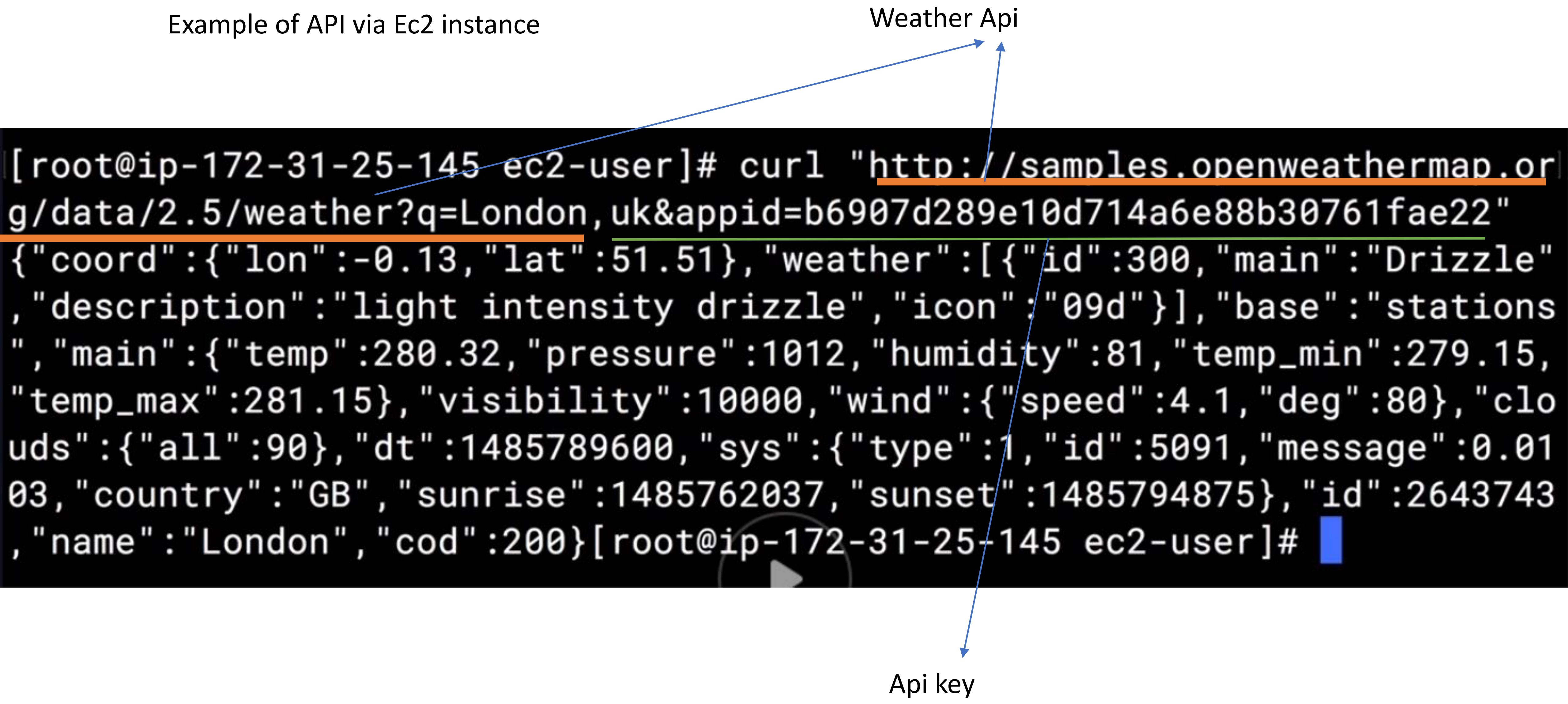
Amazon API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a "front door" for applications to access data, business logic, or functionality from your back-end services, such as applications running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application.

API Gateway

A CLOUD GURU



Example of API via Ec2 instance



```
[root@ip-172-31-25-145 ec2-user]# curl "http://samples.openweathermap.or
g/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22"
{"coord":{"lon":-0.13,"lat":51.51},"weather":[{"id":300,"main":"Drizzle"
,"description":"light intensity drizzle","icon":"09d"}],"base":"stations
","main":{"temp":280.32,"pressure":1012,"humidity":81,"temp_min":279.15,
"temp_max":281.15},"visibility":10000,"wind":{"speed":4.1,"deg":80},"clo
uds":{"all":90},"dt":1485789600,"sys":{"type":1,"id":5091,"message":0.01
03,"country":"GB","sunrise":1485762037,"sunset":1485794875},"id":2643743
,"name":"London","cod":200}[root@ip-172-31-25-145 ec2-user]#
```

What can API Gateway Do?

A CLOUD

- Expose HTTPS endpoints to define a RESTful API
- Serverless-ly connect to services like Lambda & DynamoDB
- Send each API endpoint to a different target
- Run efficiently with low cost
- Scale effortlessly
- Track and control usage by API key
- Throttle requests to prevent attacks
- Connect to CloudWatch to log all requests for monitoring
- Maintain multiple versions of your API

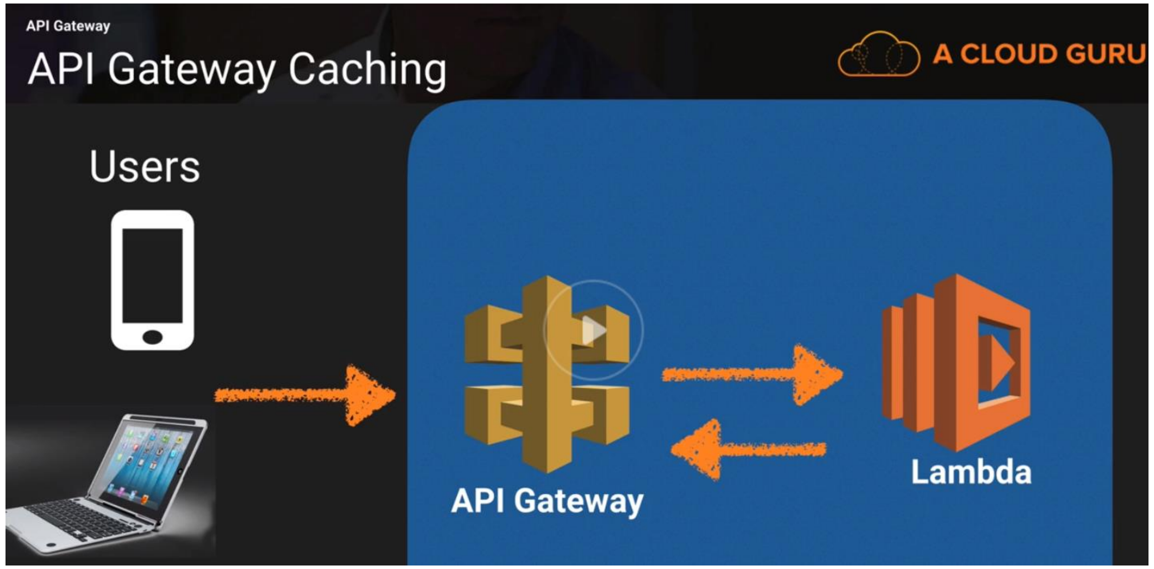
How do I configure API Gateway?

A CLOUD GURU

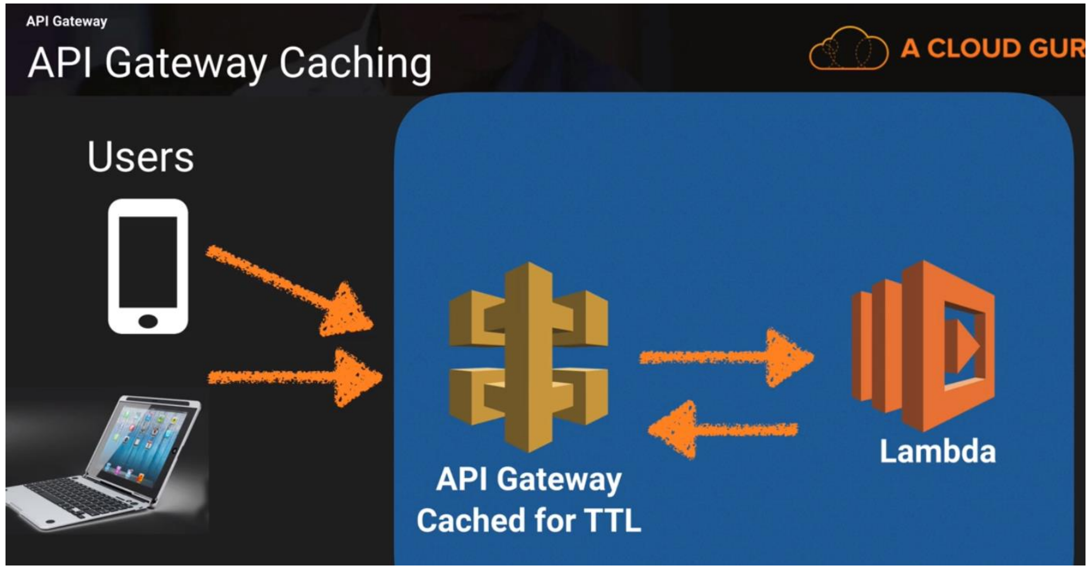
- Define an API (container)
- Define Resources and nested Resources (URL paths)
- For each Resource:
  - Select supported HTTP methods (verbs)
  - Set security
  - Choose target (such as EC2, Lambda, DynamoDB, etc.)
  - Set request and response transformations
- Deploy API to a Stage
  - Uses API Gateway domain, by default
  - Can use custom domain
  - Now supports AWS Certificate Manager: free SSL/TLS certs!

What is API Caching?

You can enable **API caching** in Amazon API Gateway to cache your endpoint's response. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of the requests to your API. When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.



When a user makes a request to api gateway and if the response is not cached, api gateway will hit lambda and the response is cached at api gateway and information is sent back to user.



When a new user makes the same request to api gateway and if the response is cached, api gateway will not hit lambda and the response is returned from cached at api gateway and information is sent back to user.

Same Origin Policy

In computing, the **same-origin policy** is an important concept in the web application security model. Under the policy, a web browser permits scripts contained in a first web page to access data in a second web page, but only if both web pages have the same origin.

This is done to prevent Cross-Site Scripting (XSS) attacks.

- Enforced by web browsers.
- Ignored by tools like PostMan and curl.

Cross-Origin Resource Sharing (CORS)

Cross-Origin Resource Sharing (CORS) is one way the server at the other end (not the client code in the browser) can relax the same-origin policy.

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts) on a web page to be requested from another domain outside the domain from which the first resource was served.

Cross-Origin Resource Sharing (CORS)

A CLOUD GURU

- Browser makes an HTTP OPTIONS call for a URL
  - OPTIONS is an HTTP method like GET, PUT, and POST

- Server returns a response that says:
  "These other domains are approved to GET this URL."

- Error - "Origin policy cannot be read at the remote resource?"
  You need to enable CORS on API Gateway.

Exam Tips

- Remember what API Gateway is at a high level
- API Gateway has caching capabilities to increase performance
- API Gateway is low cost and scales automatically
- You can throttle API Gateway to prevent attacks
- You can log results to CloudWatch
- If you are using Javascript/AJAX that uses multiple domains with API Gateway, ensure that you have enabled CORS on API Gateway
- CORS is enforced by the client