# Extending Catalog zones

## another approach in automating maintenance

Leo Vandewoestijne
dns.company

Rev 0.000

# Spoilers/Warnings

- Not much new; working in Bind already
- This presentation was prepared while having lack of time

# Introduction

Currently you may use python/jinja/yaml to generate includes for Bind, NSD, Knot, Yadifa, etc.
Which then need to be provisioned, and then loaded.

Catalog zones are DNS zones containing dynamic configuration, or better said “configuration data”:
the domain zones to be loaded/unloaded) in your DNS daemon.

In this presentation the main configuration is not addressed (see ISC examples), but focussed on the catalog zone itself.

# old code

```
$ORIGIN catzone.
$TTL 14400
@                                       IN SOA    . . 1552507036 86400 14400 86400 14400
@                                          NS     invalid.
@                                          ZONEMD 1552507036 1 0 ( 4ac63blablablaetc )
version                                 0 IN TXT    "1"
$ORIGIN zones.catzone.
2994957a552f357f9e49007e0462d3617354e9df  PTR    alila.dog.
3d5f1f25bf803861058ec11a929e38c134cbc1f6  PTR    trouwauto.limo.
2994957a552f357f9e49007e0462d3617354e9df  PTR    trouwauto.wedding.
038aa43b8bf77211435f57ad25bcc1e6ba7b0e05  PTR    unicycle.show.
43f984676423c0a9c30c8c635b3757dd2f820dff  PTR    malabarista.coffee.
019559dd498f0b289662322fceb8f0c431d92193  PTR    globeofdeath.com.
4498690ad458ebc22eeffb86ba2b7a1f181c3942  PTR    iommametaetu.pizza.
```

NOTE: to generate the hash used in ISC's examples:

```
print hashlib.sha1(dns.name.from_text("example.tld").to_wire()).hexdigest()
# or
printf '\7example\3tld\0' | openssl sha1
```

# extended code

```
$ORIGIN catzone.
$TTL 14400
@                                       IN SOA    . . 1552507036 86400 14400 86400 14400
@                                          NS     invalid.
@                                          ZONEMD 1552507036 1 0 ( 4ac63blablablaetc )
version                                 0 IN TXT    "1"
$ORIGIN zones.catzone.
2994957a552f357f9e49007e0462d3617354e9df  PTR    alila.dog.
3d5f1f25bf803861058ec11a929e38c134cbc1f6  PTR    trouwauto.limo.
2994957a552f357f9e49007e0462d3617354e9df  PTR    trouwauto.wedding.
038aa43b8bf77211435f57ad25bcc1e6ba7b0e05  PTR    unicycle.show.
43f984676423c0a9c30c8c635b3757dd2f820dff  PTR    malabarista.coffee.
019559dd498f0b289662322fceb8f0c431d92193  PTR    globeofdeath.com.
4498690ad458ebc22eeffb86ba2b7a1f181c3942  PTR    iommametaetu.pizza.

;; instead of having it configured you can put options in the zone also:

allow-transfer    0 IN APL    ( 1:192.168.1.2/32, 1:192.168.1.3/32, 2:2a03:96a2::95de/128 )
masters           0 IN A      192.168.1.111
masters           0 IN A      192.168.1.222
masters           0 IN AAAA   2a03:96a2::111
masters           0 IN AAAA   2a03:96a2::222
notifies          0 IN A      192.168.1.1
notifies          0 IN A      192.168.1.2
notifies          0 IN A      192.168.1.3
notifies          0 IN AAAA   2a03:96a2::95de
```

NOTE: RFC 3123 about the APL RR type allows multiple values in one RR
The catalog zone RFC-draft recommends multiple APL's in stead of combined values.

# new code

```
$ORIGIN catzone.
$TTL 14400
@                                       IN SOA    . . 1552507036 86400 14400 86400 14400
@                                          NS     invalid.
@                                          ZONEMD 1552507036 1 0 ( 4ac63blablablaetc )
version                                 0 IN TXT    "2"
$ORIGIN zones.catzone.
17187                                      PTR    alila.dog.
22820                                      PTR    trouwauto.limo.
38734                                      PTR    trouwauto.wedding.
47883                                      PTR    unicycle.show.
57822                                      PTR    malabarista.coffee.
65210                                      PTR    globeofdeath.com.
77404                                      PTR    iomammetaetu.pizza.
$ORIGIN x-info.catzone.
                                           ZONEMD 1552507036 240 0 ( 8gelablablablaetc )
0                                          HINFO  "0000000000" 17187
17187                                      HINFO  "1552202321" 22820
22820                                      HINFO  "1552322222" 38734
38734                                      HINFO  "1552232323" 47883
47883                                      HINFO  "1552332324" 57822
57822                                      HINFO  "1552242525" 65210
65210                                      HINFO  "1552452726" 77404
77404                                      HINFO  "1555262027" EoF
```

NOTES:

The catalog zone RFC-draft doesn't demand the hashes in the PTR's

Data integrety can be enforced with the ZONEMD record, absoleting the need for inefficient TSIG's
(which remain recommended for the authentication function)

ZONEMD got private use flag, which could be used to hash parts of zone data

The HINFO RR got "un-absoleted" by RFC 8482

# Benefits

- **No need to reconfigure**
  at each add or delete of a zone; adding or deleting a domain could be done simply with a DDNS call
  (if you were not on a SQL backend already)

- **No vendor specific commands nor maintenance ports/sockets**
  depending on what you prefer to be running at that moment.
  Like rndc, knotc, nsd-control, pdns util.
  Then yadifa CTRL command are very interesting; daemon maintainance using DNS queries...
  ...but don't have a ZONEADD or ZONEDEL command (yet).

- **No addition transport**
  scp, rsync or other needed to provision dynamic configuration / includes, and so:
- **No restricted user needed**
  to do synchronize or run such maintenance commands.

- **Have a zone for each customer**
  to include their zones and mutations, instead of assembling your config from multiple sources

- **No need to run / replicate / monitor / maintain backends at secondaries**
  like MySQL or Oracle

**...IF all vendors would adopt the to-be-made standard OR having a daemon that would translate**

# Abuse of RR types

**Initially not H**INFO **but NXT record**
Which would be appropriate ...but officially obsolete.

**Then the N**INFO **RR - AKA ZS record; “Zone Status”**
Has no RFC, only had a draft (by Jim Reid).
However is assigned by IANA.
But despite:
the **N**INFO record is missing already in dnspython, Knot and NSD.

**Then Cloudflare came with RFC 8482**
Officially un-abosoleting the **H**INFO record.

# Wishlist

Many masters are running on the non-default port.
I think it would make sense to have the "masters" better use the SRV RR.

```
_dns._tcp.masters       SRV     0 0 5300 masters
```

The same counts for notifies - for example when you have dnsdist in front of your daemon.

```
_dns._tcp.notifies      SRV     0 0 5300 masters
```

But the above examples do not address each primary specifically. So I guess this will be better something like:

```
1.masters             0 IN A      192.168.1.111
2.masters             0 IN A      192.168.1.222
3.masters             0 IN AAAA   2a03:96a2::111
4.masters             0 IN AAAA   2a03:96a2::222
_dns._tcp.masters  0 IN SRV    2 10 5300 1.masters
_dns._tcp.masters  0 IN SRV    2 20 5300 2.masters
_dns._tcp.masters  0 IN SRV    1 10 5353 3.masters
_dns._tcp.masters  0 IN SRV    1 20 5353 4.masters
```

Plus -when having Knot as primary- you may even wish to have only TCP workers, as you're only doing AXFR and IXFR

(but does "refresh" always do TCP fallback?)

# Future

“Probably” need work to make this a standard.

And “lobbying” to get it adopted by vendors.

...who may kindly decline.

So for those we need a maintenance daemon to translate commands.

Will start with putting code examples at:

**https://dns.company/catzone**

# References


**RFC Draft catalog zones** (IETF)
https://tools.ietf.org/html/draft-muks-dnsop-dns-catalog-zones-04

**A short introduction to Catalog Zones** (ISC)
https://kb.isc.org/docs/aa-01401

**Catalog zones are coming to Bind** (Jan-Piet Mens)
https://jpmens.net/2016/05/24/catalog-zones-are-coming-to-bind-9-11/

**DNS Catalog Zones** (Witold Kręcicki, Polish - just read code)
https://www.isc.org/docs/plnog16-catzones.pdf

**RFC 3123 APL record** (IETF)
https://tools.ietf.org/html/rfc3123

**RFC Draft ZONEMD record** (IETF)
https://tools.ietf.org/id/draft-ietf-dnsop-dns-zone-digest-01.html#rfc.section.2.1.3

**RFC 8482 “un-absolete the HINFO RR”** (IETF)
https://tools.ietf.org/html/rfc8482

# Questions / feedback

leo@dns.company