CoreDNS Deep Dive

Yong Tang
MobileIron, Inc.
Maintainer, CoreDNS
GitHub: yongtang
Agenda

- Introduction
- Service discovery, Corefile, and plugins
- Plugin demo: source based service discovery
CoreDNS:

- Flexible DNS server written in Go
- Focus on service discovery
- Plugin based architecture, easily extended
- Supports DNS, DNS over TLS, DNS over gRPC
- Started and led by Miek Gieben
CoreDNS:

- Incubating project in CNCF
  - Inception in 2017, incubating in 2018
  - Plan on graduation now

- Kubernetes support
  - GA in Kubernetes 1.11
  - Expected default in Kubernetes 1.13

- Growing community
  - 112 contributors (Big Thanks!)
  - 16 maintainers
  - 29+ public adopters
  - 2800+ 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
CoreDNS: Plugin Contributions

• External plugin:
  • Maintained externally
  • Link through https://coredns.io

• Plugin as a CoreDNS subproject
  • Maintained in a separate repo in CoreDNS organization
  • Separate issue and PR tracking
  • Sponsored by one maintainer

• Plugin in CoreDNS project
  • More selective in quality
  • Same issue and PR tracking in CoreDNS repo (OWNERS)
  • Sponsored by one maintainer
CoreDNS: Service Discovery

- DNS is a nice indirection
- Maximum flexibility
- Easy and simple to change DNS
- Pervasive in IT infrastructure
- Distributed in nature
Corefile: CoreDNS Configurations

```conf
:.53 {
  # By default all plugins are disabled initially, unless enabled explicitly
  kubernetes cluster.local 10.96.0.0/12 {< k8s integration
      pods insecure
  }
  route53 example.com::Z1Z2Z3Z4Z5Z6Z7 <- route53 aws cloud data sync up
  hosts example.hosts example.org {
    192.0.0.100 www.example.org
  }
  health <- healthcheck
  prometheus <- metrics
  cache 30 <- cache & performance
  forward . 1.1.1.1:53 <- forward to 1.1.1.1 (Cloudflare)
  errors
}
```
CoreDNS for Service Discovery

```bash
# /etc/hosts like inline records
hosts example.hosts example.org {
  192.0.0.100 www.example.org
}
```

CoreDNS 1.1.1.1

Cloudflare
CoreDNS for Service Discovery

# out-of-cluster k8s deploy
kubernetes cluster.local {
}

# /ets-hosts like inline records
hosts example.hosts example.org {
  192.0.0.100 www.example.org
}

1.1.1.1
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:

- // 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)
  - ...
  - }
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
}
Demo Plugin: Setup

```go
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
}
```
 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:1.11)
• docker run --rm -i -t -v \
• $PWD:/go/src/github.com/coredns/coredns \ 
• $ -w /go/src/github.com/coredns/coredns \ 
• $ golang:1.11 make
•
•
• # configure Corefile and run coredns
•
• $ ./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

CoreDNS