KubeCon | CloudNativeCon

North America 2023

KubeCon | CloudNativeCon

North America 2023

CoreDNS Plugins: A Deep Dive

Yong Tang, Ivanti & John Belamaric, Google

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
- AWS Route53, Azure DNS, Google Cloud DNS sync up

CoreDNS Community

- ***350*** Contributors (Big Thanks!)
- 28 Maintainers
- 35 Public Adopters
- ***11,200+*** Stars
- Project Lead => Steering Committee

Project Update

- 1.10.1- 1.11.1
  - **1.11.1** Released August 2023
- New plugins:
  - **timeouts** - allows configuration of server listener timeout
- New features:
  - **acl** - drop queries as an action
  - **template** - create responses with extended DNS errors
  - **cache** - option to serve original record TTLs from cache
  - **loadbalance** - adds CNAME target rewrites to plugin
  - **dnstap -** add support for "extra" field in payload
- Built with golang 1.21

Service Discovery

- DNS is a nice indirection
- Maximum flexibility
- Public-facing and internal
- Easy and simple to change
- Distributed in nature, scales to Internet
- Pervasive in IT infrastructure

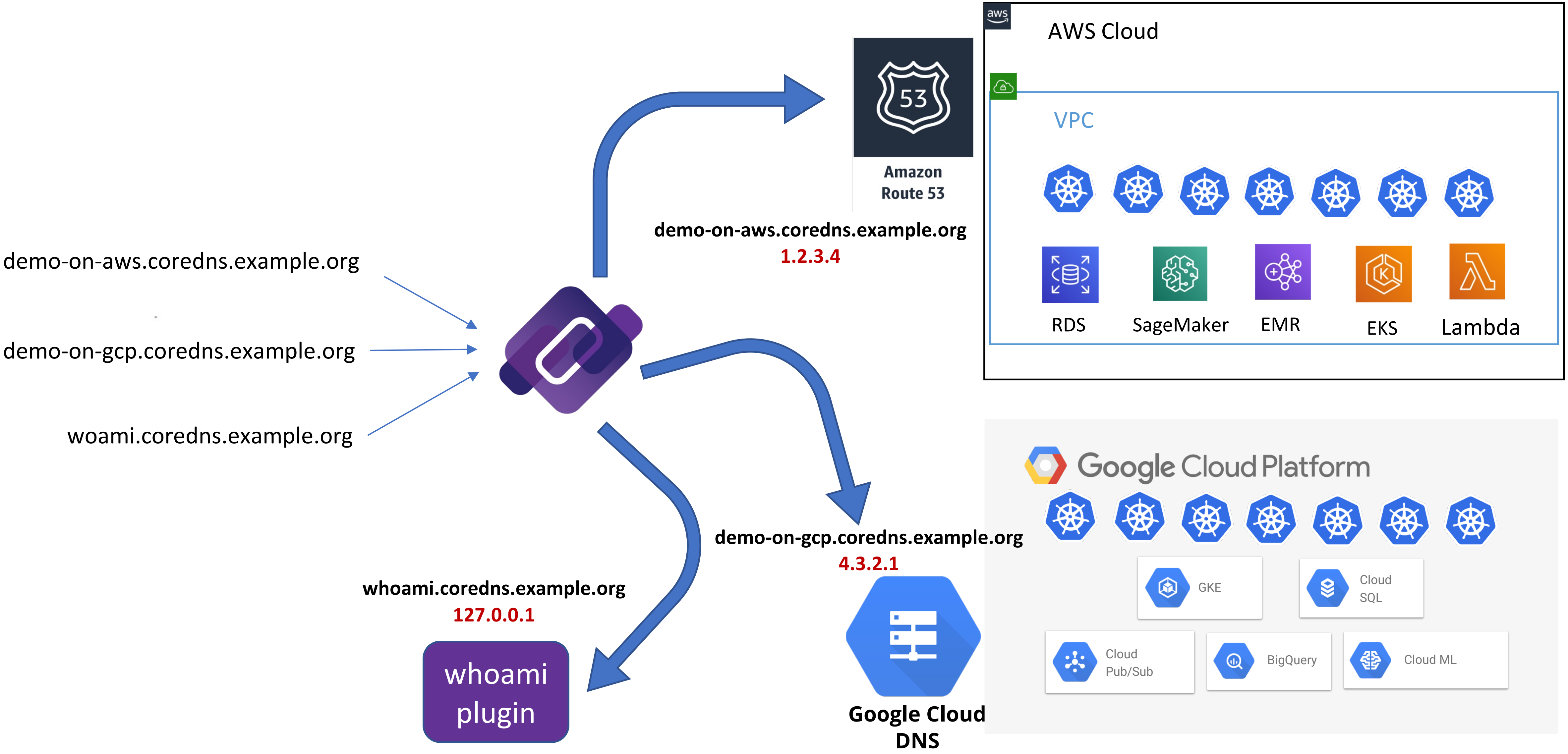
Cloud Sync with CoreDNS

- Secure communication (HTTPS vs UDP)
- Authentication and authorization
- Reliable and better error handling (TCP vs UDP)
- Separation of data sync up & DNS query

Multi-Cloud

- Multi-Cloud
  - Data Sovereignty & Residency
  - Merger & Acquisition
- Diversified source information
  - Kubernetes, cloud vendors (Customers)
  - IT infrastructure (Corp)
- “100%” SLA
  - DNS services by cloud vendors
  - Losses when SLA is not met

AWS+Google Cloud



demo-on-aws.coredns.example.org

demo-on-gcp.coredns.example.org

woami.coredns.example.org

demo-on-aws.coredns.example.org
1.2.3.4

Amazon Route 53

AWS Cloud

VPC

RDS

SageMaker

EMR

EKS

Lambda

demo-on-gcp.coredns.example.org
4.3.2.1

Google Cloud DNS

Google Cloud Platform

GKE

Cloud SQL

Cloud Pub/Sub

BigQuery

Cloud ML

whoami.coredns.example.org
127.0.0.1

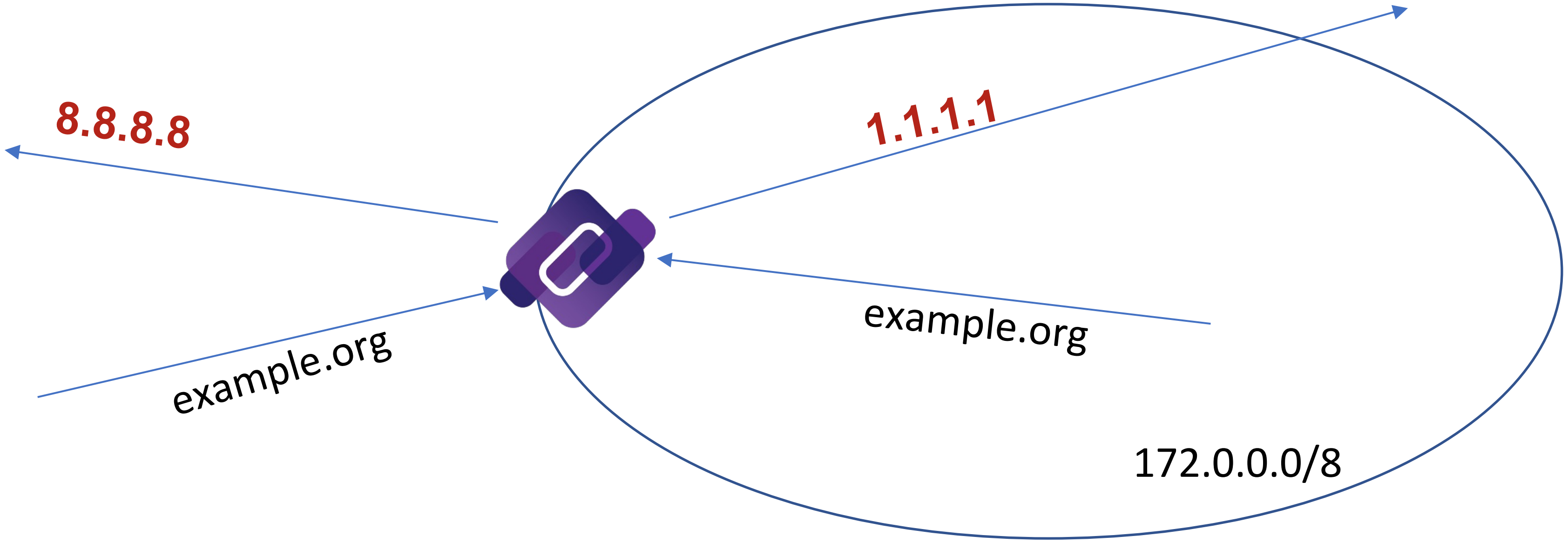
whoami plugin

AWS+Google Cloud: Corefile

```
.:53 {
    # Route53 (Amazon AWS)
    route53 coredns.example.org.:Z0123456789012345678 9 {
       fallthrough      # <- move to next plugin (clouddns)
    }
    # Cloud DNS (Google Cloud)
    clouddns coredns.example.org.:peerless-dahlia-123456:coredns-example-zone {
       fallthrough      # <- move to next plugin (e.g., whoami)
    }
    # Fallthrough (e.g., whoami)
    whoami
    # …
}
```

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 - Functions

[setup.go]:

init()

- Performs one-time initializations, register the setup function with Caddy

setup(c *caddy.Controller) error

- Parses the configuration from the file and captures it in a struct
- Adds the handler to the Config object (represents a stanza)
- Called once for each use of the plugin in the Corefile

[demo.go]:

ServeDNS(context.Context, dns.ResponseWriter,*dns.Msg) (int, error)

- Processes the DNS request and returns a response, or,
- Passes it down the chain

Demo Plugin – init()/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
}
```

Demo Plugin – ServeDNS()

```
// 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)
        answers := []dns.RR{}
        …
}
```

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:1.21)
$ docker run --rm -i -t \
$   -v $PWD:/go/src/github.com/coredns/coredns \
$     -w /go/src/github.com/coredns/coredns \
$       golang:1.21 sh -c \
$           'GOFLAGS="-buildvcs=false" make gen && GOFLAGS="-buildvcs=false" make'
$
$ # configure Corefile and run coredns
$
$ ./coredns
$
```

Demo Plugin – Source Code

https://github.com/coredns/demo

CoreDNS Contributions

- Star CoreDNS in GitHub
  - https://github.com/coredns/coredns
- Add the name to ADOPTERS.md
- Become a contributor
  - Create a pull request
- Become a maintainer
  - One significant pull request
  - Sponsored by one current maintainer

THANK YOU

Please scan the QR Code above
to leave feedback on this session