DNS - Beta

This page describes systems which have not yet been deployed to production (SN 1{0,1}) and configurations that can be made after the current DNS servers are retired (DoH @10.10.10.10).

DNS Services

You are welcome to, but are not required to use **10.10.10.10 and 10.10.10.11** for your DNS servers within NYC Mesh

NYC Mesh maintains authoritative and recursive resolvers that are hosted within NYC Mesh. The authoritative server is primarily used to resolve records *about* NYC Mesh while the recursive resolver resolves queries *from* NYC Mesh.

Types of DNS Servers

	Authoritative	**Shared Private (Recursive) Resolver**
Would resolve a query for _	wiki.mesh.nycmesh.net	wikipedia.org
Resolves queries sent by _	DNS servers from the internet resolving queries	Members within NYC Mesh
Responds to queries about _	NYC Mesh	Anything
Available from _	The internet and inside the mesh	Inside the mesh
Responds to _ queries	Standard DNS	Standard DNS and DoH

.mesh

NYC Mesh's internal DNS servers include a "fake" internal-only top-level domain (TLD) `.mesh` (dot mesh) which is primarily used by volunteers and enthusiasts to shorten URLs within the mesh. Special configuration is needed to allow Mikrotik devices to consistently resolve `.mesh` TLD records within the mesh.

The `.mesh` zone is also externally available via `.mesh.nycmesh.net`.

Privacy

NYC Mesh does not log who makes what DNS queries or which records/domains are queried.

General statistics are kept about the total number of queries made and how long it takes the servers to respond to them for the sole purpose of continual operation of the DNS servers.

Using the NYC Mesh shared private resolver means that less overall DNS queries are made outside of the mesh. For example, if 10 users query for example.com before the record's TTL expires, the server only needs to resolve the record 1 time, and the cached copy is used for the remaining 9 times. This also happens to make it difficult for a passive eavesdropper outside of the mesh to know who requested records for popular domains.

NYC Mesh also resolves encrypted queries made via DoH for added privacy and security.

DoH

DNS over HTTPS (DoH) is supported by the recursive resolvers hosted by NYC Mesh. Following standard practice, DoH traffic is terminated at the NYC Mesh recursive resolvers, which then makes one or more standard DNS queries on the internet, to then pass the response back to the requester via HTTPS. The privacy/security benefit of using DoH is that a passive or active network attacker within the mesh is not able to eavesdrop on DNS queries between the system that made the DoH query and the DNS server hosted by NYC Mesh.

DoH is supported by Mikrotik RouterOS devices, though it is not enabled by default.

Censorship

NYC Mesh does not filter the content of DNS requests.

DNS Infrastructure

Infrastructure for the NYC Mesh internal DNS servers is managed via Terraform and Ansible and deployed via GitHub Actions in the nycmeshnet/nycmesh-dns repository on GitHub. The same repository also contains the configuration for NYC Mesh organization domains which are not hosted within the mesh.

For redundancy, DNS servers are hosted at multiple locations within NYC Mesh. The recursive resolvers used by NYC Mesh members are available at the same IP addresses (anycast). Member's queries are routed to the nearest server as determined by

OSPF.

- SN1 (legacy, to be removed)
- SN3
- SN10 - Soon?
- SN11 - Soon

Technical Summary

Server (Authoritative)	Knot DNS
Server (Recursive)	Knot Resolver
Routing	FRR
Monitoring	InfluxDB + Grafana + DataDog + Uptime Kuma
Log Aggregation	DataDog
Deployment	GitHub Actions
IaC	Terraform + Ansible
Virtualization	Proxmox (KVM)
Operating System	Debian

Deployment

Changes to the infrastructure and server configuration for the NYC Mesh DNS servers are managed via Infrastructure-as-Code(IaC) in the nycmeshnet/nycmesh-dns repository. Changes to the Terraform and Ansible are automatically deployed after they are approved by specific NYC Mesh volunteers.

Changes to the `.mesh` and `.mesh.nycmesh.net` zone are managed in a zone file checked into the nycmeshnet/nycmesh-dns repository. After changes to these files are reviewed, approved, and merged by specific NYC Mesh volunteers, the DNS servers periodically update the DNS records via "git pull" operations from the servers themselves.

Anycast DNS and IPs

Recursive Resolver IPs

10.10.10.10 - Primary recursive resolver within the mesh
10.10.10.11 - Secondary recursive resolver within the mesh

The reason for two IPs rather than one is for redundancy, resiliency, and load balancing. While the servers at both IPs are designed and configured to be able to handle the full traffic from the mesh, splitting the load through multiple servers reduces the risk of service interruptions and allows clients to fail-over between IPs.

Anycast

The DNS servers are commonly made available through a "trick" called Anycast. With anycast, many DNS servers all present the same virtual IP. They announce this IP in the routing table (OSPF within the mesh routing table). With this, the clients believe they all have a very short route to the same network, but in fact it is a copy of the same service running many times with the same configuration. Any of the services may answer the request equally well. Reply packets are sent via the normal means.

Within NYC Mesh, this means that a member in Harlem sends a queries to 10.10.10.10, OSPF is used to route the packets to SN11, while the same requests made in Park Slope are routed to SN3. In each case, OSPF picks the shortest available route. Also note that if the server at SN3 is taken down for maintenance (and is removed from the routing table), queries will be sent to the closest available server at another Super Node.

Revision #2
Created 1 November 2024 01:39:36 by James
Updated 1 November 2024 04:13:24 by James