

AWS ROUTE 53



A HIGHLY AVAILABLE AND SCALABLE CLOUD DNS WEB SERVICE

DNS FUNDAMENTALS 🌐

Naming is fundamental in any distributed ecosystem as it helps to identify entities. As the internet is by far the largest distributed system, this concept is crucial.

💡 In a nutshell: with the internet's DNS, we get **human-friendly hostnames** for **computer-friendly IP addresses**.

A common example to visualize this is to think of it as a phone book for the internet. If you want to call somebody, you need some kind of directory that helps you find their number.



MANAGEMENT 🏛️

What's special about DNS: management is under **decentralized** control - therefore it's scalable and there's no single point of failure.

Furthermore: it has a hierarchical structure that is maintained by registries in different countries.

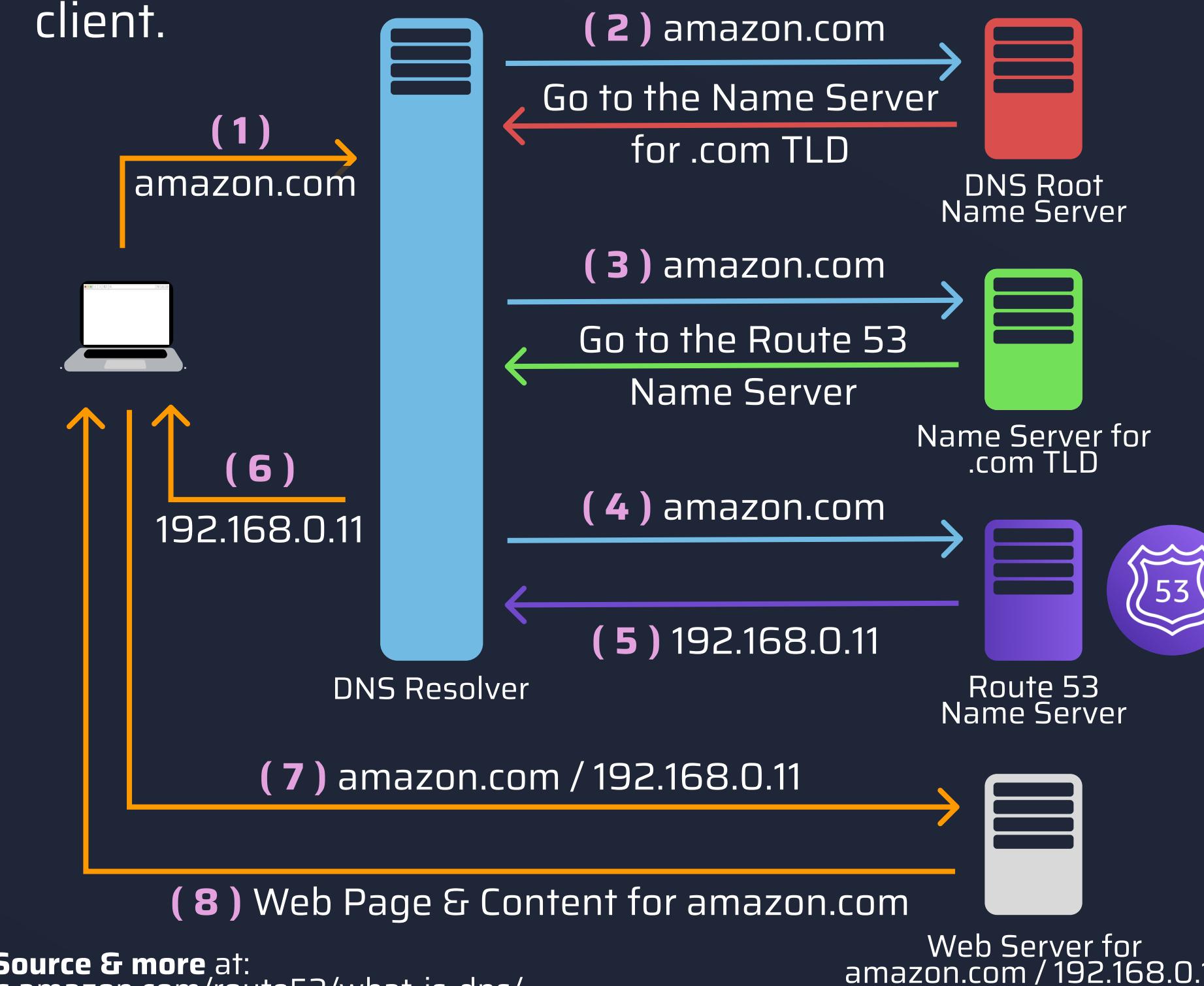
On top of the hierarchy are the **Authoritative Nameservers** which do hold the actual addresses for the DNS records.

When we go down we also find **Top Level Domain Servers**, **Root Name Servers & Recursors**.

RESOLUTION ⚙️

When resolving domain names, we need to distinguish between two types of services:

- **Authoritative DNS Service**: actually respond to DNS queries and do have the final authority over a domain. Those services actually provide IP address information to clients & recursive services.
- **Resolver or Recursive DNS Service**: does not own DNS records, but connects to other DNS services as an intermediary to resolve any given record. If a record is already cached, it can be immediately provided to a client.



CACHING 💡

Caching is essential to keep the internet's DNS resolvers healthy.

There are **different locations** where caching takes place. The two most obvious of those caching locations are your **browser** & your **operating system**. The closer the caching to your browser, the better, as fewer processing steps are necessary.

How long a DNS record is stored is specified by its **Time-To-Live (TTL)** attribute.

PRICING 💰

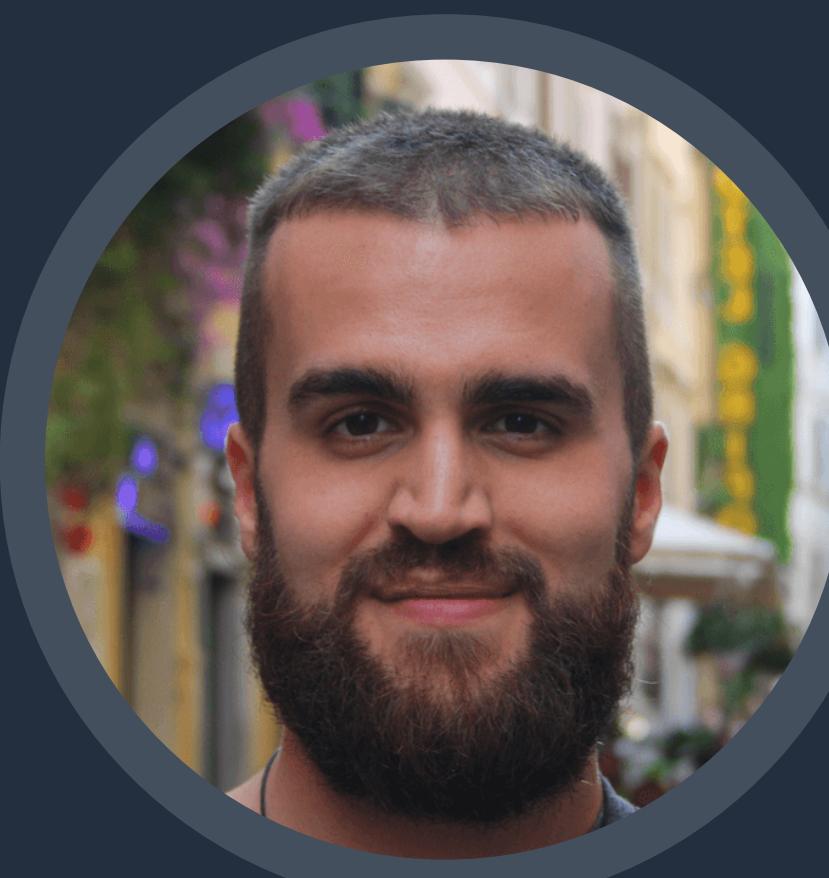
Route 53 has very **transparent pricing**:

- **pay-as-you-go** model
- no minimum usage commitments or upfront costs

What you pay for:

- \$0.5 per Managed Zone
- \$0.4 per million standard queries

Additionally, you'll pay for **health checks**, starting from \$0.50 for a default HTTP health check for AWS-hosted endpoints.



[in/in/tpschmidt](#)
[@tpschmidt_](#)

INTRODUCTION 🙋

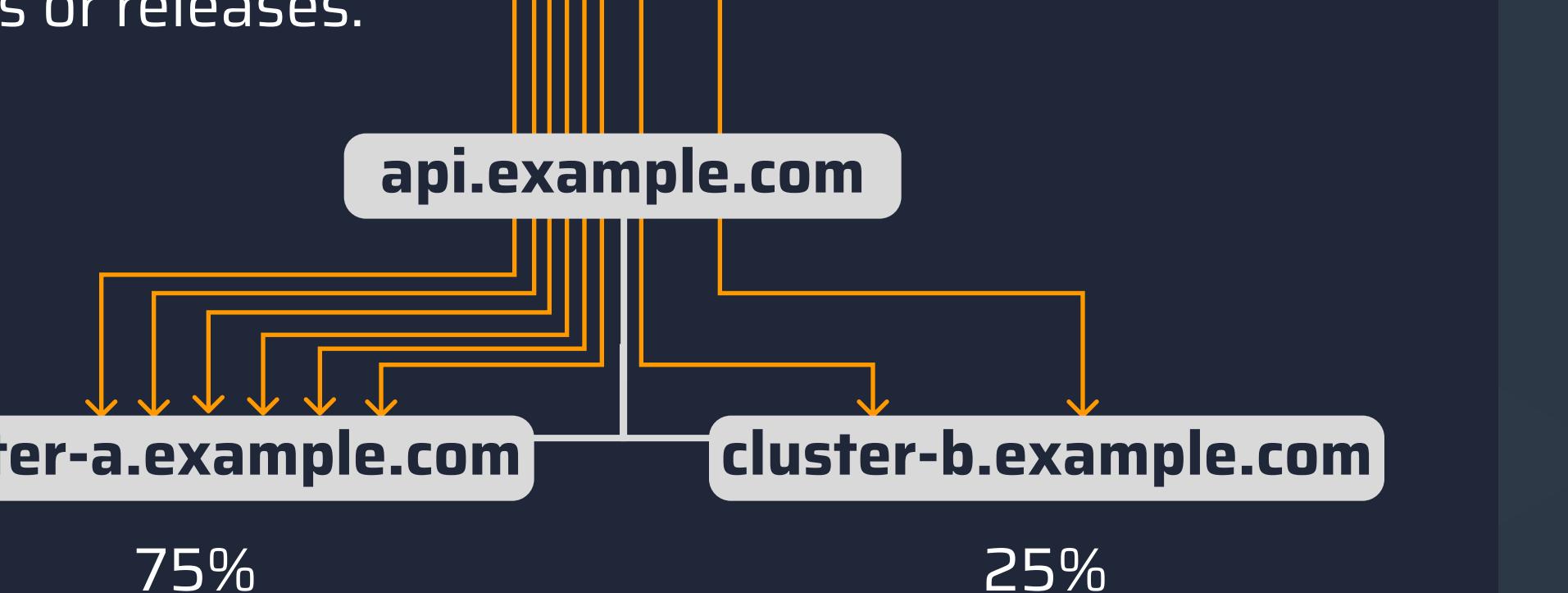
Introduced by Amazon in 2010, Route 53 is a managed service to reliably redirect traffic via domain names to your applications.

It doesn't matter if it's an S3 bucket, an Load Balancer, an API Gateway with Lambda functions, or even an on-premise server behind it.

WEIGHTED ROUTING 📈

As the name already suspects, it allows you to define multiple records for the same domain name. You choose how much traffic is routed to each of the records by giving it a percentage.

Prominent use cases are load balancing and testing new features or releases.



Weighted routing not only enables you to easily scale your application, but also build blue/green deployments or do traffic shifting that is fully in your control.

GEO-LOCATION ROUTING 🌎

Geo-Location records allow routing based on the origin of your clients. This enables you to easily localize content or implement geo-restrictions to comply with regulations.

The granularity of locations is either by

- continent
- country
- or even US state.

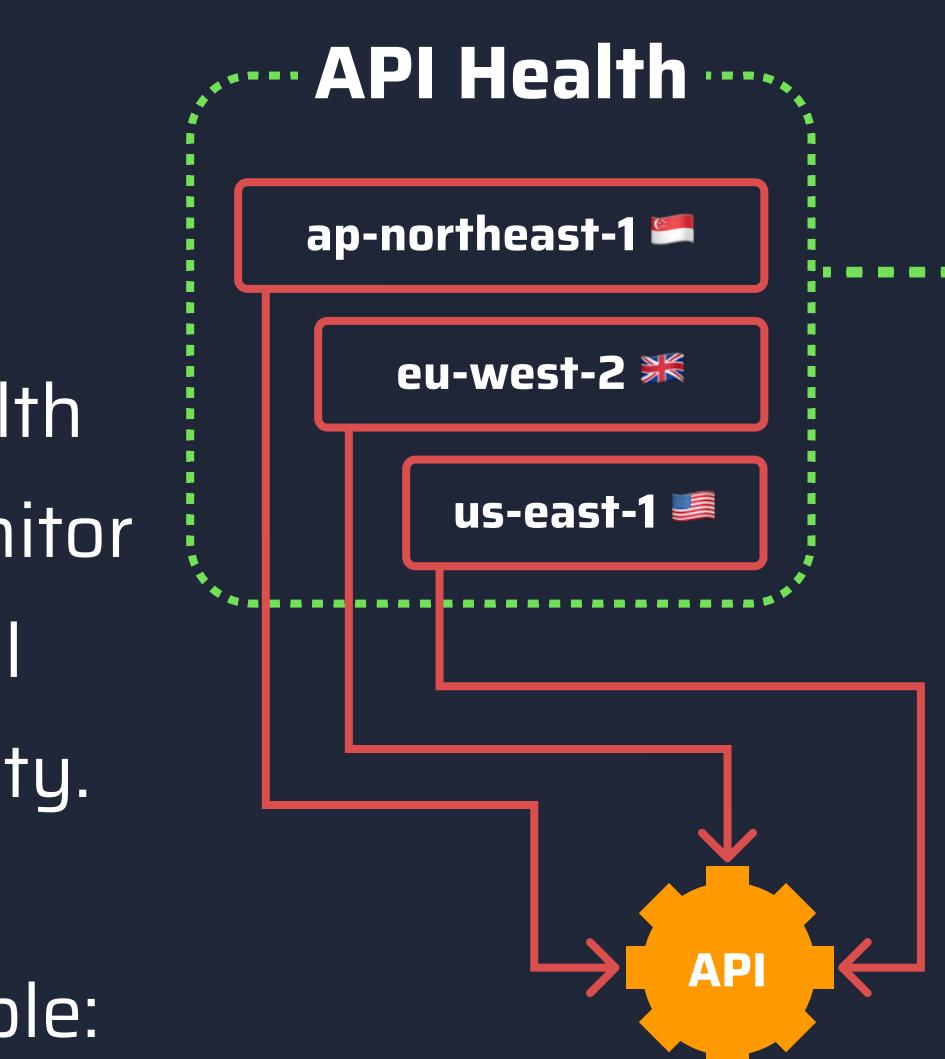
ROUTING POLICIES ➔

There are different types of routing that you can use for your domain names.

The type determines how Amazon Route 53 responds to queries for those domain names. The selection of the best fitting record types highly depends on your requirements.

HEALTH CHECKS & FAILOVERS ❤️

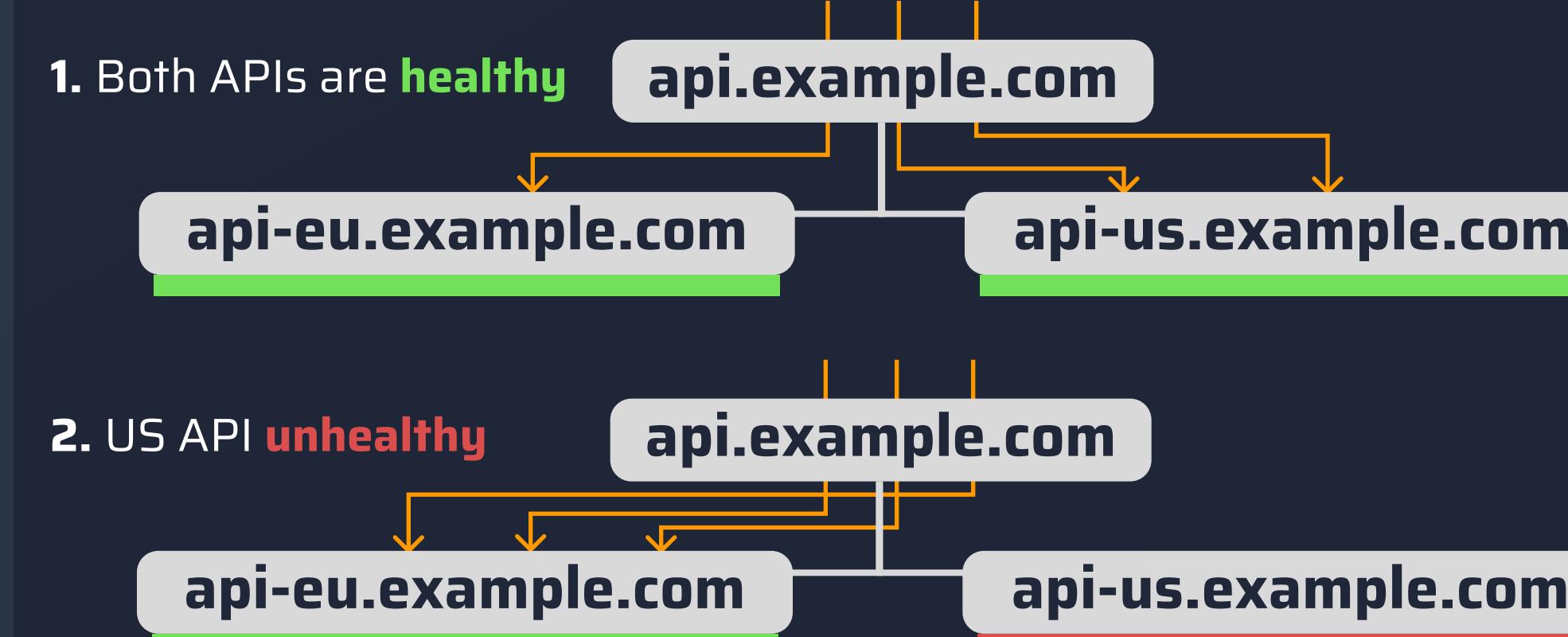
What if you have a multi-region setup with latency-based routing, but the closest region for a customer is not available for any reason? Surely, we don't want to route requests to this region.



Those **checks** are configurable:

- the time between two independent checks
- from which locations they should origin

The exciting part: you can attach those health checks for example to your Latency-based records. If a health check for one of your locations becomes unhealthy, the corresponding target won't be propagated anymore for this DNS record.



There you have it: a quite easy active-active multi-region failover setup.

SIMPLE ROUTING 📄

The standard DNS record without any strings attached. Typically used for single resources that are performing given functions for your domain.

You can't create multiple records with the same name for this record type. What you can do: specify multiple values for your single record.

Route53 will return all values for an incoming query to the client. The client can then choose one of those by himself.

LATENCY-BASED ROUTING 📈

In a multi-region setup, in most cases you want to route requests to the closest regions, as it will on average serve the fastest responses.

Via Latency-based routing, you create multiple records for a given domain name. Each record is created for a specific region and if your DNS record is queried, Route 53 will resolve it by choosing the one with the lowest latency.

💡 Latencies between hosts can change. If a client is close to several regions, routing results can vary over time.

SERVERLESS MULTI-REGION ⚡

What I found quite exciting:

If you're focusing on serverless architectures - with **pay-as-you-go pricing** - you can deploy your eco-system around the world without drastically increasing your costs.

Route 53 with Latency-based records and health checks will take care of routing requests to the fastest region.

If there's few or no traffic at all in a single region, it won't hurt in any way. If a region breaks - or you break a region by accident via a faulty deployment - there won't be any significant outage as Route 53 will quickly failover and won't return the unresponsive region in DNS queries.

