



KubeCon



CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

CoreDNS Intro & Deep Dive

Yong Tang

Maintainer: CoreDNS

GitHub: yongtang

- 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
- Started by Miek Gieben

- 1.9.3 – 1.10.0
 - 1.10.0 Released *September 2022*
 - New plugins: **tsig**, **view**
 - **tsig** validating TSIG requests and signing responses
 - **view** enabling advanced server-block routing configurations
 - Security Fixes (upstream)
 - *CVE-2022-27191*: golang.org/x/crypto
 - *CVE-2022-28948*: gopkg.in/yaml.v3
 - Built with golang **1.19.1+** since 1.10.0
 - golang < 1.17.6 security issues

CoreDNS Community

- 332 Contributors (**Big Thanks!**)
- 28 Maintainers
- 33 Public Adopters
- **9,800+** Stars
- Steering Committee

Service Discovery with CoreDNS

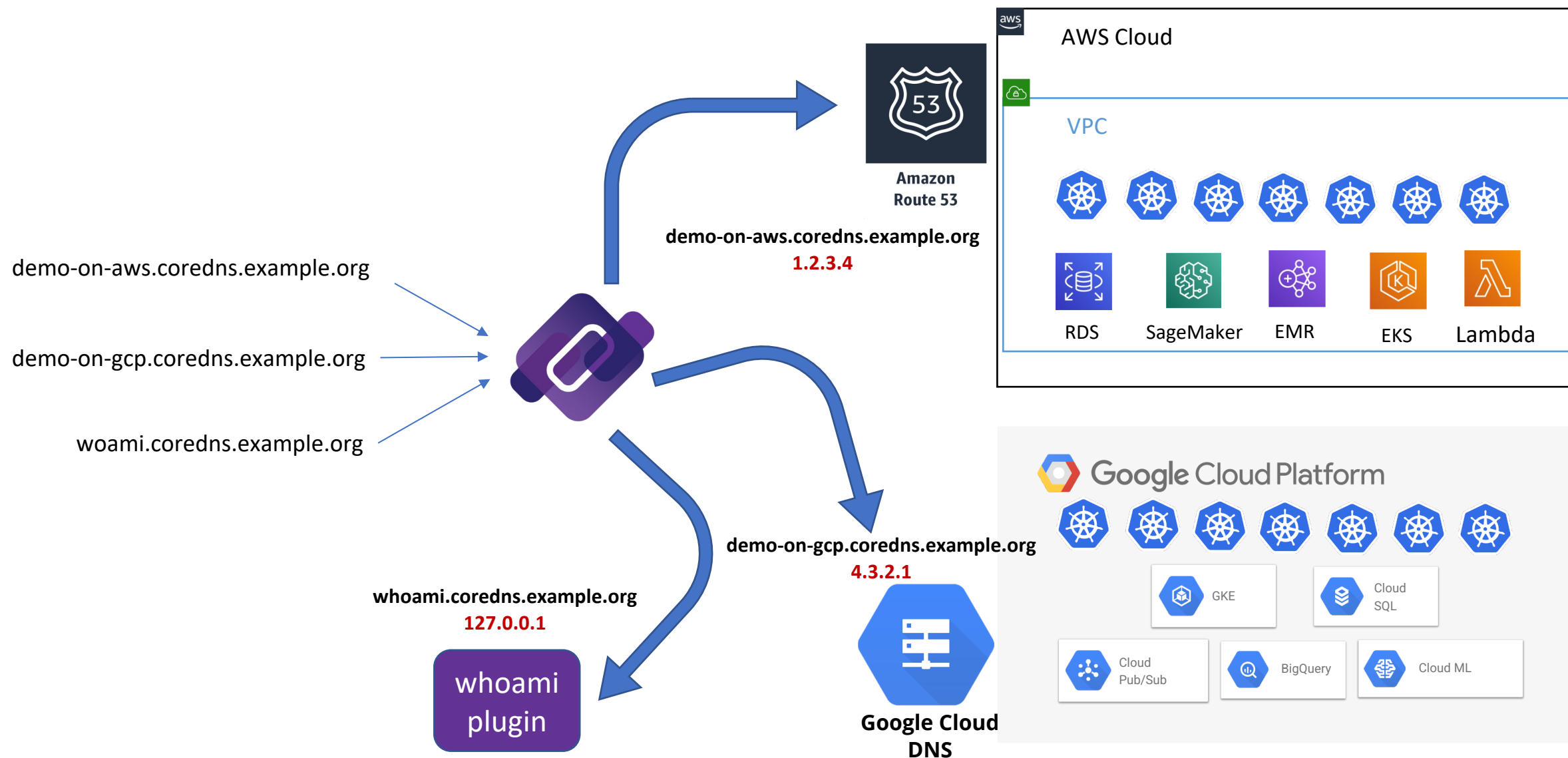
- DNS is a nice indirection
- Maximum flexibility
- Public-facing and internal
- Easy and simple to change
- Distributed in nature, scales to Internet
- Pervasive in IT infrastructure

- Multi-Cloud
 - Data sovereignty & data residency
 - Merger & Acquisition (M&A)
- Diversified source information
 - Kubernetes, cloud vendors, IT infrastructure
- Misleading “100%” SLA from all cloud vendors
 - DNS services by cloud vendors are not expensive
 - Service credit refund almost none when uptime is not meet

CoreDNS: Cloud Integrations

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

CoreDNS: AWS + Google Cloud

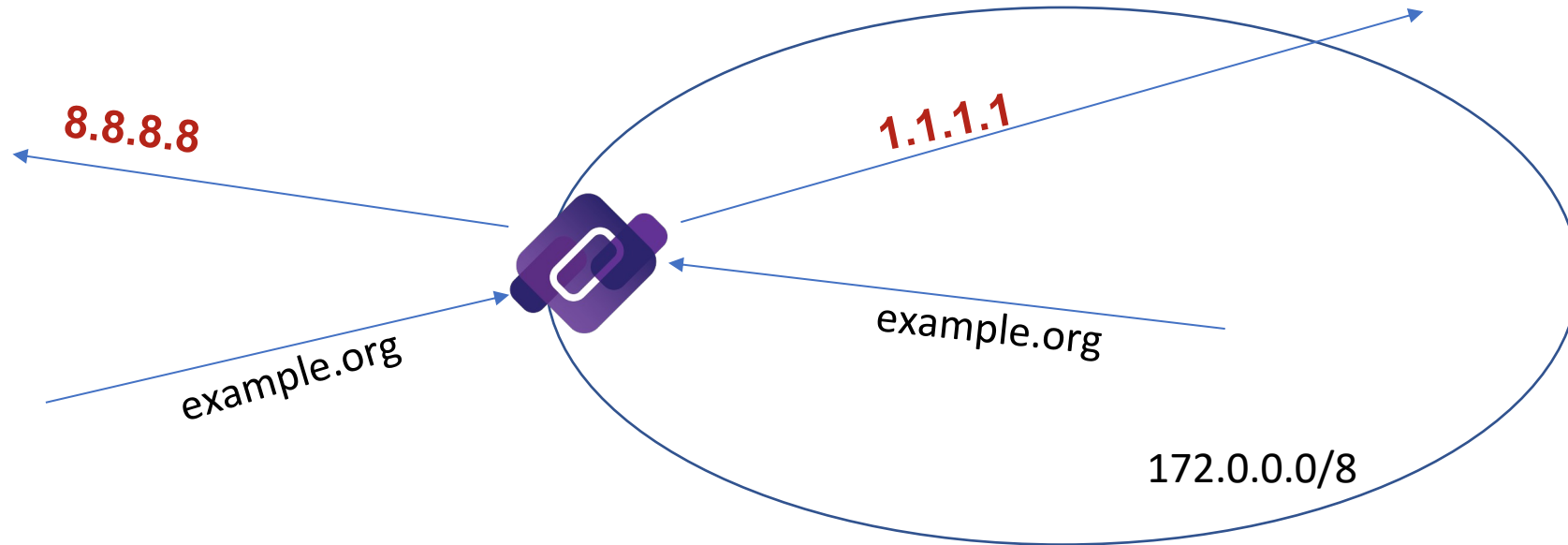


CoreDNS: AWS + Google Cloud

```
• .:53 {  
•   # Route53 (Amazon AWS)  
•   route53 coredns.example.org.:Z01234567890123456789 {  
•     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  
•   # ...  
• }
```

CoreDNS: Demo Plugin in Golang

- 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: Source Code

<https://github.com/coredns/demo>

Demo Plugin: 3 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:1.18)  
$ docker run --rm -i -t -v \  
$     $PWD:/v -w /v golang:1.18 sh -c "make gen && 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