A Deep Dive into  CoreDNS

DavadDi dwh0403@163.com

Agenda

1. Introduction

2. Architecture

3. Plugins & External Plugins

4. Service Discovery with K8S

1. Introduction

“CoreDNS: DNS and Service Discovery”

- OpenSource Github Go
- Apache License Version 2
- Plugins
- Simplicity
- Service Discovery etcd && k8s
- Fast and Flexible compile only need plugins
- Semantic Versioning **MAJOR.MINOR.PATH**

CLOUD NATIVE COMPUTING FOUNDATION

k8s 1.11 CoreDNS GA

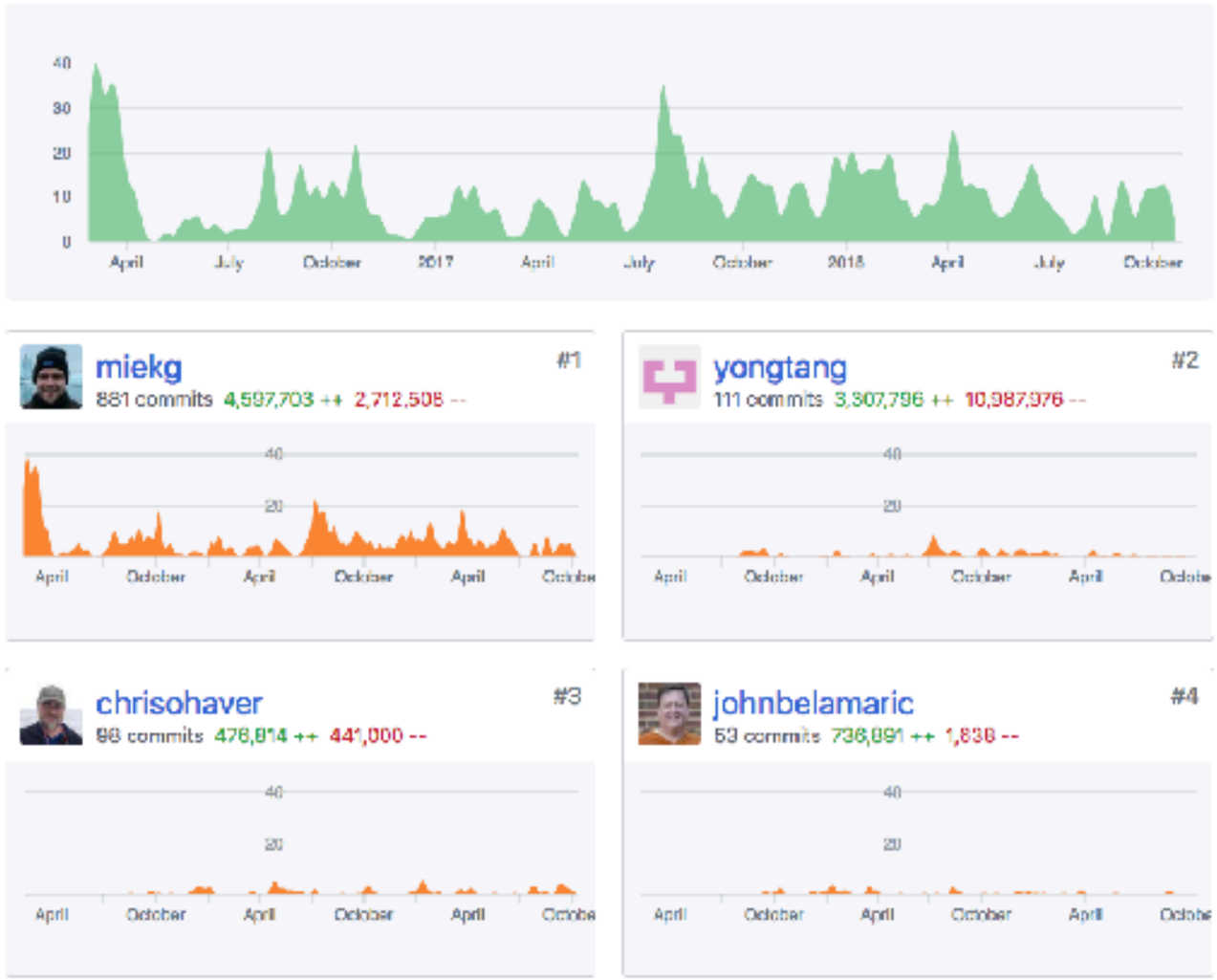
1. Introduction

Miek Gieben, The author of SkyDNS2, CoreDNS. SRE at Google

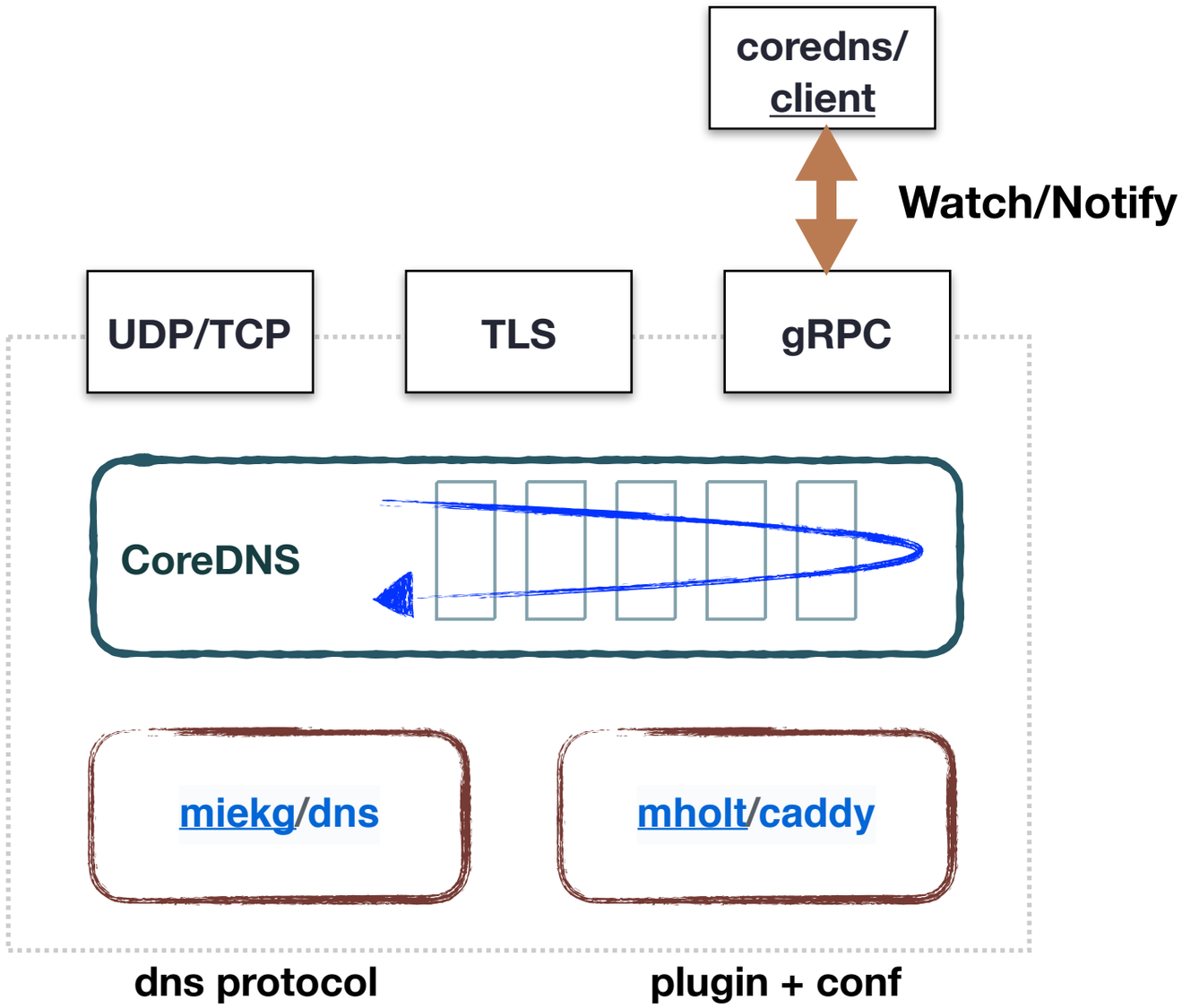
Star: **2877**
Fork: **436**
Contributors:**112**

Commits: **1494**
Releases: **32**

PR: **13/1365**
Issues: **65/ 842**



2. Architecture

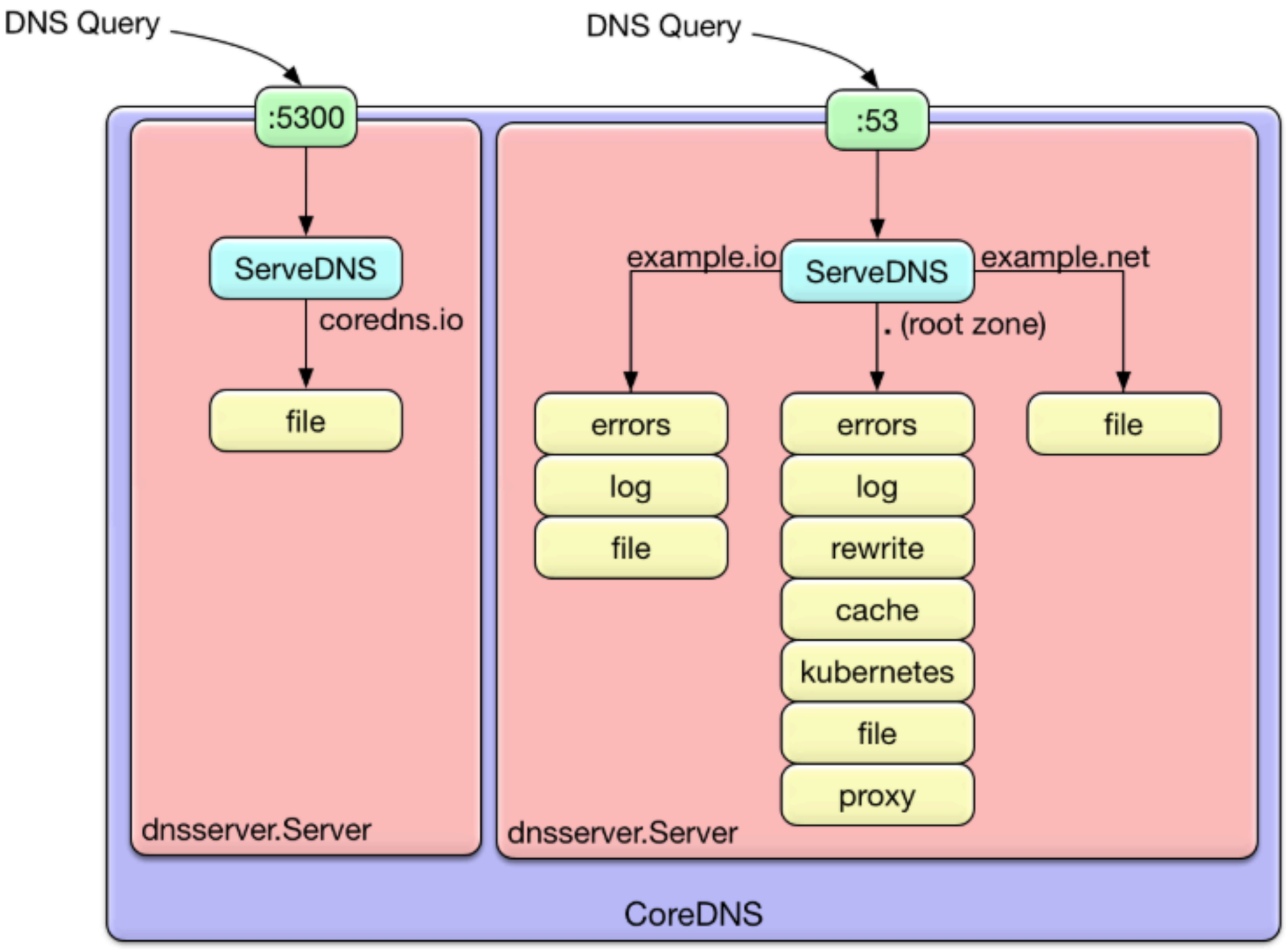


Simple + Powerful

Corefile Example

```
coredns.io:5300 {
  file /etc/coredns/zones/coredns.io.db
}

example.io:53 {
  errors
  log
  file /etc/coredns/zones/example.io.db
}

example.net:53 {
  file /etc/coredns/zones/example.net.db
}

.:53 {
  errors
  log
  health   # http://localhost:8080/health
  rewrite name foo.example.com foo.default.svc.cluster.local

  # –service-dns-domain && --service-cidr
  kubernetes cluster.local 10.0.0.0/24 # PTR
  file /etc/coredns/example.db example.org
  proxy . /etc/resolv.conf
  cache 30
}
```

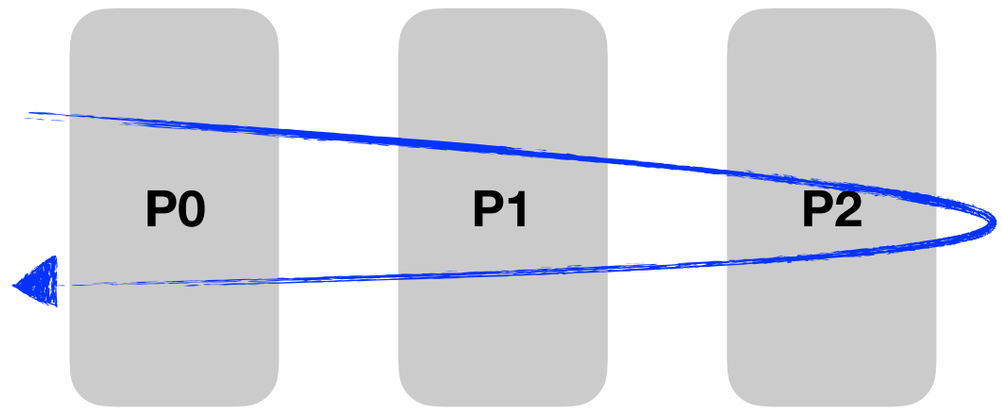
```
ZONE:[PORT] {
      [PLUGIN] ...
}
```


DNS Query
DNS Query
:5300
:53
ServeDNS
coredns.io
file
example.io
ServeDNS
example.net
. (root zone)
errors
log
file
errors
log
rewrite
cache
kubernetes
file
proxy
file
dnsserver.Server
dnsserver.Server
CoreDNS

Implement plugins in Go ?

```go
type ResponseWriter interface {
	Write()
}

type P0Writer struct {
	ResponseWriter
}

func (w *P0Writer) Write() {
	fmt.Println("In BasicWrite")
	if w.ResponseWriter != nil {
		w.ResponseWriter.Write()
	}
}

type P1Writer struct {
	ResponseWriter
}

func (w *P1Writer) Write() {
	fmt.Println("In H1Writer")
	if w.ResponseWriter != nil {
		w.ResponseWriter.Write()
	}
}

type P2Writer struct {
	ResponseWriter
}

func (w *P2Writer) Write() {
	fmt.Println("In H2Writer")
	if w.ResponseWriter != nil {
		w.ResponseWriter.Write()
	}
}

func main() {
	var p0 ResponseWriter
	p0 = &P0Writer{ResponseWriter: nil}

	p1 := &P1Writer{ResponseWriter: p0}
	p2 := &P1Writer{ResponseWriter: p1}

	p2.Write()
}
```

Iris, Gin …



interfacer Write()

3. Plugins & External Plugins

Name	Desc
auto	enables serving zone data from an RFC 1035-style master file, which is automatically picked up from disk.
autopath	allows for server-side search path completion. **autopath [ZONE...] RESOLV-CONF**
bind	overrides the host to which the server should bind.
cache	enables a frontend cache. **cache [TTL] [ZONES...]**
chaos	allows for responding to TXT queries in the CH class.
debug	disables the automatic recovery upon a crash so that you'll get a nice stack trace. text2pcap
dnssec	enable on-the-fly DNSSEC signing of served data.
dnstap	enable logging to dnstap. http://dnstap.info golang: go get -u -v github.com/dnstap/golang-dnstap/dnstap
erratic	a plugin useful for testing client behavior.
errors	enable error logging.
etcd	enables reading zone data from an etcd version 3 instance.
federation	enables federated queries to be resolved via the kubernetes plugin.
file	enables serving zone data from an RFC 1035-style master file.
forward	facilitates proxying DNS messages to upstream resolvers.

3. Plugins & External Plugins

Name	Desc
health	enables a health check endpoint.
host	enables serving zone data from a `/etc/hosts` style file.
kubernetes	enables the reading zone data from a Kubernetes cluster.
loadbalance	randomize the order of A, AAAA and MX records.
log	enables query logging to standard output.
loop	detect simple forwarding loops and halt the server.
metadata	enable a meta data collector.
metrics	enables Prometheus metrics.
nsid	adds an identifier of this server to each reply. RFC 5001
pprof	publishes runtime profiling data at endpoints under `/debug/pprof`.
proxy	facilitates both a basic reverse proxy and a robust load balancer.
reload	allows automatic reload of a changed Corefile. Graceful reload
rewrite	performs internal message rewriting. “rewrite name foo.example.com foo.default.svc.cluster.local”
root	simply specifies the root of where to find (zone) files.

3. Plugins & External Plugins

Name	Desc
router53	enables serving zone data from AWS route53.
secondary	enables serving a zone retrieved from a primary server.
template	allows for dynamic responses based on the incoming query.
tls	allows you to configure the server certificates for the TLS and gRPC servers.
trace	enables OpenTracing-based tracing of DNS requests as they go through the plugin chain.
whoami	returns your resolver's local IP address, port and transport.

External Plugins: https://coredns.io/explugins/

Helper Plugin

health

reload

tls

...

Backend Plugin

file

fallthrough

kubernetes

ServiceBackend

SD

etcd

...

Normal Plugin

cahce

metrics

log

errors

...

proxy

forward

...

upstream

```
ServeDNS(ctx context.Context, w dns.ResponseWriter, r *dns.Msg) (int, error)
```

the same and the difference ?

- **auto vs file**

auto: automatically picks up new zones.

file: handle wildcards better.

- **forward vs proxy**

At some point proxy should be deprecated. The google_https protocol is already on the chopping block for 1.1.3.

Functional; the proxy plugin supports more protocols than forward

dnstap

dnstap is a flexible, structured binary log format for DNS software. It uses Protocol Buffers to encode events that occur inside DNS software in an implementation-neutral format.

golang-dnstap

Start plugin in Corefile

dnstap /tmp/dnstap.sock full

$ dnstap -u /tmp/dnstap.sock or
$ dnstap -u /tmp/dnstap.sock -w /tmp/july.dnstap
$ dnstap -r /tmp/july.dnstap -y

plugins enabling or disabling at compile time

- Build With compile-time conf file plugin.cfg

```
...
whoami:whoami
erratic:erratic
startup:github.com/mholt/caddy/startupshutdown
...
```

- Build with external golang source code

```go
var directives = []string{
     "example",
     ...
     "whoami",
}

func init() {
     dnsserver.Directives = directives
}

func main() {
     coremain.Run()
}
```

write own plugins

1. Registration
2. Setup function
3. ServerDNS() and Name()
4. Hooking it up to pulgin.cfg
5. Using it

1. https://github.com/coredns/example
2. https://coredns.io/2017/03/01/how-to-add-plugins-to-coredns/

4. Service Discovery with k8s

```
kubernetes [ZONES...] {
    resyncperiod DURATION               # API Server resync DURATION period
    endpoint URL [URL…]                 # API Server URL
    tls CERT KEY CACERT                 # API Server connection TLS related.
    kubeconfig KUBECONFIG CONTEXT       # use kubeconfig context

    namespaces NAMESPACE…               # exposed ns
    labels EXPRESSION                   # selector labels send to API Server
    pods POD-MODE                       # pods mode: disabled,insecure,verified
    endpoint_pod_names                  # ep-hostname.service.ns.svc.<zone>. -> a.b.c.d
    ignore empty_service                # ingore empty_Service which has no eps.
    noendpoints                         # don't watch endpoints resource

    upstream [ADDRESS…]                 # CNMAE resolve addr
    ttl TTL                             # TTL default: 5s

    transfer to ADDRESS...
    fallthrough [ZONES…]                # not found RR, pass to next plugin
}
```

warm-up #1

Corefile :

```
example.org {
    file    db.example.org
}
```

Contents of file db.example.org

```
$ ORIGIN example.org.
@   3600 IN SOA sns.dns.icann.org. noc.dns.icann.org. (
            2017042745 ; serial
            7200       ; refresh (2 hours)
            3600       ; retry (1 hour)
            1209600    ; expire (2 weeks)
            3600       ; minimum (1 hour)
            )

  3600 IN NS a.iana-servers.net.
  3600 IN NS b.iana-servers.net.

    www     IN A     127.0.0.1
            IN AAAA  ::1
```

$ dig @localhost +noall +answer www.example.org A = ?

$ dig @localhost +noall +answer www.baidu.com A = ?

warm-up #2

Corefile :

```
.{
        proxy .  8.8.8.8:53
        file        db.example.org
}
```

Contents of file db.example.org

```
$ ORIGIN example.org.
@   3600 IN SOA sns.dns.icann.org. noc.dns.icann.org. (
            2017042745 ; serial
            7200       ; refresh (2 hours)
            3600       ; retry (1 hour)
            1209600    ; expire (2 weeks)
            3600       ; minimum (1 hour)
            )

  3600 IN NS a.iana-servers.net.
  3600 IN NS b.iana-servers.net.

    www     IN A     127.0.0.1
            IN AAAA  ::1
```

$ dig @localhost +noall +answer www.example.org A = ?

$ dig @localhost +noall +answer www.baidu.com A = ?

warm-up #3

Corefile :

```
.{
      proxy .  8.8.8.8:53
      file       db.example.org  exampe.org
}
```

Contents of file db.example.org

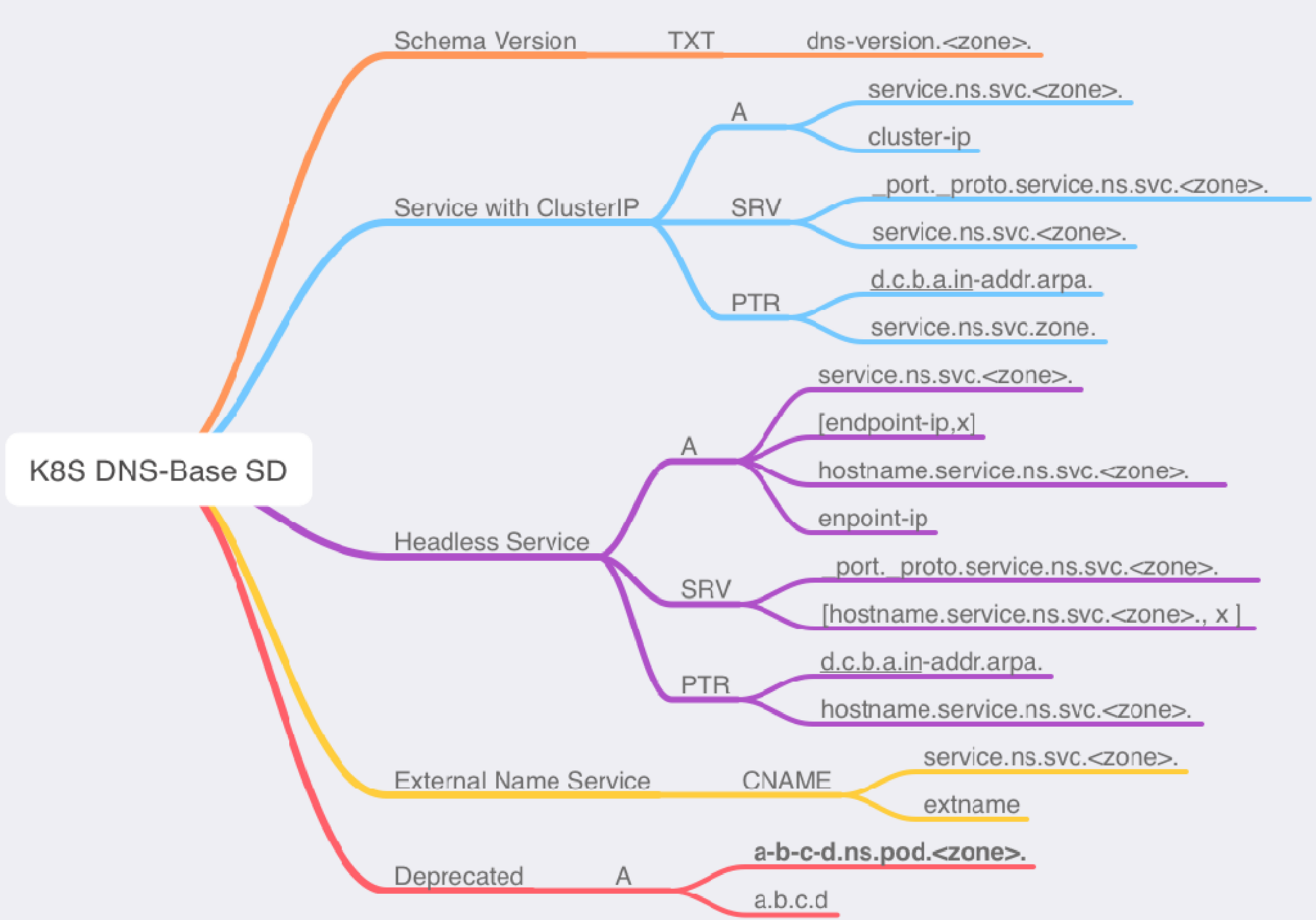
```
$ ORIGIN example.org.
@   3600 IN SOA sns.dns.icann.org. noc.dns.icann.org. (
            2017042745 ; serial
            7200       ; refresh (2 hours)
            3600       ; retry (1 hour)
            1209600    ; expire (2 weeks)
            3600       ; minimum (1 hour)
            )

  3600 IN NS a.iana-servers.net.
  3600 IN NS b.iana-servers.net.

    www     IN A     127.0.0.1
            IN AAAA  ::1
```

$ dig @localhost +noall +answer www.example.org A = ?

$ dig @localhost +noall +answer www.baidu.com A = ?

4. Service Discovery on K8S

```
.:53 {
    errors
    health

    kubernetes cluster.local. in-addr.arpa ip6.arpa {
        pods insecure
        upstream
        fallthrough in-addr.arpa ip6.arpa
    }

    prometheus  :9153
    proxy .  /etc/resolv.conf
    cache  30
    loop
    reload
    loadbalance
}
```



https://github.com/kubernetes/dns/blob/master/docs/specification.md

Indexer Store

object.Service

svc-name.ns → object.ServiceA
svcIP → object.ServiceB
object.ServiceC

api.Namespace

Api.NameSpaceA
Api.NameSpaceB
Api.NameSpaceC

object.Endpoints

ep-name.ns → object.EndpointsA
epIP → object.EndpointsB
object.EndpointsC

object.Pod

podIP → object.PodA
object.PodB
object.PodC

query

dnsController

Kubernetes

Service — List & Watch

Namespace — List & Watch

Endpoints — List & Watch

noendpoints absent

Pod — List & Watch

pods == verified

Node — Get

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
    addonmanager.kubernetes.io/mode: Reconcile
  name: system:coredns
rules:
- apiGroups:
  - ""
  resources:
  - endpoints
  - services
  - pods
  - namespaces
  verbs:
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - nodes
  verbs:
  - get
```

depend on options

- **Watch pod resource depends on：**

  pods == verified

- **Stop watch endpoints resource depend on：**

  noendpoints

- **A Records of hostname on endpoint depend on:**

  `endpoint_pod_names`

pods

```
func (k *Kubernetes) findPods(r recordRequest, zone string) (pods []msg.Service, err error){
}
```

- **disable**

  Default. Do not process pod requests, always returning `NXDOMAIN`

```
if k.podMode == podModeDisabled {
     return nil, errNoItems
}
```

- **insecure**

  Always return an A record with IP from request (without checking k8s). This option is vulnerable to abuse if used maliciously in conjunction with wildcard SSL certs. This option is provided for backward compatibility with kube-dns.

```
if k.podMode == podModeInsecure {
     if !wildcard(namespace) && !k.namespace(namespace) { // no wildcard, but namespace does not exist
          return nil, errNoItems
     }

     // If ip does not parse as an IP address, we return an error, otherwise we assume a CNAME and will try to resolve it in
backend_lookup.go
     if net.ParseIP(ip) == nil {
          return nil, errNoItems
     }

     return []msg.Service{{Key: strings.Join([]string{zonePath, Pod, namespace, podname}, "/"), Host: ip, TTL: k.ttl}}, err
}
```

pods

- **verified**

  Return an A record if there exists a pod in same namespace with matching IP. This option requires substantially more memory than in insecure mode, since it will maintain a watch on all pods.

```
for _, p := range k.APIConn.PodIndex(ip) {
    // If namespace has a wildcard, filter results against Corefile namespace list.
    if wildcard(namespace) && !k.namespaceExposed(p.Namespace) {
        continue
    }

    // exclude pods in the process of termination
    if p.Deleting {
        continue
    }

    // check for matching ip and namespace
    if ip == p.PodIP && match(namespace, p.Namespace) {
        s := msg.Service{Key: strings.Join([]string{zonePath, Pod, namespace, podname}, "/"),
                                                                    Host: ip, TTL: k.ttl}
        pods = append(pods, s)

        err = nil
    }
}
```

fallthrough

Corefile

```
kubernetes cluster.local. in-addr.arpa ip6.arpa {
   endpoint 127.0.0.1:8001
   pods insecure
   endpoint_pod_names
   upstream 192.168.65.101
   fallthrough      # in-addr.arpa ip6.arpa
}

file cluster.db cluster.local
```

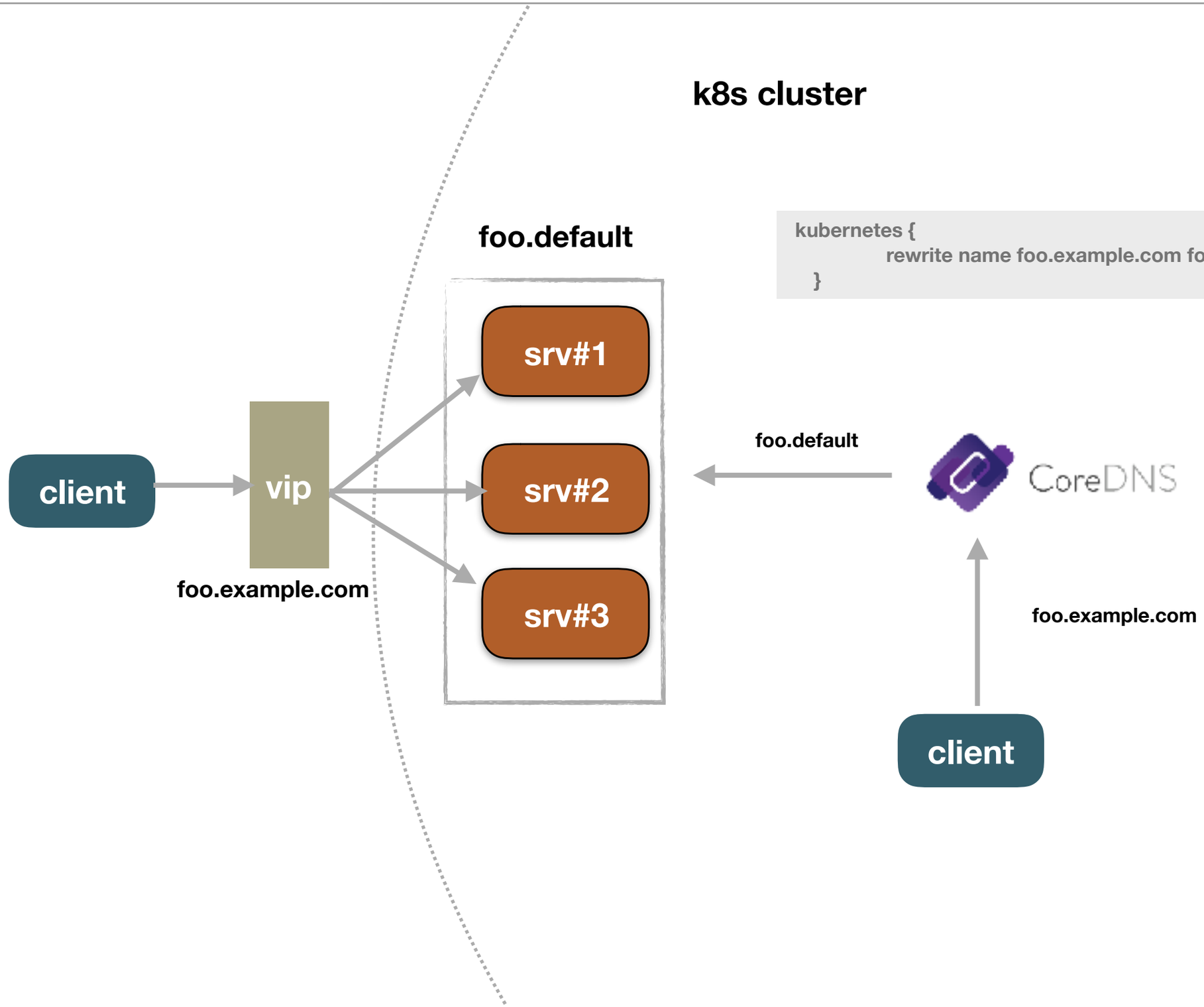
cluster.db

```
cluster.local.          IN      SOA     ns.dns.cluster.local.
                                        hostmaster.cluster.local.
                                        2015082541
                                        7200
                                        3600
                                        1209600
                                        3600

something.cluster.local.    IN      A       10.0.0.1
otherthing.cluster.local.   IN      CNAME   google.com.
```

$ dig -p 1053 @localhost +noall +answer kubernetes.default.svc.cluster.local

$ dig -p 1053 @localhost +noall +answer otherthing.cluster.local

rewrite




```
kubernetes {
        rewrite name foo.example.com foo.default.svc.cluster.local
  }
```

autopath

ns: default pod: nginx

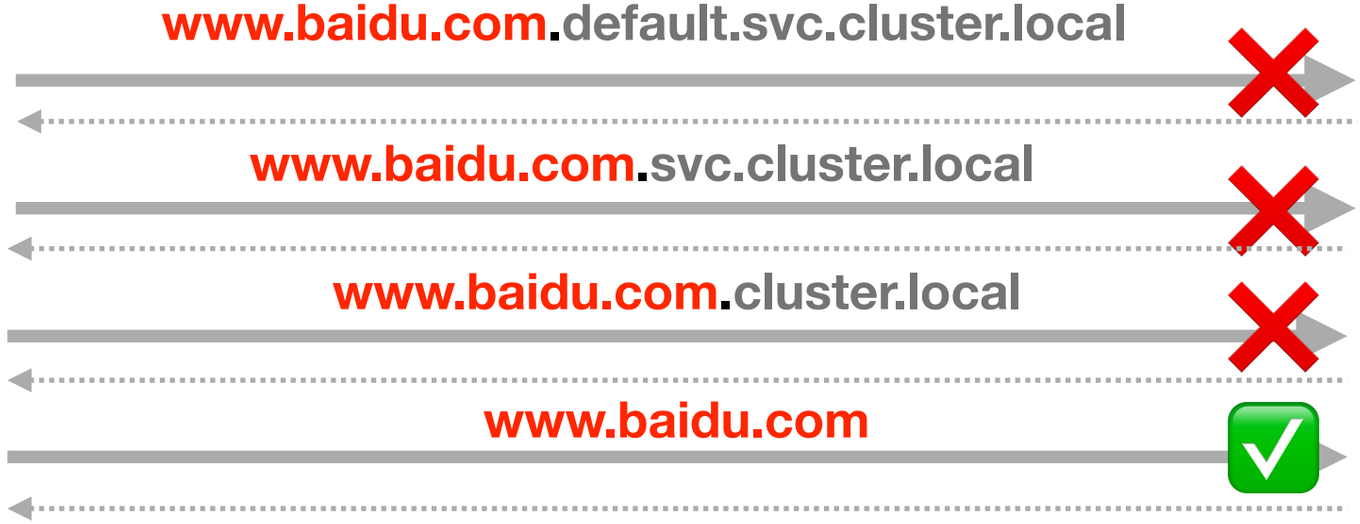
search **default.svc.cluster.local**
svc.cluster.local
cluster.local
ndot:5

→ **coredns.kube-system**

coredns.kube-system.default.svc.cluster.local

→ **www.baidu.com**

www.baidu.com.default.svc.cluster.local



ndot:2

autopath

ns: default **pod: nginx**

search **default.svc.cluster.local**
svc.cluster.local
cluster.local
ndot:5

autopath default.svc.cluster.local
svc.cluster.local
cluster.local

coredns.kube-system

coredns.kube-system.default.svc.cluster.local

CoreDNS

coredns.kube-system.svc.cluster.local

✅

www.baidu.com

www.baidu.com.default.svc.cluster.local

CoreDNS

www.baidu.com.svc.cluster.local
www.baidu.com.cluster.local
www.baidu.com

✅

autopath

ns: default **pod: nginx**

search **default.svc.cluster.local**
svc.cluster.local
cluster.local
ndot:5

autopath @kubernetes

coredns.kube-system

default.svc.cluster.local
svc.cluster.local
cluster.local

coredns.kube-system.default.svc.cluster.local

10.10.0.2

CoreDNS

coredns.kube-system.svc.cluster.local

autopath

ns: test **pod: nginx**

search **test.svc.cluster.local**
svc.cluster.local
cluster.local
ndot:5

autopath @kubernetes

coredns.kube-system

test.svc.cluster.local
svc.cluster.local
cluster.local

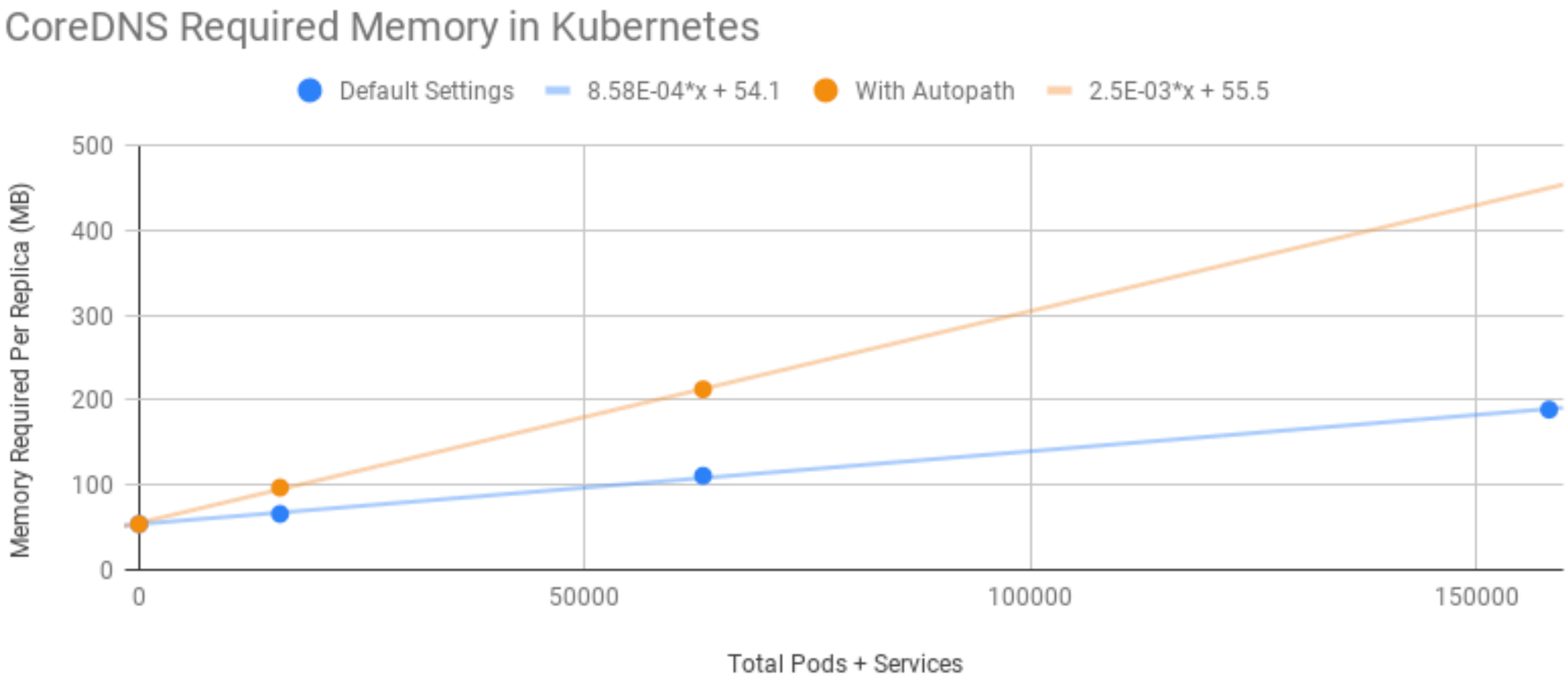
coredns.kube-system.test.svc.cluster.local

CoreDNS

coredns.kube-system.svc.cluster.local

autopath

There is no free lunch



From: https://github.com/coredns/deployment/blob/master/kubernetes/Scaling_CoreDNS.md

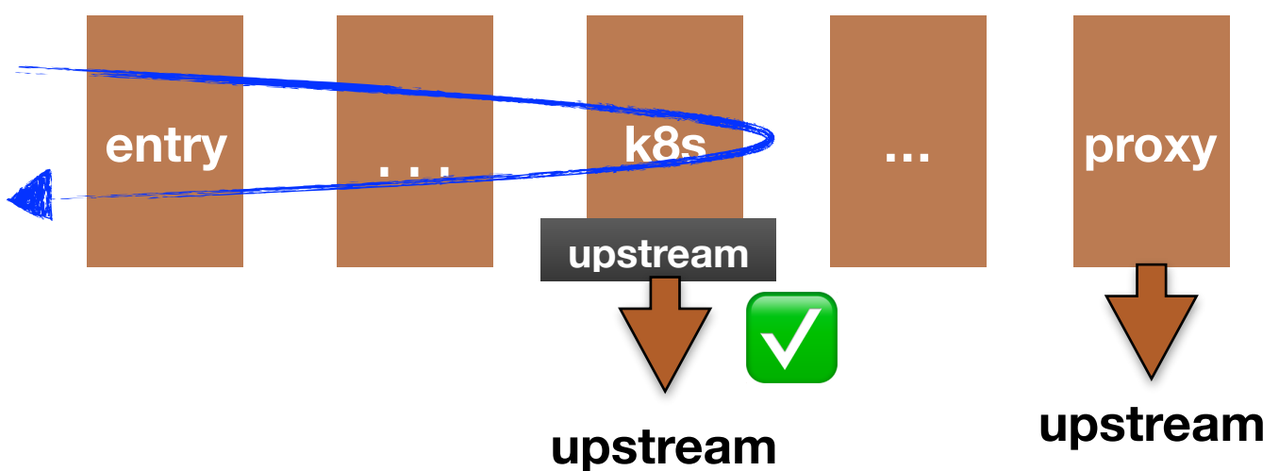
upstream

```
kind: Service
apiVersion: v1
metadata:
 name: baidu
spec:
 type: ExternalName
 externalName: www.baidu.co
```

```
kubernetes cluster.local. in-addr.arpa ip6.arpa {
  endpoint 127.0.0.1:8001
  pods insecure
  endpoint_pod_names
  upstream 192.168.65.101
  fallthrough
}

 proxy . /etc/resolv.conf
```

$ dig -p 1053 @localhost +noall +answer baidu.default.svc.cluster.local

```
baidu.default.svc.cluster.local. 5 IN CNAME    www.baidu.com.
www.baidu.com.          5    IN   CNAME    www.a.shifen.com.
www.a.shifen.com.       5    IN   A    115.239.210.27
```



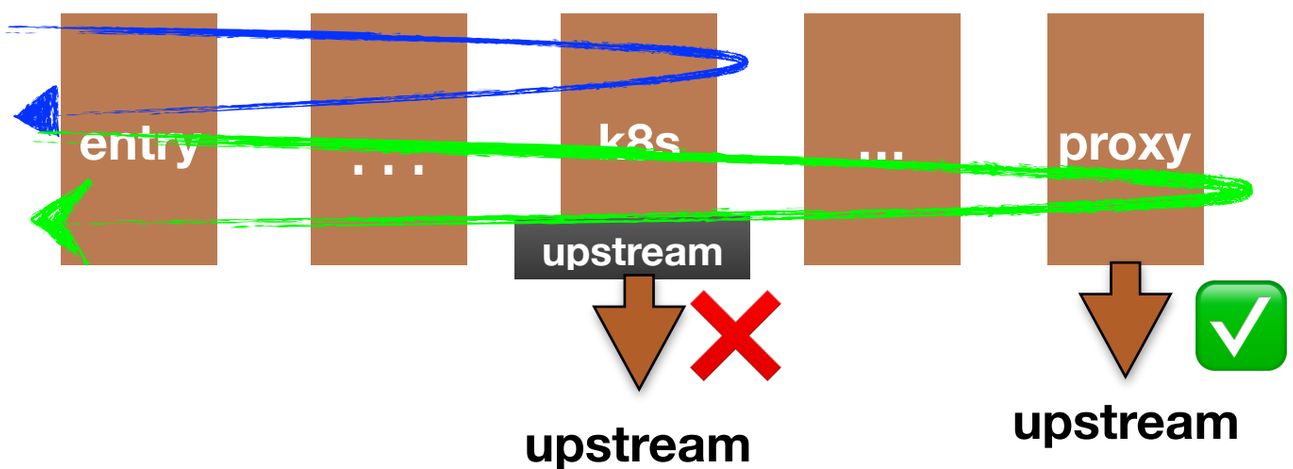
upstream

```
kind: Service
apiVersion: v1
metadata:
 name: baidu
spec:
 type: ExternalName
 externalName: www.baidu.co
```

```
kubernetes cluster.local. in-addr.arpa ip6.arpa {
  endpoint 127.0.0.1:8001
  pods insecure
  endpoint_pod_names
  upstream
  fallthrough
}

 proxy . /etc/resolv.conf
```

$ dig -p 1053 @localhost +noall +answer baidu.default.svc.cluster.local

```
baidu.default.svc.cluster.local. 5 IN CNAME    www.baidu.com.
www.baidu.com.          5    IN   CNAME    www.a.shifen.com.
www.a.shifen.com.       5    IN   A    115.239.210.27
```



wildcards

1. service

   s*ervice*.namespace.svc.zone, e.g. `*.ns.svc.cluster.local`

2. endpoint

   ***endpoint*.service.namespace.svc.zone, e.g. `*.nginx.ns.svc.cluster.local`**

3. namespace

   service.*namespace*.svc.zone, e.g. `nginx.*.svc.cluster.local`

4. port or protocol

   _*port*._*protocol*.service.namespace.svc.zone., e.g. `_http.*.service.ns.svc.cluster.local`

5. multiple wild

   `A` Request `*.*.svc.zone.` or `SRV` request `*.*.*.*.svc.zone`

Thank You !