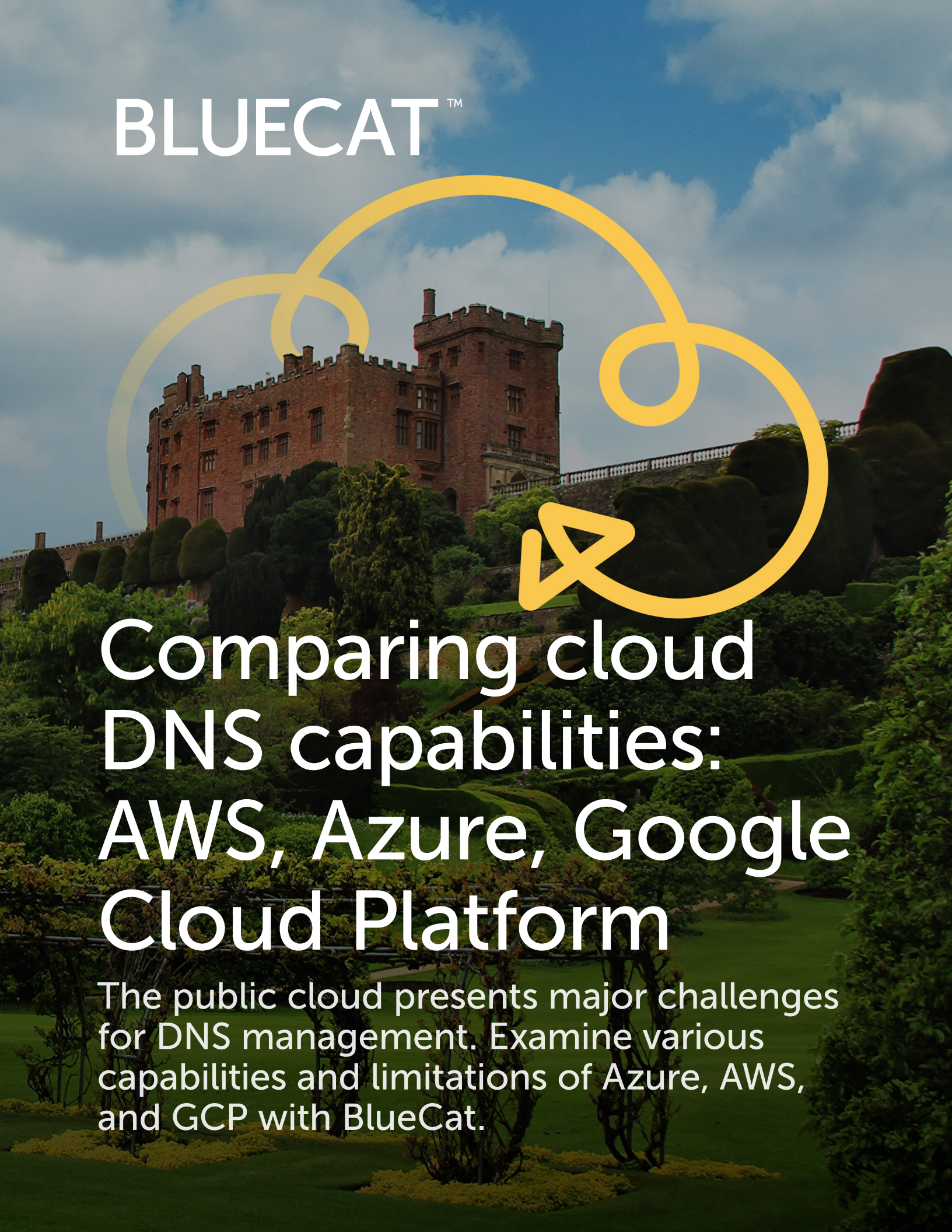


BLUECAT™

Comparing cloud DNS capabilities: AWS, Azure, Google Cloud Platform

The public cloud presents major challenges for DNS management. Examine various capabilities and limitations of Azure, AWS, and GCP with BlueCat.

Cloud is a great way to consume infrastructure.

Its self-service and on-demand capabilities for computing, storage, and networking make it an attractive alternative to maintaining data centers. The pay-as-you-go pricing is more flexible, which is especially important to businesses that value operational costs over capital investments. Furthermore, economies of scale allow cloud providers to create better experiences.

This is why developers and DevOps engineers love the cloud: The breadth and depth of services offered are hard to match in on-premises. Businesses love it, too. Infrastructure as a service makes organizations agile and nimble, removing IT as a limiting factor to delivering value.

Cloud also opens up resource consumption to more than just the IT infrastructure and operations teams. It democratizes IT across organizations for quickly building new applications using standard 'as-a-service' building blocks.

Ultimately, it makes consuming infrastructure easy by abstracting some of the complexity and removing bottlenecks to getting started.

But for all its benefits, it still requires you to have at least some working knowledge of networking, routing, DNS, DHCP, and IP address management (together known as DDI).

Teams have their own best practices, varying levels of experience, and don't know what they don't know. They cannot foresee the issues coming down the road that experts can. And they can't solve them before they become problematic. In the end, their cloud architectures suffer.

This post will discuss some of the major challenges to DNS management that public cloud providers present, including islands of cloud and walled gardens. Furthermore, it will closely examine various capabilities and limitations of Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP), particularly as they impact DNS. Finally, it will touch on how including the DDI team can set you on the right path to solving cloud complexity.

Islands of cloud decentralize DNS management

With many teams using the cloud, they create their own little islands of cloud. This creates problems with visibility across all cloud accounts (or virtual private clouds [VPCs]) at scale.

The ease of cloud consumption lowers the bar to get started; teams often 'just start' with app development without involving IT teams. Security, performance, manageability, and cost are cans that are kicked down the road—sometimes for too long. IT is involved too late in the game, making it hard for them to integrate these applications into the enterprise. These shadow IT projects can easily get out of hand.

The resulting patchwork of barely interconnected services across heterogeneous technology stacks effectively decentralizes DNS management across tens or hundreds of accounts and VPC and thousands of zones and records. This compounds complexity, creates interoperability issues, and reduces visibility and control. All of which affect application stability.

The complexity isn't in doing it or doing it right once. The complexity is in doing it right, consistently, across dozens of teams that aren't DDI experts.

This goes against the design tenets of networking and DDI teams, who build a well-oiled machine for DDI that works across the enterprise.

As many teams discover down the line, DDI is critical infrastructure for running and developing applications. That means that DDI has to be consistent, available, and reliable. Visibility into, and control over, DDI across the organization is vital for well-functioning applications. It impacts performance, cost, reliability, security, and compliance.

Walled gardens are a problem for multi- and hybrid cloud

Unfortunately, opening up cloud services for easier DNS centralization and allowing for interoperability across on-premises and different public clouds isn't in the best interest of public cloud companies.

Locked into an ecosystem

Instead, these companies hope to lock you into their ecosystem. They offer a set of services that integrate and interoperate with each other but not outside of their specific walled garden. The user experience is great as long as you stay within the garden.

Furthermore, things become purposefully deficient if you venture outside of their boundaries. In lieu of using on-premises or a competitor's cloud services, they want you to use more of their services.

In an increasingly multi-cloud world, these walled gardens are becoming an architectural hazard. Interoperability is at risk. It limits visibility into an organization's cloud estates, increasing complexity and cost to keep applications running smoothly. This is why hybrid and multi-cloud management is so difficult to get right.

Heterogeneous support for DNS features

One of the more striking challenges is the heterogeneous support for basic DNS features, each with its own capabilities and limitations. This jungle of inconsistency and associated complexity is often the root cause of interoperability problems between clouds and on-premises resources.

Solving it requires specialized knowledge of the drawbacks of each of the public clouds.

For instance, every public cloud has limiting and differing support for DNS record types. And the landscape is constantly evolving. Public cloud vendors change their walled gardens continuously, sometimes for the better and sometimes for worse.

But knowing that cloud providers are actively counteracting you is only half the battle. The other half is managing those varying capabilities across a heterogeneous environment of on-premises, Azure, AWS, and GCP.

Capabilities and limitations of public cloud

The sections below should give you a solid understanding of which public cloud features to look out for. (Of course, the accuracy and completeness of what is provided here will change over time.) They will also discuss how these features impact your enterprise DNS interoperability designs, and where the public clouds fall short and create DNS dead ends.

And there are many of these dead ends.

Cloud DNS capabilities by provider

	aws	Azure	Google Cloud Platform
Resolving between accounts	✗	✗	✗
Resolving between VNETs/VPCs	✗	✗	✗
Zone delegation (to)	✓	✓	✓
Zone delegation (from)	✗	✗	✗
Private endpoints	✓	✓	✓
Forced internal DNS	✓	✓	✓
Record types supported	A, CNAME, MX, AAAA, TXT PTR, SRV, SPF, NS, SOA	A, AAAA, CAA, CNAME, MX NS, PTR, SOA, SRV, TXT SPF supports via TXT records	A, AAAA, CAA, CNAME, DNSKEY, DS, IPSECKEY, MX, NAPTR, NS, PTR, SOA, SPF, SRV, SSHFP, TLSA, TXT
DNSSEC support	✓	✗	✓
DoH/DoT support	In progress	✗	✓
DNS firewalling	Not yet intended for hybrid and multi cloud use	✗	✗
Domain registration	✓	✗	✗
Maximums (zone)	10	250	10000
Geo IP tagging	✓	✗	✓
Cost	$0.40/million queries	$0.40/million queries	$0.40/million queries

Resolving records beyond a service boundary

As previously discussed, cloud vendors want you to stay within their walled gardens. This means that, by design, resolving records outside of an account or VPC is cumbersome. Each cloud has different (and sometimes conflicting or missing) options for zone delegation, recursion, or even forwarding.

In a best-case scenario, making resolving work for your internal zones requires connectivity (like a VPN direct connection) and has associated costs (like running a DNS forwarder in a cloud virtual machine). Realistically, you're making your architecture more complex and more fragile to work around the obstructions cloud vendors put in place to keep you within their walled garden. But even then, you'd need a third-party product (like BlueCat's Cloud Discovery & Visibility) to create end-to-end visibility and discover resources across service boundaries.

Often, you lack visibility into the right VPCs or can't automatically discover resources due to each cloud's limitations. As a result, you can't resolve the right resources inside of them and end up with a fragmented and incorrect view of your cloud world.

Many different boundaries to take into account

Whether this is across cloud vendors, across cloud accounts, or across virtual networks (VNets) or VPCs, there are many different services boundaries to take into account. Each has its own gotchas and hidden limitations for services inside that boundary.

For instance, public cloud vendors don't support zone delegation to authorities outside of their walled gardens. This is crucial for smooth-running hybrid and multi-cloud setups. Often, in an attempt to keep you in their walled garden, you can delegate to them but not from them.

Public clouds will not release the authority of their DNS zones to anyone, forcing you to manually create complex multi-hop forwarding tables. These tables are hard to understand, error-prone, and fragile. In a similar fashion, vendors actively counteract transfer zones. In best-case scenarios, you can transfer zones to, but not from, cloud providers.

Various workarounds to resolve between accounts

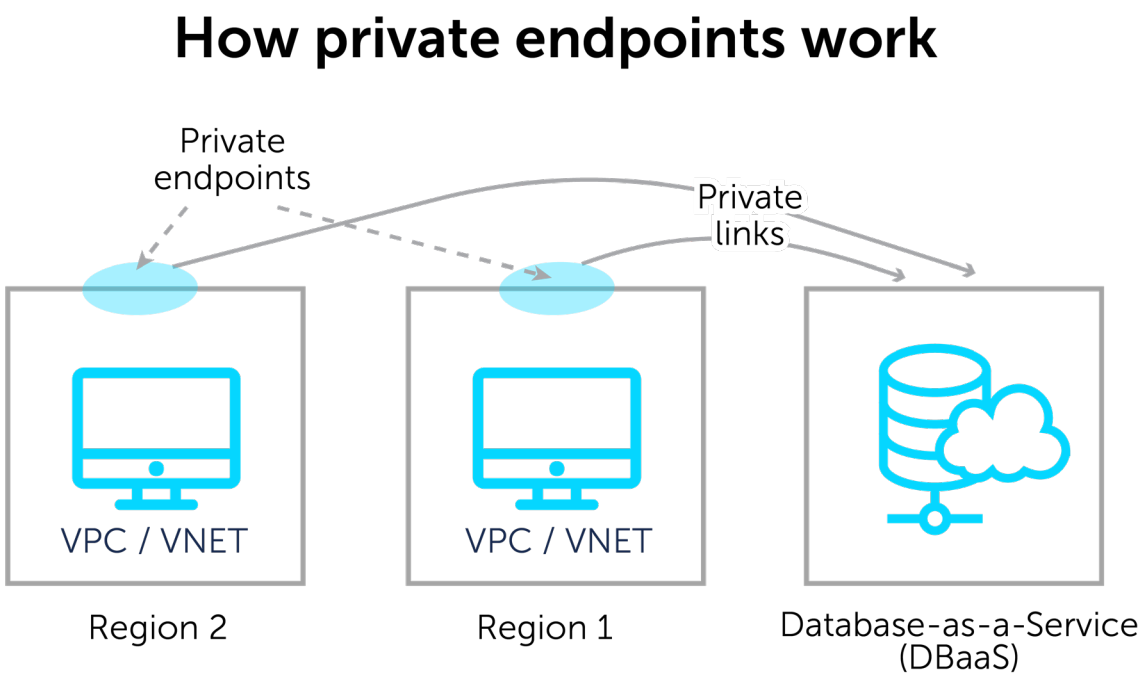
Resolving between accounts can also be a challenge. And the road to success is laden with complex forwarding rules from a central account to other accounts. Furthermore, the solutions for each cloud are different.

For instance, AWS's solution to this problem is to simplify DNS management in a multi-account environment with Route 53 Resolver. The most common way of resolving private zones across different accounts is by implementing a local DNS server in each account and using conditional forwarders. However, this approach does not scale well in large multi-account scenarios and is very error-prone.

Private endpoints

Private endpoints are IP addresses with a DNS name in a tenant for a cloud service, like S3. They are helpful to access cloud resources inside a tenant, so you don't have to egress out of that tenant (and you won't get a bill for traffic or have a performance or bandwidth penalty).

But with many similar services inside of many different accounts and VPCs, how do you manage private endpoints? Multiple similarly named services per tenant across hundreds or thousands of accounts quickly become confusing and hard to manage. They also require using the public cloud vendor's DNS with local forwarding. This increases complexity and the risk of getting the wrong response due to misconfiguration.



Forced internal DNS

Even if you don't want to use public cloud DNS services, you're forced to use them. Spinning up a resource, like a virtual machine, will include a DNS record for that resource. And developers then start to use those records.

But without bringing those records into enterprise management, applications, internet-facing websites, or even automation workflows, pipelines will start to break. There will be missing or unresolvable records and wrong answers due to the duplication of record names across VPCs.

To solve these issues, organizations use complex, fragile, and error-prone forwarding rules from their on-premises networks. Teams must manually untangle the tangled mix of VPCs across cloud accounts and vendors. Or they turn to a product like Cloud Discovery & Visibility to generate custom zones automatically.

Another oft-used approach is to never use those cloud-internal records outside of that cloud boundary, preventing mishaps in the first place. Furthermore, they can require teams to create their own records using automation when creating and changing cloud services.

Supported record types vary between cloud vendors

There is no consistent coverage of supported record types across the public cloud vendors. The more specialized record types especially suffer. As you can see in this table, support varies wildly:

DNS record type Support

	aws	Azure	Google Cloud Platform
A	✓	✓	✓
AAAA	✓	✓	✓
CAA	✗	✓	✓
CNAMES	✓	✓	✓
MX	✓	✓	✓
TXT	✓	✓	✓
PTR	✓	✓	✓
SRV	✓	✓	✓
SPF	✓	via TXT	✓
NS	✓	✓	✓
SOA	✓	✓	✓
DNSKEY	✗	✗	✓
DS	✗	✗	✓
IPSECKEY	✗	✗	✓
NAPTR	✗	✗	✓
SSHFP	✗	✗	✓
TLSA	✗	✗	✓
All others	✗	✗	✗

And without broad support for record types, you may end up having to run your own third-party DNS infrastructure. You might know this before you venture out into the cloud and thus carefully and consciously design your own DNS infrastructure while effectively partnering with your internal teams.

But this is quite different from scrambling to set up a half-baked DNS infrastructure without the right stakeholders on board.

You end up in a pressure cooker. Unknown limitations across cloud vendors will rear their heads at the most critical phase of your cloud migration project. For instance, you may find that the cloud vendor is lacking support for Name Authority Pointer (NAPTR) (for SIP and other VoIP services) or DNSKEY (for DNSSEC).

Mixed DNSSEC support

A prime example of heterogeneous support for a technology that's quickly becoming a security and compliance staple is DNSSEC.

DNSSEC validates DNS requests, protecting against spoofing attacks where rogue DNS servers send malicious answers back to clients. It digitally signs DNS records, so the client can check whether the response is authentic.

While AWS and GCP provide support for DNSSEC, Azure does not currently support it at all. Partial or entirely lacking support was acceptable as the technology was being adopted. But there's no real excuse for lacking support in 2021.

DNSSEC is a security best practice and should be table stakes for all cloud providers. However, the reality is that you'll need a third-party solution to provide full DNSSEC coverage. This is a critical feature to prevent DNS spoofing attacks and phishing for any application dealing with personally identifiable information or other sensitive data.

Mixed DoH and DoT support as well

For all its benefits, DNSSEC does not protect the last mile of DNS between the client and the server against snooping eyes.

DNS over HTTPS (DoH) and DNS over TLS (DoT) are two similar technologies that encrypt communication between DNS client and server. By doing so, intermediates and anyone with access to the network, like ISPs and cloud (connectivity) providers, cannot track DNS requests.

As with DNSSEC, support for DoT and DoH is quickly becoming a common requirement, especially because DNS data can be sensitive and is easily monetized by cloud vendors. AWS is currently working on their support for DoH and DoT; Google Cloud supports it. Azure doesn't currently support DoH or DoT.

Nascent support for DNS firewalling

In addition to DNSSEC and DoH/DoT, there is more DNS security under the horizon. While commonplace in enterprise deployments, DNS firewalling support in public clouds is still in its infancy.

Currently, the only cloud provider that offers this is AWS. And they've only recently begun supporting it with their Amazon Route 53 Resolver DNS Firewall. It offers allowlists and blocklists for resolving records. For added security, it also supports two managed domain lists—malware domains and botnet command-and-control domains.

While new to the market and expected to improve, there are several immediate shortcomings of the AWS Route 53 Resolver DNS Firewall. This includes a lack of automatic detection for malware with dynamically generated domain names (using domain generation algorithms). There is also no detection for DNS data exfiltration (which misuses DNS to exchange a large amount of unstructured data between attacker and victim). Furthermore, there are no custom block/allow lists, which could be a problem for those with regulatory requirements to support their own customer blocklists.

For now, Route 53 Resolver DNS Firewall is fairly limited and not intended for use in hybrid or multi-cloud scenarios. Using the DNS firewall requires you to ship all of your traffic to the cloud, incurring the traffic costs as well as Route 53 Resolver DNS Firewall costs.

Domain name registration

AWS is the only cloud provider that will let you directly register a domain with them while utilizing your AWS services.

Both GCP and Azure rely on third-party services for domain name registration, with lackluster integration and support to boot. It requires an additional service from a third party, and additional automation and glue code to integrate it into automated processes. Lack of complete support for functionality like this makes the DDI landscape more complex and more fragile than it needs to be. Ultimately, it impacts the performance metrics of DDI solutions.

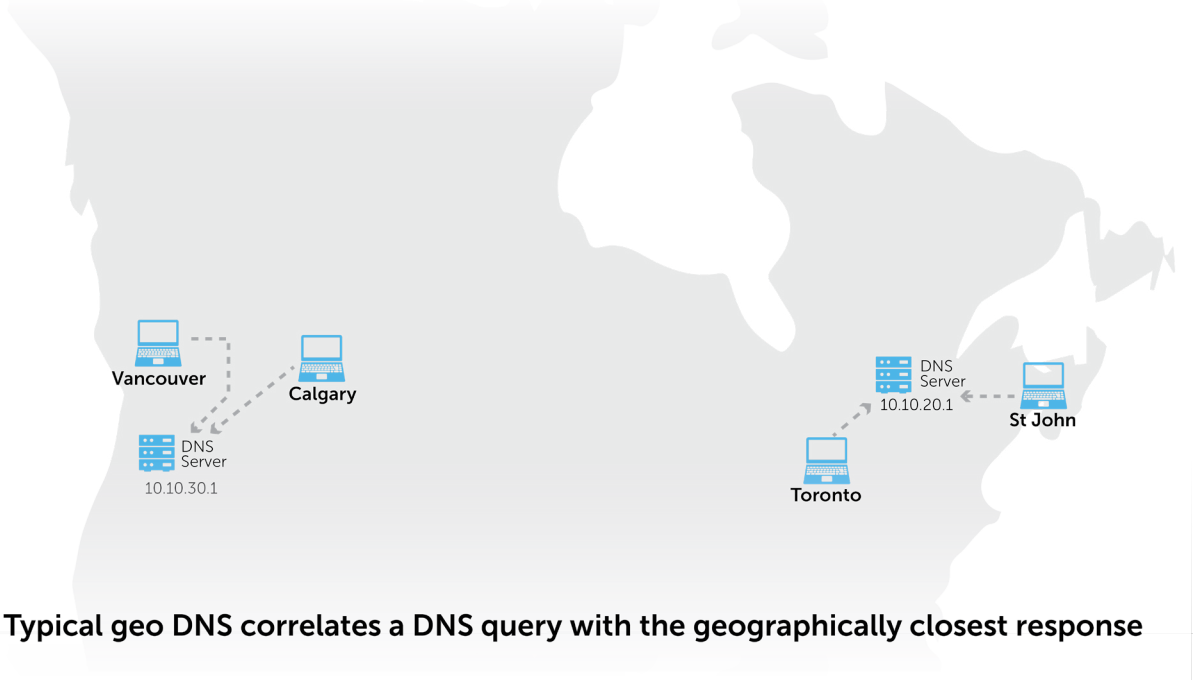
Zone maximums

Each cloud has artificial limitations and different capabilities for sharing private DNS zones across cloud tenants (or VPC accounts and the like). This sometimes requires complex forwarding rules. All three cloud providers limit the number of DNS zones you can host per account: AWS limits you to 10, Azure is 250, and GCP is 10,000. Of course, these limits are artificially put in place and can be lifted or changed by talking to their respective sales teams.

Geolocation inconsistency

There is no guarantee that the DNS resolver that the cloud vendor provides is close to the DNS clients in question. In fact, there's little to no control over or visibility into where cloud providers are routing your DNS requests. Furthermore, these routes can differ and change over time.

When these routes change, it creates problems with consistency. You are often left in the dark as to why DNS responses are suddenly inconsistent and reply with a non-local IP address for the requested record. Should the content be modified based on geo-location, this becomes a problem. And while GCP and AWS support geo IP tagging, Azure does not.



Typical geo DNS correlates a DNS query with the geographically closest response

But sometimes this happens

Indirect costs

There is one thing you can be certain will be part of every decision about cloud direction: cost. It's obviously important when we are talking about cloud DNS, as you pay per query. With Azure, AWS, and GCP, the direct cost is about $0.40 per million queries.

However, oftentimes it's not the direct cost of the services that inflates the bill. It's the indirect costs of adjacent services that are required to run an enterprise-grade DNS infrastructure across the hybrid and multi-cloud. Furthermore, you are running it when the cloud is carefully designed with clever pricing to lock you into a vendor's ecosystem and keep you within their walled garden.

Solving multi-cloud complexity

The capabilities and limitations of public cloud DNS services are nuanced and subtle. But the smallest of detail can have a large impact on interoperability. Solving these issues requires the DDI team to be part of the cloud conversation. This helps to create a solution with broad support of enterprise DDI features across the multi-cloud spectrum regardless of their native capabilities and limitations.

By combining forces, business teams can move full steam ahead with creating applications and functionality. Meanwhile, the DDI experts can ensure interoperability across clouds and back to the on-premises data centers is taken care of. This will ensure visibility and control to optimize security, performance, and cost.

Their DDI expertise can also help you to prevent the proverbial pitfalls that occur six months down the road. In the end, you can continue to create value instead of being bogged down by toil and technical debt.