How we un-scattered our DNS setup and unlocked new automation options

eGYM

Dan Lüdtke
Technical Lead SRE @ eGym GmbH
● Make the gym work for everyone!
● Digital strength machines
● "Fitness Cloud"
  ○ Unify training data across vendors
● Data Analysis
● Apps
● Research Projects
  ○ Improve Diabetes patients symptoms through special training program
A year ago...
Issues

- Ran into maximum *Managed Zone* limit on Google Cloud DNS
- Horrible lookups!
  - Slowing down customers
  - Hard to debug
- Deployment Strategy 😁 #YOLO
  - "Haunted Graveyard"
    - Only few were allowed to touch DNS
    - Even fewer dared to touch DNS
Lessons Learned

Organizational structure and infrastructure evolve differently.

Don't force one onto the other.

Use company-wide unique artifact names in DNS.
Let's Improve!
What is the Problem here?

- One does not simply change DNS
- How to rollback?
- Web interface does not provide atomicity!
- Agility!
  - We build it, we run it!
- SRE is too slow changing DNS

SREs  Devs
Divide and Conquer DNS Data

- **Volatile**
  - Special test domain
  - No availability guarantees
  - Everyone can change directly
  - No reviews
  - No tests
  - No atomicity (no changesets)

- **Production**
  - Version control
  - Reviewed changes
  - Tested for common mistakes
  - Tested for syntax, logic, deployment feasibility
  - Atomic deployment of whole changeset

---

Agility

Reliability
Do we really have competing goals?

**SREs**

We need reviewed, version-controlled changes in production.

**Devs**

We need rapid change during development.
Storing DNS Data
Zone Data

- Version Control
  - Git repository
  - All developers have access
- YAML-based format
  - Developer love it
    - compared to zone files ;)
  - Easy to read and understand
- Templating functionality

```yaml
zones:
  - zone: egym.coffee
description: Test zone.
ttl: 300
templates:
  - gmail
  - website
names:
  - name: '@'
texts:
    data:
    - foobar-site-verification-123456
    - name: paloalto
forwarding:
ttl: 60
target: flaky.cloud.example.com.
  - name: losangeles
addresses:
literals:
  - 192.0.2.99
  - 2001:db8:200::99
```
Zone Data (Template)

- Tradeoff between
  - Principle of Least Surprise
  - Don't Repeat Yourself (DRY)

- Typical templates
  - Set of mail servers
  - Set of name servers (delegation)
  - Domain Parking
  - Redirect to commercial website

```yaml
templates:
- template: gmail
description: >
  This template adds Google mail servers to a zone.
  mail servers:
      priority: 10
      priority: 20
  names:
    - name: '@'
      mail:
        ttl: 604800
      mailservers:
      priority: 10
      priority: 20
  texts:
    google._domainkey
data:
  - >
    v=DKIM1;
    k=rsa;
    p=foobar123456
```

Validating DNS Data
Resource Record Database (RRDB)

- Go package
- Limited dependencies
  - Go Standard Library
  - YAMLv2
- High test coverage
- Unfortunately: Battle-tested 🙈

<table>
<thead>
<tr>
<th>Files</th>
<th>≡</th>
<th>▶</th>
<th>▨</th>
<th>▣</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>loader.go</td>
<td>115</td>
<td>111</td>
<td>2</td>
<td>2</td>
<td>96.52%</td>
</tr>
<tr>
<td>rrdb.go</td>
<td>332</td>
<td>332</td>
<td>0</td>
<td>0</td>
<td>100.00%</td>
</tr>
<tr>
<td>Folder Totals (2 files)</td>
<td>447</td>
<td>443</td>
<td>2</td>
<td>2</td>
<td>99.10%</td>
</tr>
<tr>
<td>Project Totals (7 files)</td>
<td>729</td>
<td>658</td>
<td>6</td>
<td>65</td>
<td>90.26%</td>
</tr>
</tbody>
</table>
RRDB Internals: Trie Data Structure

Root node

- com
- de
- it
- pl

com

- egym

egym

- my-service
  - A
  - AAAA
  - MX
  - TXT

my-service
RRDB Internals: Today's Features

- Logic checks within nodes
  - E.g. CNAME and most other record types are mutually exclusive
- Back-and-forth traversal
  - Parent pointers
- Logic checks across nodes
  - E.g. Node with NS records should not have children
- Walk and query the Trie
- Idea: Inheritance of certain values (e.g. TTL)
RRDB Internals: Past Disasters

What we believed to be serving:
- com
- egym
- foobar

What we actually served:
- com
- egym
- foobar

E N D   O F   L I F E

old DNS server
New Process
New Deployment Workflow

Push Commit

Files changed (1)

- zonedata/com.egym.ts.co.yml

Templates:
  - gmail
Names:
  - name: authz
  - addresses:
    35.195.2.164
  - name: bpm-onboarding
  forwarding:
New Deployment Workflow

- Push Commit
- YAML Lint

Pipeline #59

- Successful
- 1 min 29 sec • 14 days ago
- Push by Stephan

Commit
- c7a1d54

Pipeline
- View configuration
- check syntax
- 21s

Steps:

- Build setup
- apt-get update
- apt-get install -y yamllint
  - yamllint zonedata/*.yml
  - yamllint zonedata/*.yml
- Build teardown
New Deployment Workflow

- Push Commit
- YAML Lint
- RRDB Logic Checks

Pipeline

- check syntax: 21s
- check logic: 14s
- deploy to staging: 53s

dbcheck
2018/07/20 13:42:22 Looks good!
New Deployment Workflow

- Push Commit
- YAML Lint
- RRDB Logic Checks
- Deploy to DNS Staging
New Deployment Workflow

Push Commit -> YAML Lint -> RRDB Logic Checks -> Deploy to DNS Staging -> Review

Deletetes DNS entry for authz

#20 MERGED at c7ald54 authz-deletion → master

Overview  Commits  Activity

Author  Stephan

Reviewers

Description  It is not in use for some time, who needs authorization?
  • Searched for the IP in our infrastructure to be sure
New Deployment Workflow

- Push
- Commit
- YAML Lint
- RRDB Logic Checks
- Deploy to DNS Staging
- Review
- Deploy to DNS Production

Deployment on Production DNS
- egym.com. 2018/08/03 17:17:37 1 records to be deleted
- egym.com. 2018/08/03 17:17:37 *35.195.2.164
- egym.com. 2018/08/03 17:17:37 skipping action! (dry run)
- egym.directory. 2018/08/03 17:17:24 nothing to change
- egym.bg. 2018/08/03 17:17:24 nothing to change
- egym.training. 2018/08/03 17:17:24 nothing to change
- egym.ba. 2018/08/03 17:17:24 nothing to change
Benefits of New Process

- DNS workflow and moving parts are out-of-band
  - Code and Pipeline on Bitbucket
  - Independent from the records we serve
- Pipeline run takes ~1.5 minutes
  - Before: review took hours or days
  - Including all checks
  - Including full staging deployment
Lessons Learned

Automated checks lower the entry barrier and empower developers.

Democratize critical infrastructure!

De-haunt the graveyards!
Battle-tested Existing Tools

- **Record Store (Shopify)**
  - No Cloud DNS support (added Jan '18)
  - We were just moving away from Ruby within SRE
- **OctoDNS (Github)**
  - No Cloud DNS support (added Oct '17)
- **Denominator (Netflix)**
  - No Cloud DNS support
- **DNSControl (Stack Exchange)**
  - Go
  - Uses Domain Specific Language
  - We did not know about it 😞
Lesson Learned

We may have fallen for Not-Invented-Here...?

Do proper research!
Use our tools if all of the following apply

- You love YAML
- You need a Go library (RRDB)
- Google Cloud DNS is your only DNS provider
- You need to walk & query the final dataset
  - Custom checks
  - Service Discovery
  - Special Needs
- Prefer a small binary
  - that fits into out-of-band pipelines
Achievements Unlocked

- **DNS is finally out-of-band**
- **DNS is not scary anymore!**
  - Spreads the review load from SRE to everyone
- **Certificate Automation in Kubernetes**
  - Cluster Issuer uses DNS-01 challenge
    - works for client certificate protected hostnames
  - Developers can request valid Let's Encrypt certificates via Certificate Resource
    - even before DNS is pointed to the corresponding Ingress Resource
- **Configuration-less Delegation Monitoring**
  - Automatically monitors all domains that appear on Cloud DNS
  - Alert on domain take-over
  - Alert on delegation errors
Open Source *dns-tools* and *RRDB*

- [https://bitbucket.org/egym-com/dns-tools/](https://bitbucket.org/egym-com/dns-tools/)

Full story of our DNS Journey in our tech blog!

- [https://code.egym.de/](https://code.egym.de/)

Fitness and engineering careers: [egym.com](http://egym.com)

Mostly non-political, tech-related, (re-)tweets: @danrl_com

I blog about SRE and technology: [https://danrl.com](https://danrl.com)