
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2019

Intro + Deep Dive: CoreDNS

Yong Tang

CoreDNS

CLOUD NATIVE
COMPUTING FOUNDATION

Speaker

- Yong Tang
  - GitHub: yongtang
  - Maintainer: CoreDNS and Docker
  - SIG IO Lead & Maintainer: TensorFlow
  - Director of Engineering, MobileIron

CoreDNS

CoreDNS

Agenda

- Introduction
- Status Update
- Google Summer of Code
- Technical Deep Dive
- Demo Plugin: Source Based Service Discovery
- Q & A

CoreDNS

- Flexible DNS server written in Go
- Focus on service discovery
- Plugin based architecture, easily extended
- Default DNS server in Kubernetes
- Supports DNS, DNS over TLS, DNS over gRPC
- Support Route53 DNS sync up
- Started and led by Miek Gieben

CoreDNS

CoreDNS Community

- 153 Contributors (Big Thanks!)
- 14 Maintainers
- 30 Public Adopters
- 4200+ Stars

CoreDNS Community

- Most active:
  - GitHub: https://github.com/coredns/coredns
  - Slack: #coredns on https://slack.cncf.io
- More resources:
  - Web: https://coredns.io
  - Blog: https://blog.coredns.io
  - Twitter: @corednsio
  - Mailing list/group (not very active):
    - coredns-discuss@googlegroups.com

Google Summer of Code

- Participated 3 years in a row (2017-2019)
- 2017: Varyoo (GitHub: varyoo)
- 2018: Jiacheng Xu (GitHub: jiachengxu)
- 2019: Proposal accepted and in progress

CoreDNS

Technical Deep Dive

- Service Discovery with DNS
- Updates for Kubernetes Users
- How Queries Are Resolved
- Building Your Own Plugin

Service Discovery with DNS

- DNS is a nice and flexible indirection
- DNS is easy and simple, for Dev/DevOps/IT
- DNS has been there for a long time & part of the existing IT infrastructure
- Works with hybrid environments (in & out of k8s cluster)
- DNS is distributed in nature, scales really well

CoreDNS

Updates for Kubernetes

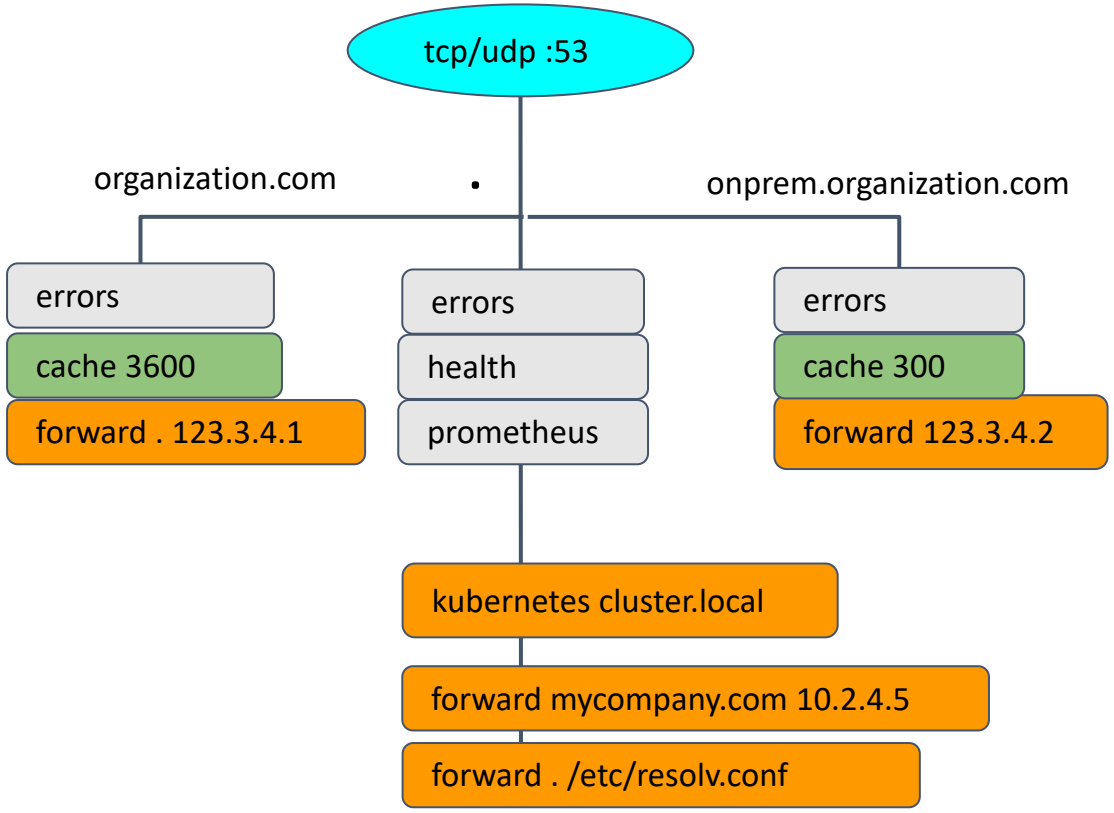
- Kubernetes 1.11 - 1.14 use CoreDNS 1.2.6
- Key, relevant changes from 1.2.6 to 1.5.0
  - k8s_external plugin proxy -> forward
  - health -> health & ready
  - grpc
  - kubernetes TTLs now apply to negative responses (e.g., NXDOMAIN)
  - upstream option no longer needed
  - multiple API server endpoints no longer supported
  - API server resync disabled

Updates for Kubernetes

- **BACKWARDS INCOMPATIBLE CHANGES**
  - These are why 1.5.0 is not yet used by default in Kubernetes - different deprecation policies
  - Corefile migration tool in progress
  - proxy -> forward
  - health -> health & ready
  - upstream option changes
  - multiple API endpoints no longer supported

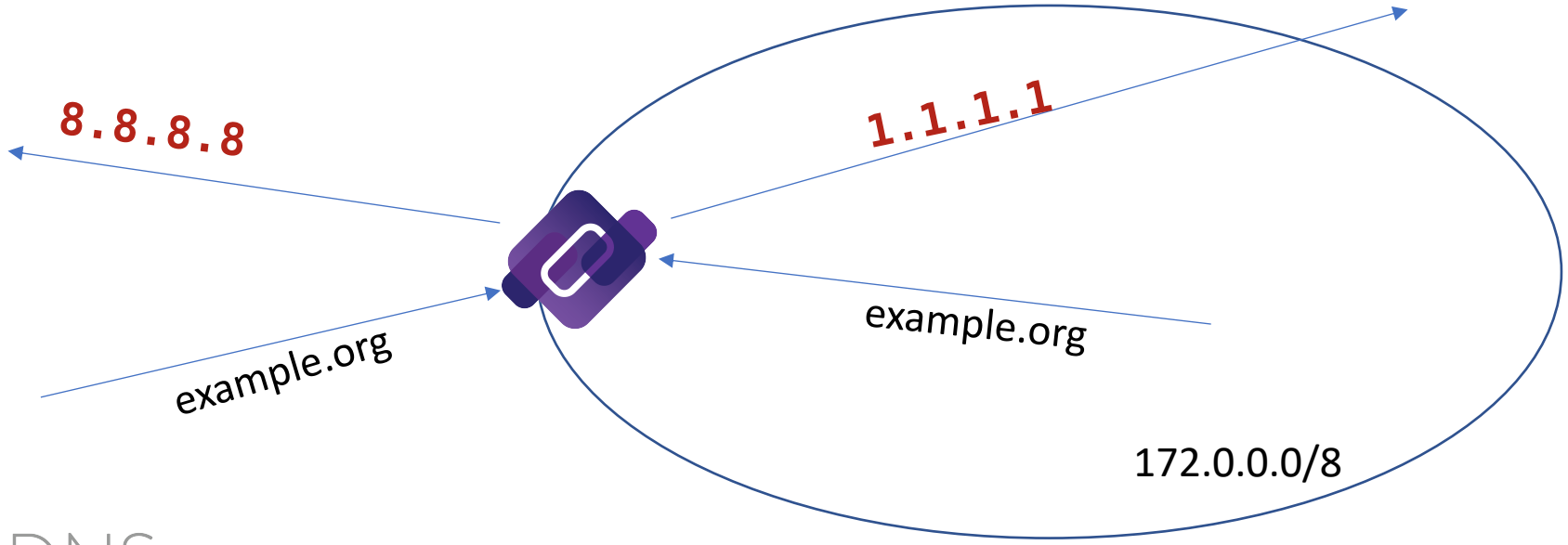
Resolving a Query


```
organization.com:53 {
    errors
    cache 3600
    forward . 123.3.4.1
}
onprem.organization.com:53 {
    errors
    cache 300
    forward . 123.3.4.2
}
.:53 {
    errors
    health
    kubernetes cluster.local ... {
        ...
    }
    forward mycompany.com 10.2.4.5
    forward . /etc/resolv.com
}
```



Demo Plugin

- Source IP based service discovery
  - Returns `"1.1.1.1"` for 172.0.0.0/8 or 127.0.0.0/8
  - Returns `"8.8.8.8"` otherwise



Demo Plugin: Setup


```
func init() {
        caddy.RegisterPlugin("demo", caddy.Plugin{
                ServerType: "dns",
                Action:     setup,
        })
}
func setup(c *caddy.Controller) error {
        c.Next() // 'demo'
        if c.NextArg() {
                return plugin.Error("demo", c.ArgErr())
        }
        dnsserver.GetConfig(c).AddPlugin(func(next plugin.Handler) plugin.Handler {
                return Demo{}
        })
        return nil
}
```

CoreDNS

Demo Plugin: ServDNS (1)


```
// ServeDNS implements the plugin.Handler.ServeDNS.
func (p Demo) ServeDNS(ctx context.Context, w dns.ResponseWriter, r *dns.Msg) (int, error) {
        state := request.Request{W: w, Req: r}
        qname := state.Name()
        reply := ”8.8.8.8"
        if strings.HasPrefix(state.IP(), "172.") || strings.HasPrefix(state.IP(), "127.") {
                reply = ”1.1.1.1"
        }
        fmt.Printf("Received query %s from %s, expected to reply %s\n", qname, state.IP(), reply)
        ...
}
```

CoreDNS

Demo Plugin: ServDNS (2)


```
func (p Demo) ServeDNS(ctx context.Context, w dns.ResponseWriter, r *dns.Msg) (int, error) {
        ...
        rr := new(dns.A)
        rr.Hdr = dns.RR_Header{Name: qname, Rrtype: dns.TypeA, Class: dns.ClassINET}
        rr.A = net.ParseIP(reply).To4()
        answers = append(answers, rr)
        m := new(dns.Msg)
        m.SetReply(r)
        m.Authoritative = true
        m.Answer = answers

        w.WriteMsg(m)
        return dns.RcodeSuccess, nil
}
```

CoreDNS

Demo Plugin: Corefile


```
.:1053 {
    # By default all plugins
    # are disabled initially,
    # unless enabled explicitly
    #
    demo
}
```

Demo Plugin: Build


```
$
$ # add demo:demo to plugin.cfg
$
$ # build with docker (golang:1.12)
$ docker run --rm -i -t -v \
$     $PWD:/v -w /v golang:1.12 sh -c 'make gen && make'
$
$ # configure Corefile and run coredns
$
$ ./coredns
$
```

Contribution Welcome

- Star CoreDNS in GitHub:
  - https://github.com/coredns/coredns
- Add the name to **ADOPTERS.md**
- Participate in GitHub or Slack discussions
  - Slack: #coredns on https://slack.cncf.io
- Create a PR to become a contributor
- Become a maintainer
  - One significant pull request
  - Sponsored by one current maintainer

THANK YOU

CoreDNS