

More AI won't fix bad data (More data won't fix bad AI)

Andy Burghes / Stefan Spiewok

IT Automation with AI – the starting concerns

The dream is full IT automation but is it a reality?



IT Automation with AI – the starting concerns

The dream is full IT automation but is it a reality?



Data Sovereignty

What happens to our data once a model sees it? We need to know whether it is used for training or shared beyond us.



IT Automation with AI – the starting concerns

The dream is full IT automation but is it a reality?



Data Sovereignty

What happens to our data once a model sees it? We need to know whether it is used for training or shared beyond us.



Cost

Token usage drives the bill.
Feeding huge datasets to a model gets expensive fast.



IT Automation with AI – the starting concerns

The dream is full IT automation but is it a reality?



Data Sovereignty

What happens to our data once a model sees it? We need to know whether it is used for training or shared beyond us.



Hallucinations

An automation platform that acts on a wrong answer makes the wrong change to the network.



Cost

Token usage drives the bill. Feeding huge datasets to a model gets expensive fast.

CIOs are looking at how AI will change their business



incorrect

fabrication

not true

made up

false

misinformation

guesswork

confidently wrong

fiction

error

invented facts

unverified

Hallucinations

plausible nonsense

fake source

wrong answer

inaccurate

bogus citation

mistake

confabulation

misleading



MCP – How to share Data with AI

MCP is the USB-C of AI, connecting the AI model with external systems



Standard

An open standard from Anthropic, now supported by every major model vendor. We build the connection once and it works with whichever model we choose, instead of a custom integration per vendor.



Tools

The model never touches the database directly. MCP exposes named tools with defined inputs and permissions, so we decide exactly what can be asked for, and every call can be logged and audited.



Reasoning

The model requests only the data a question needs, at the moment it needs it. Nothing is bulk-loaded into the context window, which keeps token cost down and keeps our data in our systems.

MCP is interface for AI to take Siloed data



BUT - The AI Data Problem

AI models are powerful reasoners. But give them bad data or too much data and they hallucinate.



Unstructured

Free-text logs require parsing, regex, and guesswork. Every vendor formats differently. AI spends tokens interpreting format, not finding answers.



Too Much Data

Logs and flow records run to billions of rows a day. Raw volume will not fit in a context window, and pushing it there is slow and expensive.



Hallucinations

Given messy or partial data, the model fills the gaps and computes its own statistics. The answer reads well and is wrong, and automation acts on it.

What if there was a dataset that was structured, complete, and right-sized for AI?



NETSCOUT has the AI-Ready Dataset

AI models need data that is pre-calculated, understood, and right-sized.



Pre-Calculated

Response Times | Error Rates

ASI computes the metrics at ingestion: response times, error rates, percentiles, throughput. The model receives answers, not raw data to process.



Understood

SYN/ACK Time | TCP Resets

The models already know our metrics. SYN/ACK time, TCP resets, retransmits: no fine-tuning, no training on our data, no glossary to maintain.



Right-Sized

Metrics, Not Records

Instead of billions of raw events, AI gets access to pre-aggregated metrics with full context: who, what, when, how long, and whether it worked.

Clean inputs, confident answers – the foundation for Automation you can trust.



How deep the dataset goes

Packet-derived metadata, curated for AI, drillable to a single transaction.



Packet-Derived

Deep Packet Inspection

Every metric comes from the packets on the wire, not from agents, logs, or device counters.



Curated Views

Metric and Dimension Selection

Define the aggregate views, so the model reads considered metrics rather than a raw dump.



Drill-Down

MCP Tool Capability

AI moves from a summary to the single transaction behind it, with the same meaning at every level.

One fact table per protocol, 100+ dimension tables for context.



IT Automation - when AI gets access to this data



Troubleshooting

"Why is the web store slow?" AI traces symptom to root cause across application, transport, and infrastructure. No dashboards, no queries.



Anomaly Explanation

Not just "anomaly detected" but "MySQL response time spiked to 1,168ms; 45 standard deviations above the 5.6ms mean; affecting all 3 middleware hosts." Context makes the finding actionable.



Impact Assessment

"If this database server fails, what breaks?" AI maps every dependent host, application, and user community before the change window opens.



Ticket Enrichment

Every ticket arrives with the relevant metrics attached and a probable root cause, so the engineer starts at diagnosis, not data gathering.



Change Validation

AI compares the metrics either side of a change window and reports whether the change worked, made no difference, or degraded a site.



Alert Triage

AI checks each alert against the dataset, discards the noise, and routes the rest to the right team with the evidence already attached.



What else can AI do with the data?



Custom Dashboards

Ask for the view you need and AI builds it: this application, these sites, this week. No dashboard backlog, no ticket to the tools team.



Executive Reporting

A monthly service summary in plain language: what broke, how long it lasted, who it affected, and what changed since last month.



Incident Summaries

As an incident closes, AI writes the timeline, the impact, and the probable root cause from the data, ready for review.



Issue Explanation

AI explains what a metric means and why it matters, so a level-1 engineer can act on it instead of escalating.



Capacity Trends

AI reads months of metrics and shows which links, sites, and servers run out of headroom first.



Knowledge on Demand

Ask how a service is built and AI answers from observed dependencies and traffic, not tribal knowledge.



What that looks like

One dataset, three outputs on request.

Security Operations Center Dashboard

Network Security Monitoring • Last 24 Hours

42.1M

1 Normal volume

1.5M

3.57% failure rate

540K

Elevated

Overview

Alerts

Protocols

DNS

Auth

Certs

Anomalies

TLS Version Distribution

TLS 1.3

2,934,635 (60.1%)

TLS 1.2

1,796,132 (36.7%)

TLS 1.1

107,291 (2.2%)

TLS 1.0

49,000 (1%)

3.2% of connections using deprecated TLS 1.0/1.1

Self-Signed Certificates Detected

192.168.132.1:443

CN: netscout-lb-01.local

First seen: 2025-03-15

DNS Service Degradation

December 9, 2025 • 11:15 - 12:15 UTC

HEALTH SCORE

71.1%

Degraded

FAILURE RATE

26.4%

106 / 402 requests

Failure Rate Over Time



DNS Server Status

9.9.9.9 (Quad9)

402 requests

1.1.1.1 (Cloudflare)

72 requests

8.8.8.8 (Google)

2 requests

8.8.4.4 (Google)

1 requests

Probable Root Cause

WAN Path Issue

Network routing problems to external DNS providers

ISP/Upstream

Packet loss or congestion at upstream provider

DNS Service Degradation Report

Report Date: December 9, 2025

Severity: HIGH

Affected Service: DNS Resolution (External)

Time Window: 11:15 - 12:15 UTC (ongoing)

Health Score: 71.1% (degraded)

Executive Summary

The DNS service is experiencing significant degradation due to connectivity issues with external DNS resolvers. The primary affected server is Quad9 (9.9.9.9), which is showing a 26.4% failure rate and 1,893ms average latency. All failures are complete timeouts (no response received), not server-side errors such as SERVFAIL or REFUSED.

This issue has been persistent throughout the past hour with no signs of self-recovery. The pattern suggests network path issues to external DNS providers rather than DNS server problems.

Impact Summary

DNS Server	Failure Rate	Avg Latency
9.9.9.9 (Quad9)	26.37%	1,893 ms
1.1.1.1 (Cloudflare)	0%	N/A
8.8.8.8 (Google)	0%	7,486 ms
8.8.4.4 (Google)	0%	17,557 ms

Detailed Findings

Error Analysis

All 106 DNS failures over the monitoring period were categorized as "Failed" — meaning requests that received no response (timeouts). Notably:

- Zero SERVFAIL errors — DNS servers are not returning server-side errors
- Zero NXDOMAIN errors — Queries are not being rejected as non-existent
- Zero REFUSED errors — Servers are not actively refusing queries

This pattern strongly indicates a network connectivity issue rather than a DNS server problem.

Timeline Analysis

The issue has been persistent throughout the entire monitoring window (11:15-12:15 UTC):

Time (UTC)	Requests	Errors	Failure %	Latency
11:20	24	0	25.0%	1,221 ms
11:25	32	0	18.8%	1,400 ms
11:30	32	10	31.3%	2,431 ms
11:35	28	0	21.4%	2,625 ms
11:40	32	0	18.8%	2,331 ms
11:45	38	14	36.8%	1,814 ms
11:50	50	20	40.0%	1,501 ms
11:55	28	0	21.4%	1,633 ms
12:00	38	10	26.3%	1,940 ms
12:05	30	0	20.0%	1,939 ms
12:10	34	0	17.7%	1,334 ms
12:15	38	10	26.3%	2,351 ms

Peak degradation occurred at 11:45-11:50 UTC with 40% failure rate.

Root Cause Analysis

Based on the evidence collected, the most likely causes are:

- WAN Path Degradation: Network routing issues between our infrastructure and external DNS providers. The fact that multiple external resolvers show elevated latency supports this hypothesis.
- ISP/Upstream Provider Issue: Potential packet loss or congestion at an upstream provider affecting outbound UDP/53 traffic.
- Firewall/Security Appliance: Possible rate limiting or connection tracking exhaustion affecting DNS traffic specifically.
- Quad9-Specific Routing: The significantly higher failure rate to 9.9.9.9 suggests possible BGP routing issues specific to Quad9's anycast network.

Recommended Actions

Immediate (Next 1-2 Hours)

- Run traceroute and MTR to 9.9.9.9, 8.8.8.8, and 1.1.1.1 to identify packet loss points
- Check firewall connection tables for UDP/53 state exhaustion
- Review any recent network changes (routing, firewall rules, NAT policies)
- Consider temporarily shifting DNS traffic away from Quad9 to Cloudflare (1.1.1.1) which shows 0% failures

Short-Term (Next 24-48 Hours)

- Contact ISP/upstream provider to report potential routing issues
- Implement DNS failover configuration with automatic health checking
- Review and potentially increase DNS timeout thresholds on clients

Long-Term Recommendations

- Deploy local DNS caching/forwarding infrastructure to reduce external dependency
- Implement multi-provider DNS with automatic failover (e.g., Cloudflare + Google)
- Configure proactive DNS monitoring with alerting thresholds

Report Generated By

NETSCOUT nGeniusONE Service Dashboard — Automated Network Intelligence

Data Source: ASI (Application Service Intelligence) real-time packet analysis

COPYRIGHT © 2026 NETSCOUT SYSTEMS, INC. | NETSCOUT PUBLIC

13

IT Automation with AI – Summary

IT Automation is coming but it will be a stepped process



Data Sovereignty

Frontier models promise not train on corporate data. Expect smaller models or agents to work with the data, with a larger model checking the outcome.



Hallucinations

Nothing is 100% correct today. With the right inputs, automation is safe to run, and to start with a human will stay in the loop.



Cost

Token charges keep falling, so the case for automation gets stronger. Staff move from firefighting to project work.

AI will change your IT Operations



ENTERPRISE SCENE #1

☀ Good morning, Andy

are there any network issues currently?



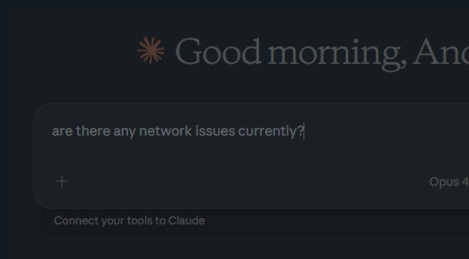
Opus 4.6 Extended ▾



Connect your tools to Claude



ENTERPRISE SCENE #1



Examined network health and anomalies comprehensively ▾

🕒 The user is asking about current network issues. Let me check the health overview and anomalies to get a comprehensive picture.

✅ Done

Let me check your network health right now.

Identified and synthesized network concerns for user clarification ▾

🌀 Get health overview

Result

🕒 DNS has a very high error rate at ~22%. Let me get more details on the anomalies and DNS errors.

🌀 Get anomalies

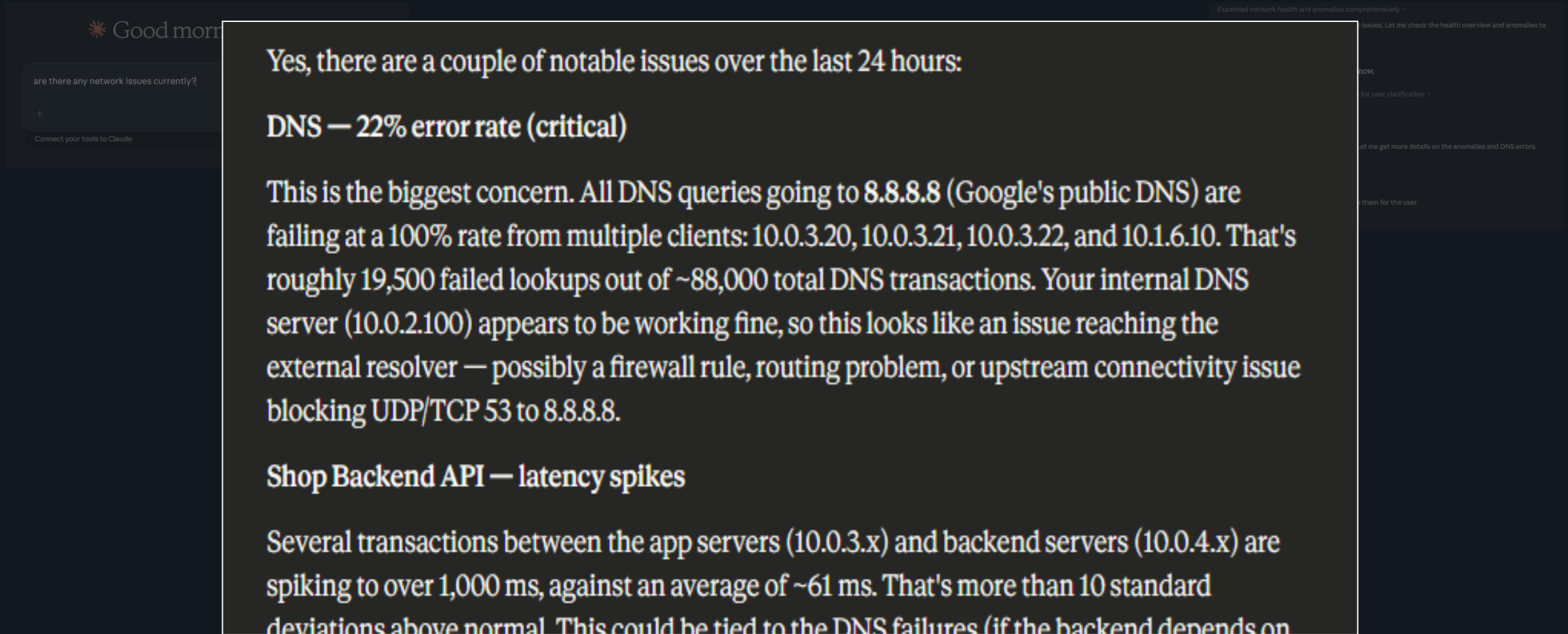
Result

🕒 There are clear issues. Let me summarize them for the user.

✅ Done



ENTERPRISE SCENE #1



Good morning

are there any network issues currently?

+

Connect your tools to Claude

Examined network health and anomalies comprehensively. ~

Issues. Let me check the health overview and anomalies to

now.

for user clarification. ~

at me get more details on the anomalies and DNS errors.

them for the user.

Yes, there are a couple of notable issues over the last 24 hours:

DNS — 22% error rate (critical)

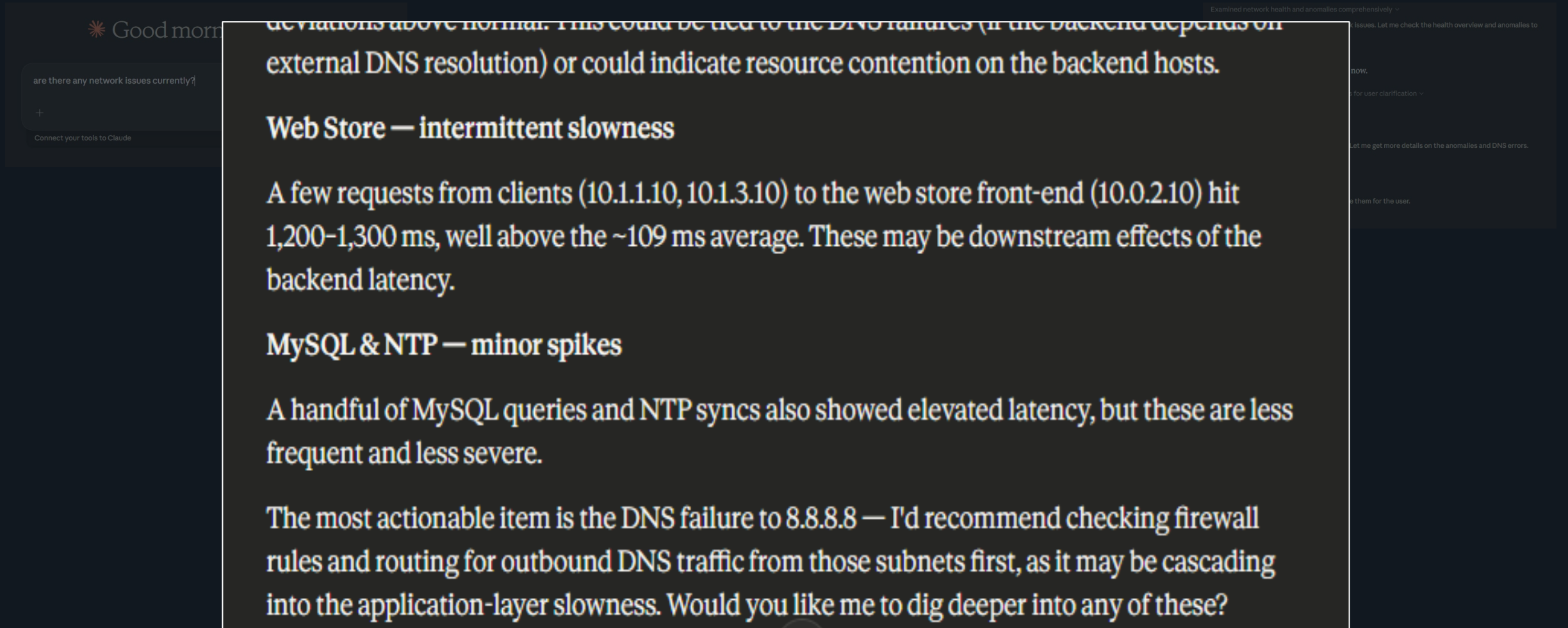
This is the biggest concern. All DNS queries going to 8.8.8.8 (Google's public DNS) are failing at a 100% rate from multiple clients: 10.0.3.20, 10.0.3.21, 10.0.3.22, and 10.1.6.10. That's roughly 19,500 failed lookups out of ~88,000 total DNS transactions. Your internal DNS server (10.0.2.100) appears to be working fine, so this looks like an issue reaching the external resolver — possibly a firewall rule, routing problem, or upstream connectivity issue blocking UDP/TCP 53 to 8.8.8.8.

Shop Backend API — latency spikes

Several transactions between the app servers (10.0.3.x) and backend servers (10.0.4.x) are spiking to over 1,000 ms, against an average of ~61 ms. That's more than 10 standard deviations above normal. This could be tied to the DNS failures (if the backend depends on



ENTERPRISE SCENE #1



The image is a screenshot of a network monitoring dashboard. On the left, a sidebar contains a 'Good morning' greeting, a search bar with the text 'are there any network issues currently?', a plus icon, and a 'Connect your tools to Claude' button. The main area displays a chat conversation with Claude. The chat history shows a user asking about network issues, Claude responding with a detailed analysis of DNS failures and latency, and the user asking for more details. The current chat input field contains the text 'The most actionable item is the DNS failure to 8.8.8.8 — I'd recommend checking firewall rules and routing for outbound DNS traffic from those subnets first, as it may be cascading into the application-layer slowness. Would you like me to dig deeper into any of these?'. The chat interface includes a 'Send' button and a 'Cancel' button.

Good morning

are there any network issues currently?

+

Connect your tools to Claude

Examine network health and anomalies comprehensively. ~

Issues. Let me check the health overview and anomalies to

now.

for user clarification. ~

Let me get more details on the anomalies and DNS errors.

them for the user.

deviations above normal. This could be tied to the DNS failures (if the backend depends on external DNS resolution) or could indicate resource contention on the backend hosts.

Web Store — intermittent slowness

A few requests from clients (10.1.1.10, 10.1.3.10) to the web store front-end (10.0.2.10) hit 1,200-1,300 ms, well above the ~109 ms average. These may be downstream effects of the backend latency.

MySQL & NTP — minor spikes

A handful of MySQL queries and NTP syncs also showed elevated latency, but these are less frequent and less severe.

The most actionable item is the DNS failure to 8.8.8.8 — I'd recommend checking firewall rules and routing for outbound DNS traffic from those subnets first, as it may be cascading into the application-layer slowness. Would you like me to dig deeper into any of these?



ENTERPRISE SCENE #1

 Good morning, Alex

are there any network issues currently?

+

Connect your tools to Claude

Network Operations Center

Incident Triage & Remediation

Last Updated
Dec 18, 2025 15:12 UTC

Active Incidents: 10

P1: 3

P2: 2

P3: 4

P4: 1

TOTAL REQUESTS
88,260

FAILURES
59,346

DNS UPTIME
25.9%

HTTP UPTIME
94.9%

PRIORITY	TYPE	TARGET	ISSUE	FAIL %	DURATION
P1	DNS	192.168.10.121	Complete Outage Server not responding to any queries	100.0%	2h 15m
P1	HTTP	192.168.142.1:6970	Complete Outage Service not responding	100.0%	1h 45m
P1	DNS	10.1.6.11	Complete Outage Server not responding to any queries	100.0%	1h 38m
P2	DNS	192.168.75.10	High Failure Rate 48.6% of queries failing	48.6%	1h 38m
P2	HTTP	192.168.174.1:32843	Service Degradation 25.4% requests failing	25.4%	1h 7m
P3	HTTP	192.168.171.1:32843	Elevated Errors 17.7% error rate	17.7%	52m

and anomalies comprehensively. ~

current network issues. Let me check the health overview and anomalies to capture.

ork health right now.

network concerns for user clarification. ~

or rate at ~22%. Let me get more details on the anomalies and DNS errors.

Let me summarize them for the user.

of notable issues over the last 24 hours:

(critical)

tern. All DNS queries going to 8.8.8.8 (Google's public DNS) are from multiple clients: 10.0.3.20, 10.0.3.21, 10.0.3.22, and 10.1.6.10. That's okups out of ~88,000 total DNS transactions. Your internal DNS ears to be working fine, so this looks like an issue reaching the ssibly a firewall rule, routing problem, or upstream connectivity issue to 8.8.8.8.

latency spikes

etween the app servers (10.0.3.x) and backend servers (10.0.4.x) are s, against an average of ~61 ms. That's more than 10 standard al. This could be tied to the DNS failures (if the backend depends on n) or could indicate resource contention on the backend hosts.

ent slowness

ents (10.1.1.10, 10.1.3.10) to the web store front-end (10.0.2.10) hit ove the ~109 ms average. These may be downstream effects of the

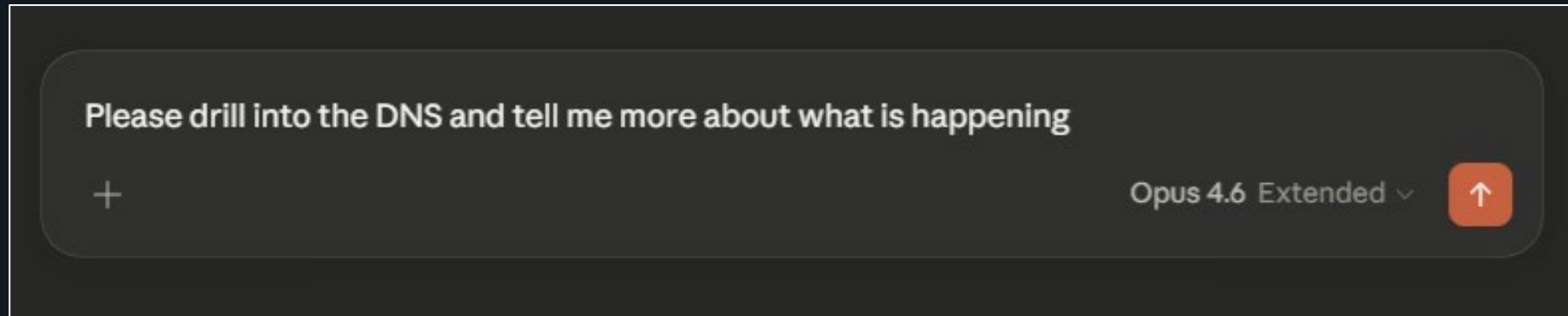
or spikes

eries and NTP syncs also showed elevated latency, but these are less e.

m is the DNS failure to 8.8.8.8 — I'd recommend checking firewall lbround DNS traffic from those subnets first, as it may be cascading er slowness. Would you like me to dig deeper into any of these?



ENTERPRISE SCENE #2



ENTERPRISE SCENE #2

Diagnosed internal domain leakage to external DNS resolver ▾

🕒 The user wants more details on the DNS issues. Let me gather more information.

🔍 Get application summary

Result

🕒 Let me get the DNS error details and also look at the DNS queries for the failing ones.

🔍 Get errors

Result

🕒 Let me also check the DNS queries to see what domains are failing.

🔍 Get dns queries

Result

🕒 Now I can see the pattern clearly. Let me also check what domains are being queried against the internal DNS to compare.

Please drill into the DNS and

+



ENTERPRISE SCENE #2

🔍 Get dns queries

Result

🕒 Now the picture is very clear. Let me also check which unique domains are being sent to 8.8.8.8.

🕒 Need to check the schema first.

🔍 Describe table

Result

🔍 Execute query

Result

🕒 Now I have a very clear picture. The issue is that internal `.lab` domain names are leaking to the external DNS server 8.8.8.8, which obviously can't resolve them and returns NXDOMAIN (response_code 3).

✅ Done

This tells a very clear story. Here's what's happening:



ENTERPRISE SCENE #2

The root cause: internal domain names are leaking to Google's public DNS.

The three app servers (10.0.3.20, .21, .22) are sending DNS queries for internal `.webinfra.lab` hostnames to 8.8.8.8 instead of (or in addition to) your internal DNS server at 10.0.2.100. Google DNS has no idea what `app1.webinfra.lab`, `app2.webinfra.lab`, or `app3.webinfra.lab` are, so it returns NXDOMAIN (response code 3) every time — that's all 19,500 errors.

The breakdown of failing queries to 8.8.8.8:

Domain	A queries	AAAA queries
app3.webinfra.lab	~34,130	~34,134
app1.webinfra.lab	~34,006	~34,008
app2.webinfra.lab	~33,934	~33,938



ENTERPRISE SCENE #2

lb.webinfra.lab

~254

~12

Please drill into the DNS and tell me

+

Every single one returns NXDOMAIN. Meanwhile, the same domains resolve fine against your internal server 10.0.2.100 (response code 0 = success).

There's also a malformed domain showing up — `lb.webinfra.lab.webinfra.lab` — which looks like the search domain is being appended twice, another sign of a DNS client misconfiguration.

What likely needs fixing: The DNS resolver configuration on those three app servers (10.0.3.20-22) is either listing 8.8.8.8 as a primary/secondary nameserver without a split-DNS rule, or the search domain / `resolv.conf` settings are causing internal lookups to fall through to the external resolver. You'd want to ensure internal `.lab` queries only go to 10.0.2.100, and only truly external lookups go to 8.8.8.8.

Want me to check whether this DNS noise is correlating with the latency spikes on the Shop Backend API?

DNS resolver -
ues. Let me gather more information.

ok at the DNS queries for the failing ones.

hat domains are failing.

to check what domains are being queried against the

heck which unique domains are being sent to 8.8.8.8.

hat Internal '.lab' domain names are leaking to the
r can't resolve them and returns NXDOMAIN

appening:



ENTERPRISE SCENE #2

Please drill into the DNS and tell me more about what is happening

+

Opus 4

DNS Service Incident Report

DNS Service Incident Report

Date: December 8, 2025 | **Time Range:** 12:05 - 13:05 UTC | **Source:** NETSCOUT nGeniusONE

Executive Summary

DNS service health has degraded to **73%** due to elevated failure rates and high latency affecting the primary DNS resolver. The Quad9 DNS server (9.9.9.9) is experiencing a **22% failure rate** with average response times of **2,798 ms** (versus expected <100ms). This is impacting downstream services including WebServices, which shows an 18% transaction failure rate likely attributable to DNS resolution timeouts.

DNS Server Status

The following table summarizes the current performance metrics for all monitored DNS servers:

DNS Server	Requests	Failures	Failure %	Avg Latency
9.9.9.9 (Quad9)	408	90	22.06%	2,798 ms
8.8.8.8 (Google)	2	0	0%	15,614 ms
8.8.4.4 (Google)	1	0	0%	2,925 ms
1.1.1.1 (Cloudflare)	72	0	0%	< 1 ms ✓
185.91.129.30	2	0	0%	< 1 ms

Key Findings

- Primary DNS Resolver Degraded:** Quad9 (9.9.9.9) is handling 84% of DNS traffic but experiencing 22% failures and latency 28x higher than acceptable thresholds.
- Google DNS Latency Spike:** 8.8.8.8 shows extreme latency (15+ seconds), suggesting potential routing issues or service degradation.
- Cloudflare Performing Well:** 1.1.1.1 shows zero failures and sub-millisecond response times across 72 requests.
- Downstream Impact:** WebServices showing 17.9% transaction failure rate (780 failed transactions), likely caused by DNS resolution timeouts.

Impact Assessment

Service Impact

The DNS failures are causing cascading effects across dependent services. When DNS resolution fails or times out, HTTP connections cannot be established, resulting in the elevated WebServices failure rate. With 4,359 total web transactions and 780 failures,

Diagnosed internal domain leakage to external DNS resolver ▾

① The user wants more details on the DNS issues. Let me gather more information.

● Get application summary

Result

② Let me get the DNS error details and also look at the DNS queries for the failing ones.

● Get errors

Result

③ Let me also check the DNS queries to see what domains are failing.

● Get dns queries

Result

④ Now I can see the pattern clearly. Let me also check what domains are being queried against the internal DNS to compare.

● Get dns queries

Result

⑤ Now the picture is very clear. Let me also check which unique domains are being sent to 8.8.8.8.

⑥ Need to check the schema first.

● Describe table

Result

● Execute query

Result

⑦ Now I have a very clear picture. The issue is that internal `.lab` domain names are leaking to the external DNS server 8.8.8.8, which obviously can't resolve them and returns NXDOMAIN (response_code 3).

⑧ Done

This tells a very clear story. Here's what's happening:

The root cause: Internal domain names are leaking to Google's public DNS.

The three app servers (10.0.3.20, 21, 22) are sending DNS queries for internal `.webinfra.lab` hostnames to 8.8.8.8 instead of (or in addition to) your internal DNS server at 10.0.2.100. Google DNS has no idea what `app1.webinfra.lab`, `app2.webinfra.lab`, or `app3.webinfra.lab` are, so it returns NXDOMAIN (response code 3) every time — that's all 19,500 errors.

The breakdown of failing queries to 8.8.8.8:

Domain	A queries	AAAA queries
app3.webinfra.lab	~34,130	~34,134
app1.webinfra.lab	~34,006	~34,008
app2.webinfra.lab	~33,934	~33,938
lb.webinfra.lab	~254	~12

Every single one returns NXDOMAIN. Meanwhile, the same domains resolve fine against your internal server 10.0.2.100 (response code 0 = success).

There's also a malformed domain showing up — `lb.webinfra.lab.webinfra.lab` — which looks like the search domain is being appended twice, another sign of a DNS client misconfiguration.

What likely needs fixing: The DNS resolver configuration on those three app servers (10.0.3.20-22) is either listing 8.8.8.8 as a primary/secondary nameserver without a split-DNS rule, or the search domain / `resolver.conf` settings are causing internal lookups to fall through to the external resolver. You'd want to ensure internal `.lab` queries only go to 10.0.2.100, and only truly external lookups go to 8.8.8.8.

Want me to check whether this DNS noise is correlating with the latency spikes on the Shop Backend API?



ENTERPRISE SCENE #3

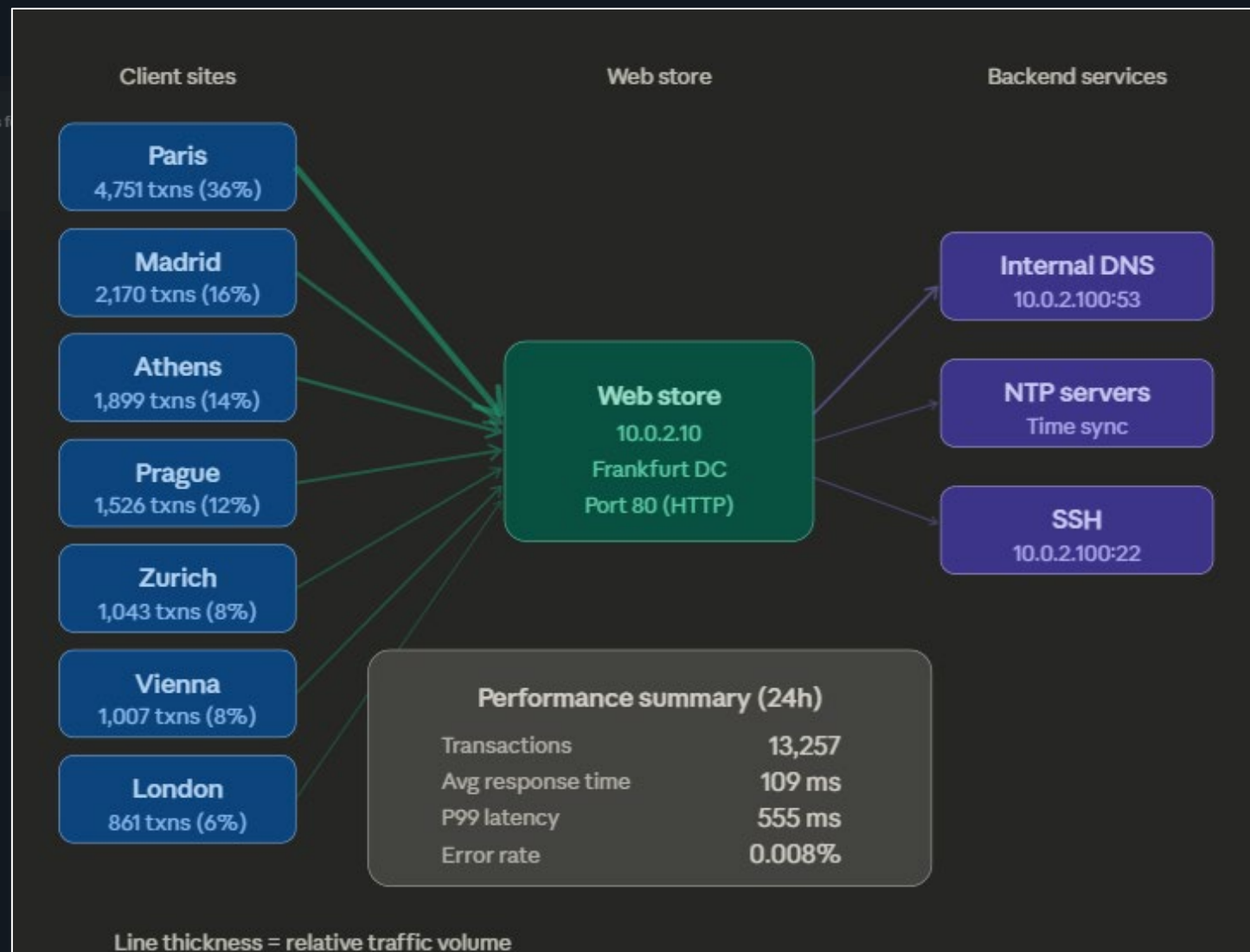
I would like to understand more about the traffic flows for the Web Store. Can you summarize this?



Opus 4.6 Extended ▾



ENTERPRISE SCENE #3

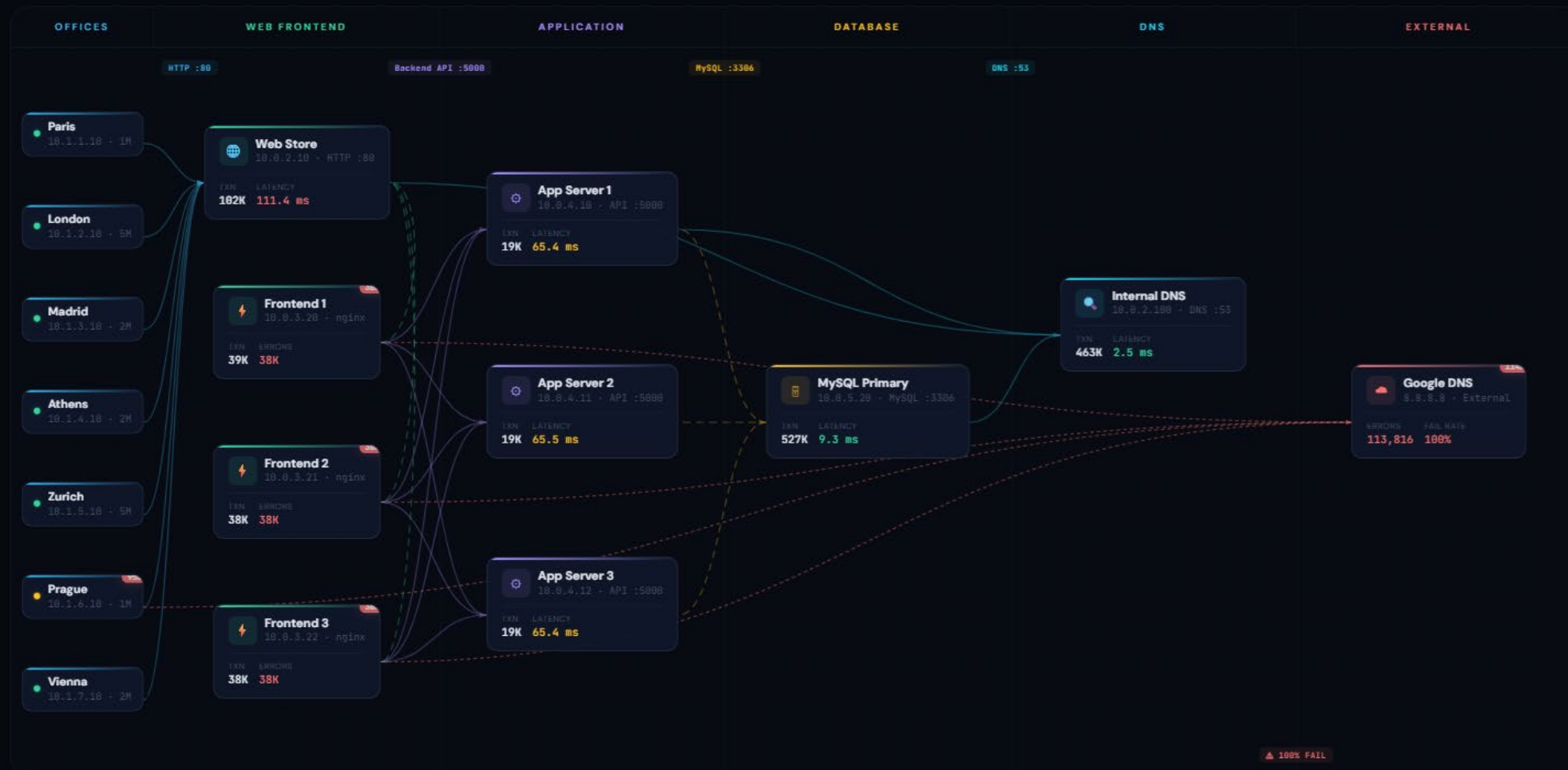


ENTERPRISE SCENE #3

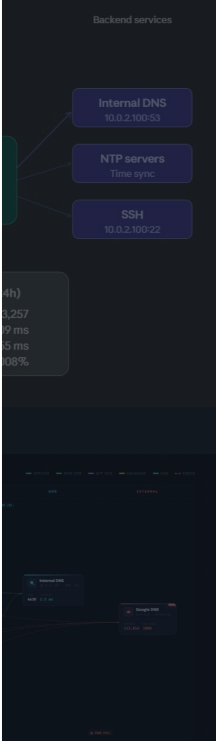
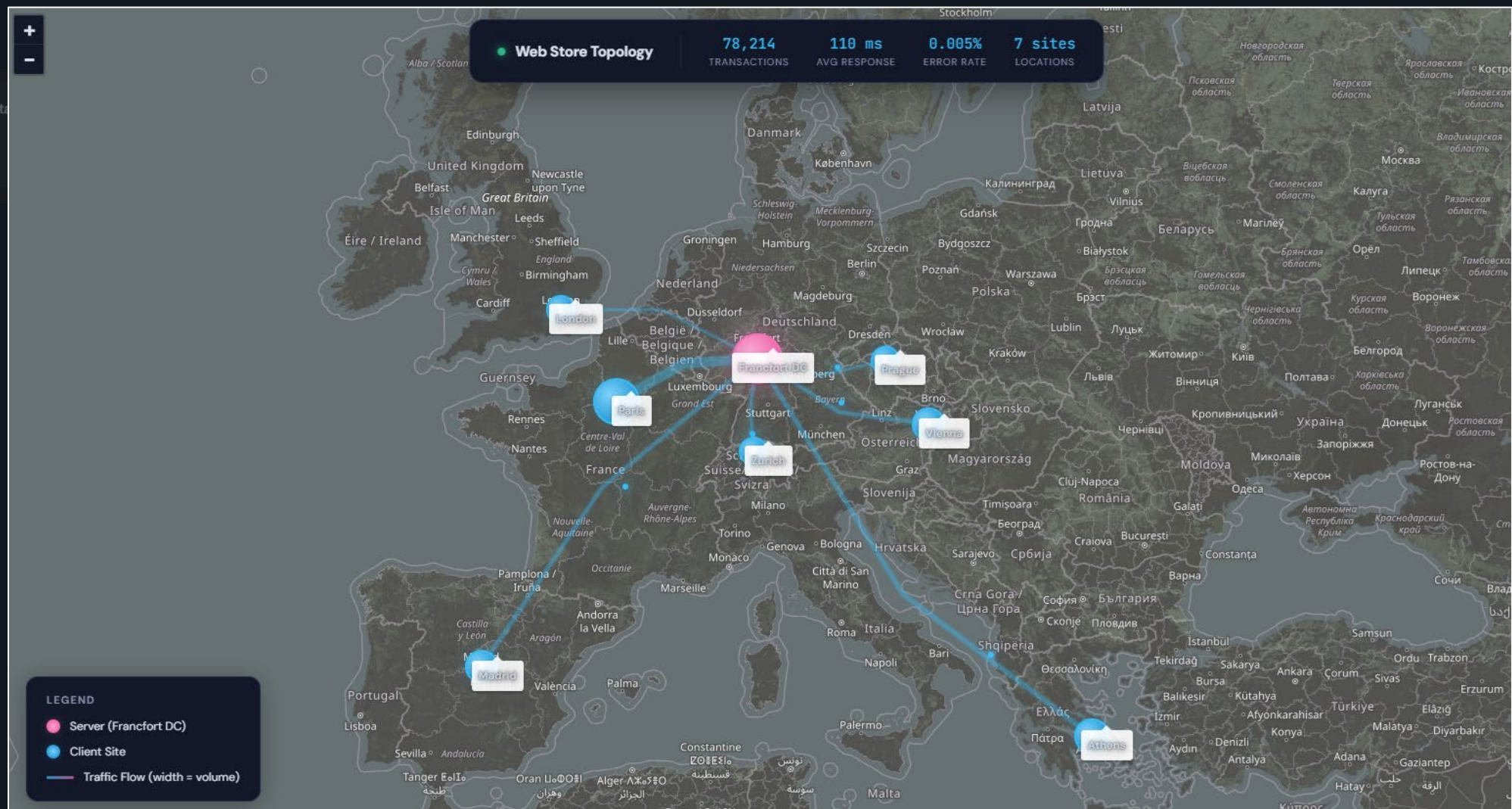
Logical Application Topology

Service dependency map — Frankfurt DC & 7 branch offices - 24h window

OFFICES WEB TIER APP TIER DATABASE DNS ERROR



ENTERPRISE SCENE #3



ENTERPRISE SCENE #3

I would like to understand more about the traffic flows for the Web Store this?

+

Web store — performance dashboard

00:00 – 04:20 UTC - April 2, 2026

TRANSACTIONS

13,257

Across 5 hours

AVG LATENCY

82 ms

P50: 96 ms - P90: 257 ms

PEAK LATENCY

1,328 ms

12x above average

ERROR RATE

0.008%

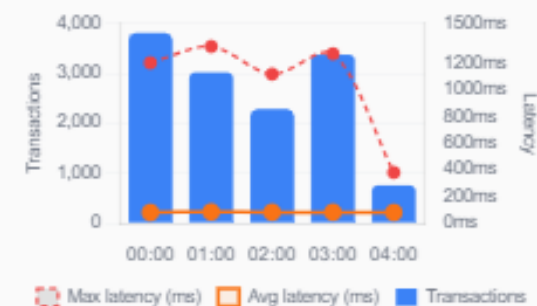
1 failure (Prague)

DATA TRANSFERRED

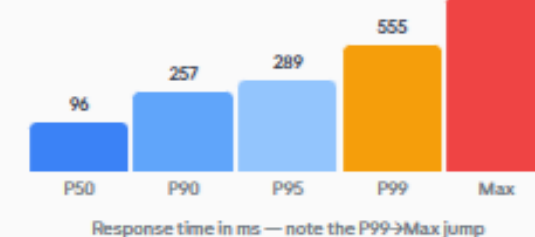
44 MB

9.4 in - 36.2 out

Throughput & latency by hour

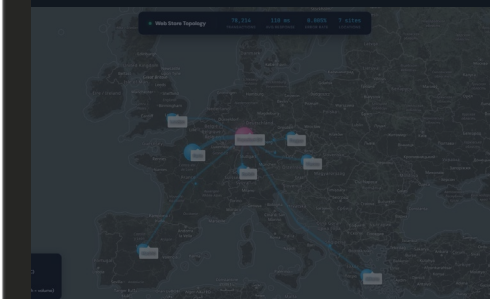
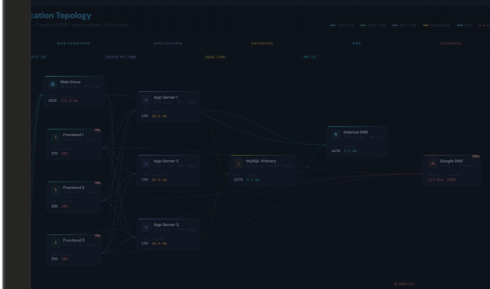
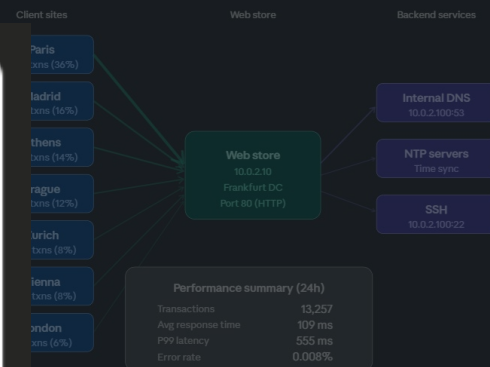


Latency distribution

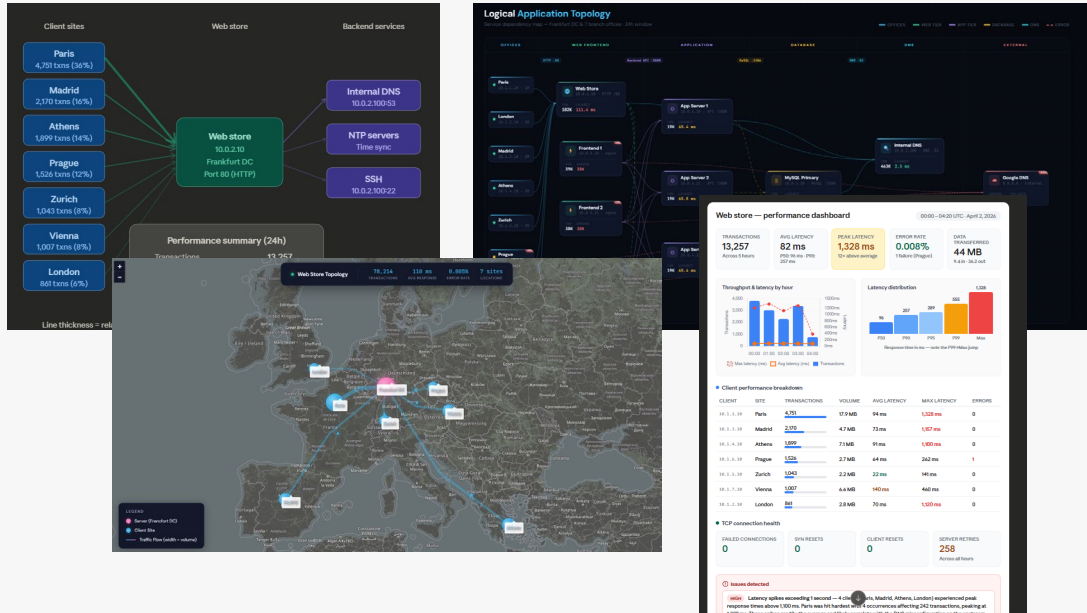


Client performance breakdown

CLIENT	SITE	TRANSACTIONS	VOLUME	AVG LATENCY	MAX LATENCY	ERRORS
10.1.1.10	Paris	4,751	17.9 MB	94 ms	1,328 ms	0
10.1.3.10	Madrid	2,170	4.7 MB	73 ms	1,157 ms	0
10.1.4.10	Athens	1,899	7.1 MB	91 ms	1,100 ms	0
10.1.6.10	Prague	1,526	2.7 MB	64 ms	262 ms	1



Remember – Think beyond dashboards



AI reasons with ASI data.

- **Accurate & Truthful:** grounded in real data
- **Signal, Not Noise:** AI focuses on what matters
- **Insight-Driven Decisions:** not chart-driven

CxO want to work towards self healing IT Operations

Thank you



NETSCOUT®

Guardians of the Connected World