The Internet is down.... It was DNS, again

4 min read · Oct 20, 2025

adrian cockcroft

Photo by Adrian — Peacock skylight at Burgh Island, Devon

One of the common ways web services fail is problems with DNS. I've talked about how to harden systems against DNS related outages before but I don't have my thoughts written down in one place. Today there was a large AWS outage related to DNS, so it seems timely to get this done.

> *the main errors we are seeing are Could not resolve host: dynamodb.us-east-1.amazona*

The Domain Name System (DNS) is a simple lookup that translates the written name of a web service to it's IP address. Something like aws.amazon.com to a routable internet address.

```
% nslookup aws.amazon.com
Server:  192.168.178.1
Address: 192.168.178.1#53

Non-authoritative answer:
aws.amazon.com canonical name = tp.8e49140c2-frontier.amazon.com.
tp.8e49140c2-frontier.amazon.com canonical name = dr49lng3n1n2s.cloudfront.net.
Name: dr49lng3n1n2s.cloudfront.net
Address: 3.166.65.11
Name: dr49lng3n1n2s.cloudfront.net
Address: 3.166.65.58
Name: dr49lng3n1n2s.cloudfront.net
Address: 3.166.65.78
Name: dr49lng3n1n2s.cloudfront.net
Address: 3.166.65.124
```

In this case looking up the address first went to my local Wifi router on port 53 and that looked it up for me via a DNS server and translated it into four of the AWS cloudfront service addresses. Four to provide redundancy in case one of them has an issue.

However when DNS breaks, there is no way to lookup the address, so the requests fail, and the service itself sees no traffic arriving. From the point of view of the people operating the service, it's still up and running.

In this most recent failure it was the DNS endpoint for DynamoDB in the AWS us-east-1 region that wasn't working. From the current status page:

> ***Oct 20 2:01 AM PDT*** *We have identified a potential root cause for error rates for the DynamoDB APIs in the US-EAST-1 Region. Based on our investigation, the issue appears to be related to DNS resolution of the DynamoDB API endpoint in US-EAST-1. We are working on multiple parallel paths to accelerate recovery. This issue also affects other AWS*

> *Services in the US-EAST-1 Region. Global services or features that rely on US-EAST-1 endpoints s* *riencing issues. Dur* *Cases.*

The problem with this particular failure is that lots of internal AWS services use DynamoDB to store their information, so many additional services broke, as well as several customers. Amazon retail, Amazon Alexa, Lyft, Snapchat, Signal etc. Even customers that are not hosted in us-east-1 would be affected indirectly as they could not create support cases, or change configurations of Identity and Access Management (IAM).

. . .

There are several common causes of DNS outages, the most common is a configuration change that breaks the endpoint. It could occur during manual configuration, or as part of an automated update that didn't work properly or didn't propagate to all the places it was supposed to go. Although in today's case it appears to just be a single endpoint, the entire DNS service itself can go down, there have been cases where DNS providers have failed. There are many companies that provide DNS services, and its usually where the domain is registered. The other cause of DNS outages is failure to maintain and renew the registration. There was a case where a SaaS company forgot to renew, and their entire service went offline for a few days while they tried to recover, with internal email and everything else down, the CEO was tweeting apologies to their customers.

. . .

So what can we do about it to survive DNS failures? They occur relatively infrequently so it's not common to take the measures I'm going to describe, but I think it's certainly possible to survive many DNS outages with careful planning.

1. Use more distinct domains. Don't run your entire company on a single domain, for your email, your internal company services and your externally facing product. Easy if you think about it before you deploy your first customer facing product...

2. Support more than one DNS provider in your operations automation, and test your ah em synchr *pany.net* to work interchangeably and register them at different providers, with different account expiration dates.

3. Support multiple domains in the code that looks up your services, via APIs or web pages, so that if the code fails to reach *company.com* it automatically tries *company.net* for example.

4. Don't use DNS if you don't really need to. The Netflix internal Ribbon client library would look up services and get the IP address from Eureka as well as the full URL. The IP address for a microservice will never change before its replaced, so it was called directly using the IP rather than the name, bypassing any possible DNS issues.

It's difficult to keep DNS configurations automated and in sync across multiple providers. Many years ago when I was worrying about this problem at Netflix, I worked with Adrian Cole and others to build an open source Java library called Denominator that could maintain a DNS configuration and which had pluggable API support for several of the major DNS services and a simulator for testing. It's fallen out of use nowadays, but something similar would be needed to automate DNS updates reliably. This still leaves the operational issues of changing configurations incorrectly, but by staging across more than one provider with a delay to verify the change, it could be a lot more reliable.

I hope people find these ideas useful as they figure out more resilient approaches to DNS.

AWS | Cloud Computing | DNS | Resilience | Netflix

Written by adrian cockcroft

13.1K followers · 464 following

Work: Techno

ery Ventures, Netflix, eBay, S

Responses (7)

Bgerby

What are your thoughts?

Tim Robinson
Oct 22

The problem wasn't caused by DNS - that's like saying the car broke down because it had an engine. The problem was caused because DNS failed, but they haven't told us whether this was hardware failure, misconfiguration, overloading, hackers or whatever.

1 Reply

Saint Fear
Oct 22

I love this article; it articulates the insufficiency of our current DNS system really well. Actually, my current project is building an alternative to DNS; here's the link to the whitepaper... more

1 Reply

Peter HJ van Eijk
Oct 25

https://dnshaiku.com

Reply

More from adrian cockcroft

adrian cockcroft

Directing AI Native Development

I've gone AI native, I have figured out how to get the tools to do what I want and produce good results with minimal input and minimal...

Oct 5 74 7

adrian cockcroft

Deep dive into NVIDIA Blackwell Benchmarks—where does the 4x training and 30x inference...

I've been trying to write this blog post since the announcements at GTC, and was planning to publish it as a story for The New Stack, but...

Sep 23, 2024 125

adrian cockcroft

Failure Modes and Continuous Resilience

A resilient sy… re are many possible fail…

Nov 11, 2019

adrian cockcroft

So many bad takes—What is there to learn from the Prime Video microservices to monolith story

The Prime Video team published this story: Scaling up the audio/video monitoring service and reducing costs by 90%, and the internet piled…

May 6, 2023 858 9

See all from adrian cockcroft

Recommended from Medium

In Towards AI by Teja Kusireddy

We Spent $47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic's Model Context Protocol (MCP) are revolutionary. But...

Oct 16 2.8K 96

Tosny

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.

Sep 23

In Level Up Coding by Jacob Bartlett

The Terrible Technical Architecture of my First Startup

A system that could only be built with the confidence of a 25-year-old Deloitte graduate with an AWS certification

6d ago 311 8

In Entrepreneurship Handbook by Joe Procopio

Amazon's Return To Office Mandates Backfire

The exodus they broke the internet

Oct 29 2.8K 62

Abhinav

Docker Is Dead—And It's About Time

Docker changed the game when it launched in 2013, making containers accessible and turning "Dockerize it" into a developer catchphrase.

Jun 8 7.2K 207

Noah Byte

Welcome back. You are signed into your member account
bg••••@jaxondigital.com.

I Rewrote

A dark comedy about choosing the “wrong” technology.

Oct 11 749 42

See more recommendations