DNScontrol: Version-Controlled DNS Management for Modern Networks

Presented at RIPE NCC Network and DNS Operations Session

>_Whoami

Kourosh Tafreshi
Site Reliability Engineer@Bolt
AWS and Terraform certified

A computer-networks enthusiastic who keeps machines up and running!

➔ **Contact**

kouroshtaf@gmail.com

➔ **When? Where?**

24 - 25 September 2024

Bishkek, Kyrgyzstan

Agenda

- ➔ Introduction
- ➔ The Need for Version-Controlled DNS Management
- ➔ DNSControl Overview
- ➔ Real World Use cases
- ➔ Implementation
- ➔ Best Practices
- ➔ Conclusion
- ➔ Q&A

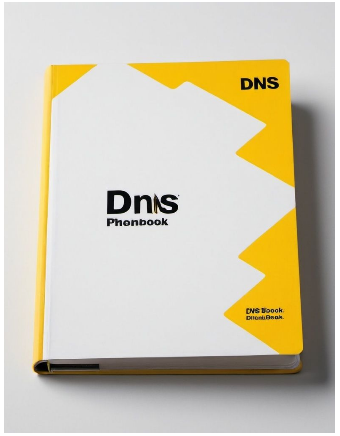
The Phonebook of the Internet *is broken!*

Imagine the chaos if phone numbers kept changing without a central system or record.

That's the potential risk with unmanaged DNS configurations!

Challenges of Modern DNS Management

- Managing records across multiple providers
- Complex configurations with frequent updates



The Phonebook of the Internet *is broken!*

DNS management sucks if:

- Having more than 1 Domain registrars
  - Local Regulated ccTLDs: cctld.kg, nic.ir, …
  - Open gTLDs: Amazon Route53, Gandi, Domainnameshop, …

- Having more than 1 DNS Provider
  - Self hosted/manged: BIND, PowerDNS, ...
  - Cloud: Cloudflare, Amazon Route53, …
  - Mix of above!

How many of you have heard of *Version Control*, *Git*, *IaC* and *GitOps*?

Tip

Version Control: System for tracking and managing changes to files over time.

Git: Distributed version control system for source code management.

IaC: Infrastructure as Code; managing IT infrastructure using code and automation.

GitOps: Infrastructure management using Git as a single source of truth.

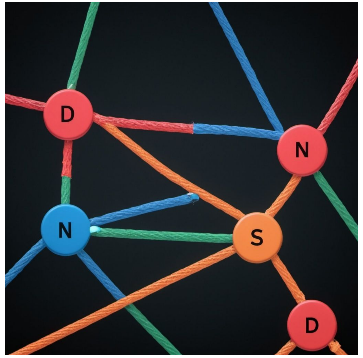
The Need for Version-Controlled DNS Management

Common Issues:

- Inconsistent changes across providers
- Lack of audit trails for troubleshooting
- Difficulty rolling back mistakes
- Human errors in manual updates

Benefits of Version Control:

- **Traceability:** See who made changes and when
- **Rollback Capability:** Easily revert to previous configurations
- **Collaboration:** Work together on DNS updates securely
- **Automation:** Streamline deployments and reduce errors



DNSControl Overview

What is DNScontrol?

- Open-source tool for managing DNS records across multiple providers
- Uses a Domain-Specific Language (DSL) for easy configuration
- Integrates with popular version control systems (e.g., Git)

Key Features:

- Multi-provider support (Cloudflare, AWS Route 53, Google Cloud DNS, etc.)
- Unified configuration for all your domains
- Preview changes before applying them
- Integrates with CI/CD pipelines for automated deployments

Tip

dnscontrol.org

docs.dnscontrol.org

DNSControl! DNS as code.

Universal DNS Language(DSL)
One syntax to rule them all - speak to multiple providers effortlessly!

GitOps Revolution
Transform DNS changes into PR-driven processes.
Boost efficiency, improve accuracy, and empower developers.
Say goodbye to email chains and ticket limbo!

Democratize DNS Management
DNSControl makes it safe for non-experts to contribute.
Spread the load, reduce your stress - be the DNS hero, not the bottleneck!

Tip

Uses a Domain-Specific Language aka DSL.

Examples

➔ Typical DNS Records

```
D("example.com", REG_MY_PROVIDER, DnsProvider(DSP_MY_PROVIDER),
    A("@", "1.2.3.4"),  // The naked or "apex" domain.
    A("server1", "2.3.4.5"),
    AAAA("wide", "2001:0db8:85a3:0000:0000:8a2e:0370:7334"),
    CNAME("www", "server1"),
    CNAME("another", "service.mycloud.com."),
    MX("mail", 10, "mailserver"),
    MX("mail", 20, "mailqueue"),
    TXT("the", "message"),
    NS("department2", "ns1.dnsexample.com."), // use different nameservers
    NS("department2", "ns2.dnsexample.com."), // for department2.example.com
END);
```

➔ Set default records modifiers

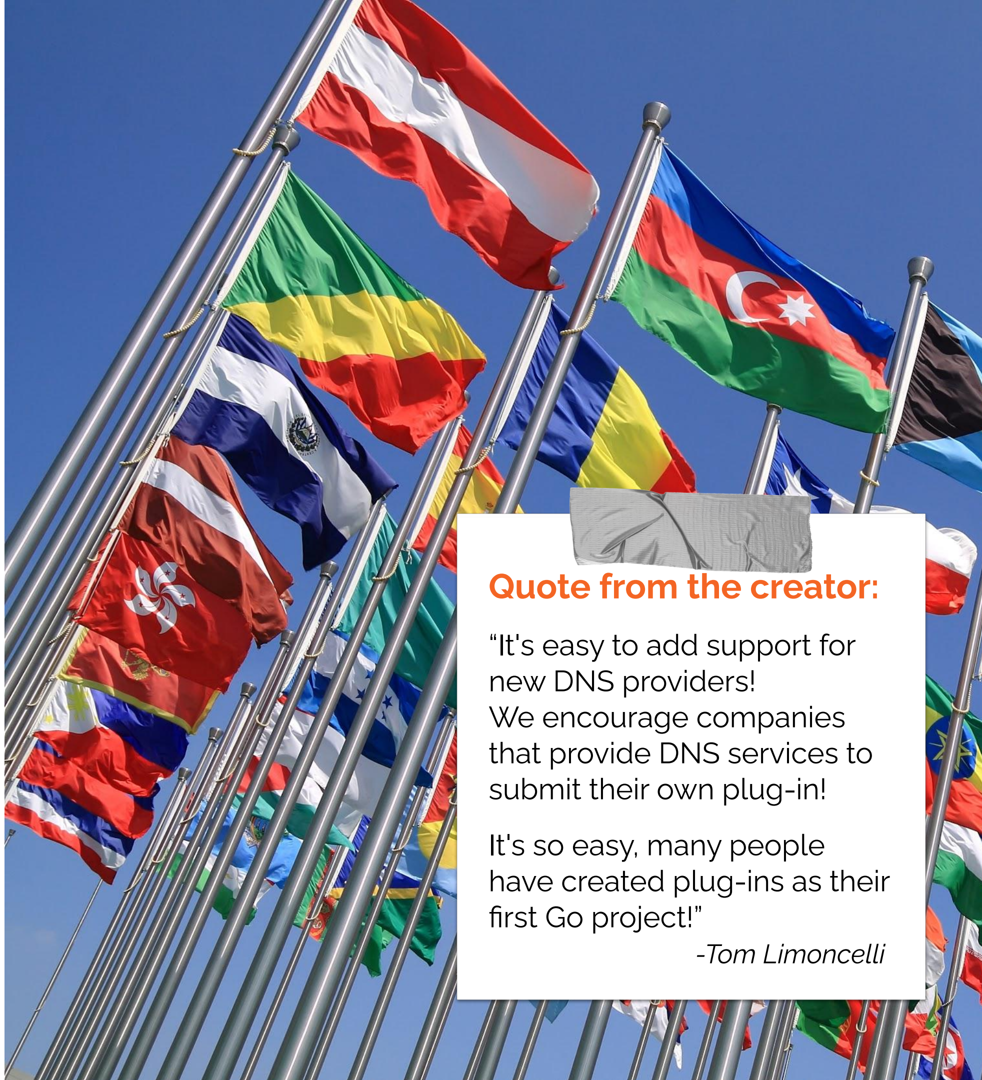
```
DEFAULTS(
    NAMESERVER_TTL("24h"),
    DefaultTTL("12h"),
    CF_PROXY_DEFAULT_OFF,
END);
```

Multi-Provider Support

DNSControl supports a wide range of popular DNS providers, allowing you to manage everything from a single platform.
Here are some examples:

- Cloudflare
- AWS Route 53
- Google Cloud DNS
- DigitalOcean
- Linode
- ...and many more!

Example: With DNSControl, you can manage failover between primary and secondary providers for increased redundancy.



Quote from the creator:

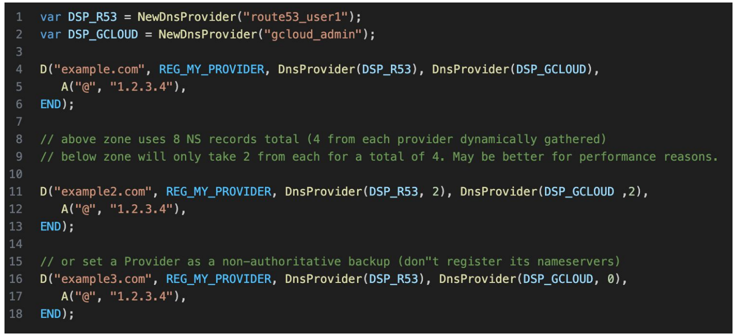
"It's easy to add support for new DNS providers!
We encourage companies that provide DNS services to submit their own plug-in!

It's so easy, many people have created plug-ins as their first Go project!"

-Tom Limoncelli

Example

➔ **Dual DNS Providers**


```
var DSP_R53 = NewDnsProvider("route53_user1");
var DSP_GCLOUD = NewDnsProvider("gcloud_admin");

D("example.com", REG_MY_PROVIDER, DnsProvider(DSP_R53), DnsProvider(DSP_GCLOUD),
   A("@", "1.2.3.4"),
END);

// above zone uses 8 NS records total (4 from each provider dynamically gathered)
// below zone will only take 2 from each for a total of 4. May be better for performance reasons.

D("example2.com", REG_MY_PROVIDER, DnsProvider(DSP_R53, 2), DnsProvider(DSP_GCLOUD ,2),
   A("@", "1.2.3.4"),
END);

// or set a Provider as a non-authoritative backup (don"t register its nameservers)
D("example3.com", REG_MY_PROVIDER, DnsProvider(DSP_R53), DnsProvider(DSP_GCLOUD, 0),
   A("@", "1.2.3.4"),
END);
```

This configuration demonstrates DNSControl's flexibility in managing multi-provider setups and fine-tuning NS record distribution for performance or redundancy.

Real-World Use Case: Large ISP

Before DNSControl

1. Manual updates across different provider portals
2. Inconsistent configurations and errors
3. Time-consuming change management

After DNSControl

1. Centralized management and unified configuration
2. Automated deployments with version control
3. Improved accuracy and consistency across all domains

Implementation Steps

1. Install DNSControl or use Docker!
2. Set up provider credentials
3. Create your initial DNS configuration in DSL
4. Test and validate your configuration
5. Integrate with your preferred version control system (e.g., Git)
6. Set up a CI/CD pipeline for automated deployments

Best Practices

- Use meaningful commit messages for version control
- Implement peer review for DNS changes to catch potential issues
- Regularly audit and clean up unused DNS records
- Leverage DNScontrol's preview feature to avoid accidental changes
- Maintain separate configurations for different environments (e.g., production, staging)

Conclusion

Version-controlled DNS management for simplified tracking and rollbacks

Multi-provider support for centralized management

Improved reliability and consistency in your DNS

Benefits of DNSControl

DNSControl, Born from Tom Limoncelli's vision, powered by our community.

Join us at dnscontrol.org

Your contribution could be the next game-changer in DNS management!

Quote from the creator:

Volunteers needed! If you'd like to get involved, we make it easy to join!

-Tom Limoncelli

Questions?
Comments?
DNS dad jokes??
Git Conflicts???