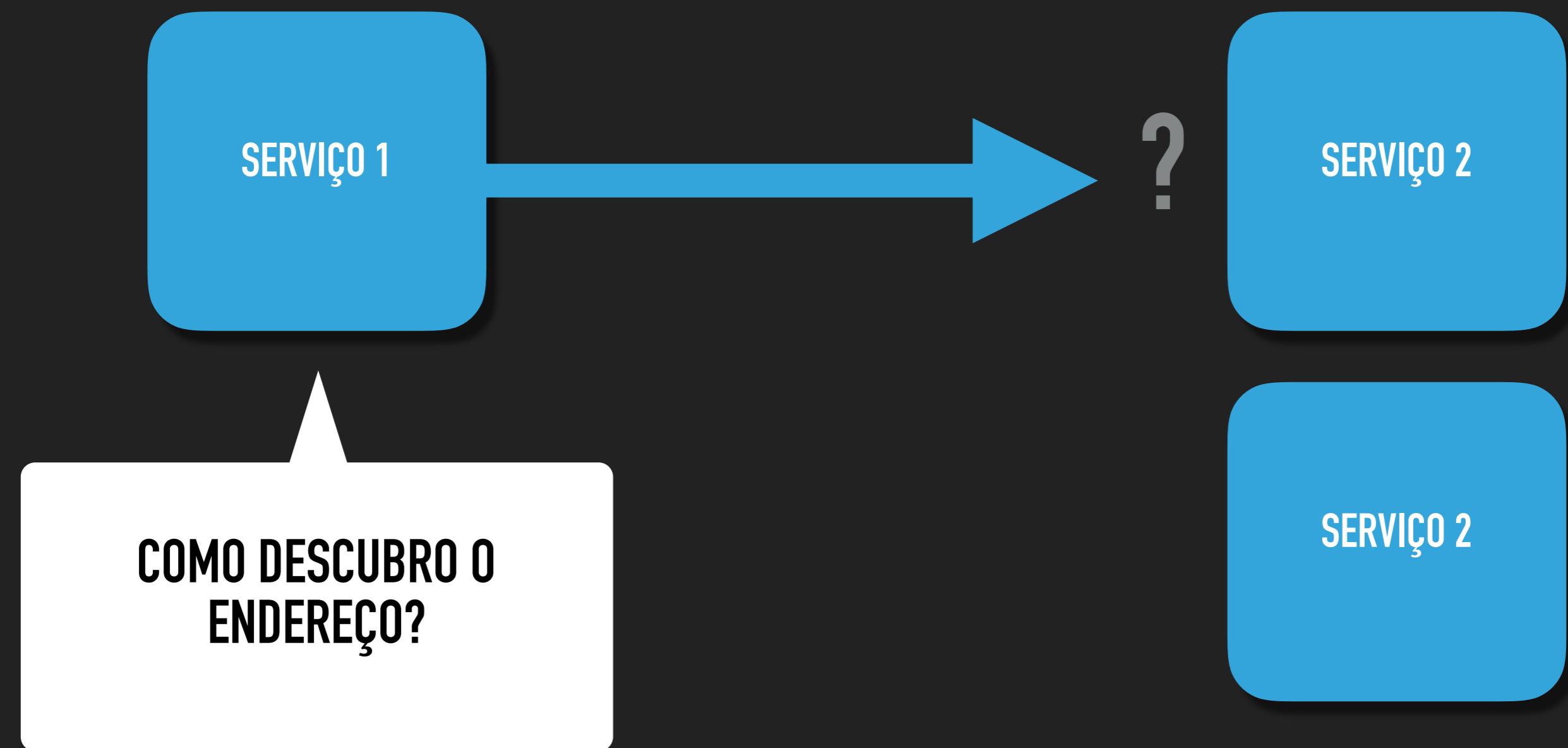




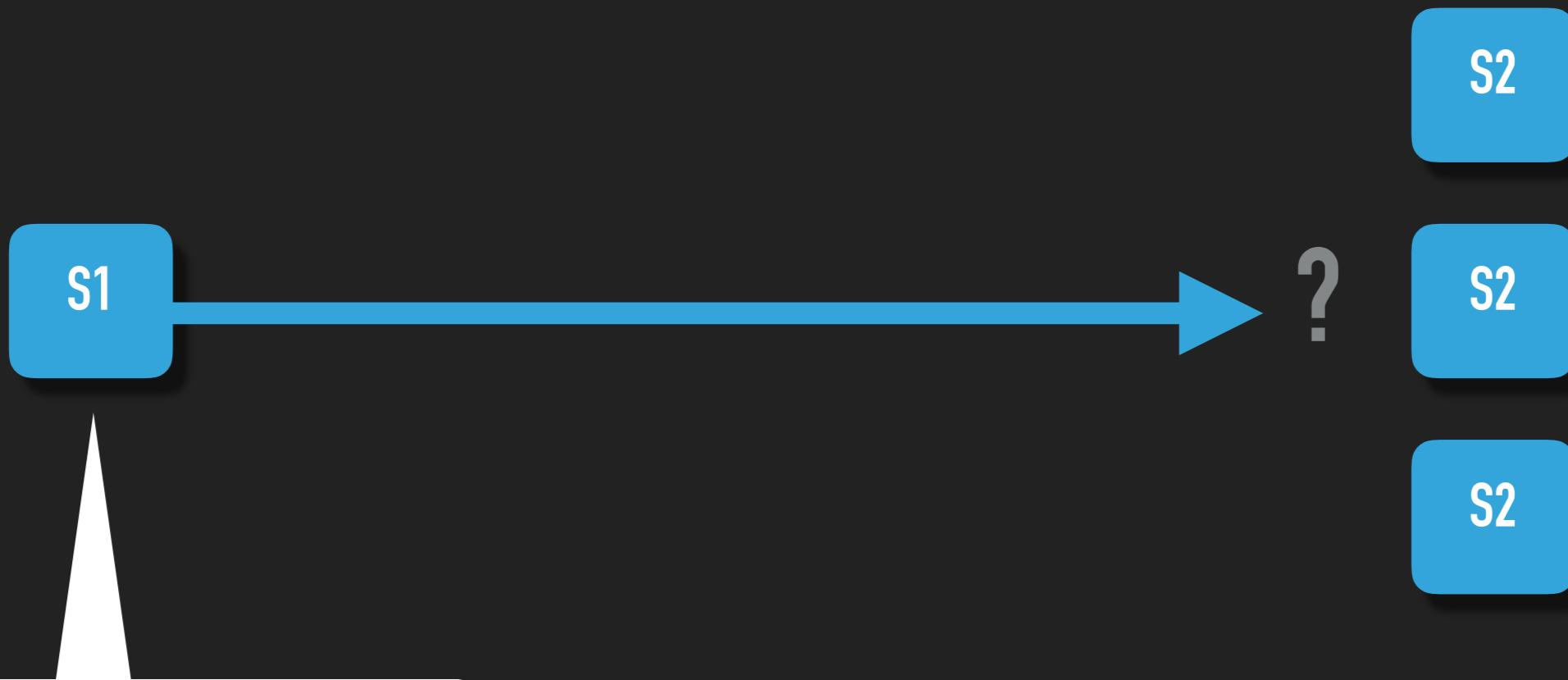
DESCOBERTA DE SERVIÇOS VIA DNS

DNSDISCO

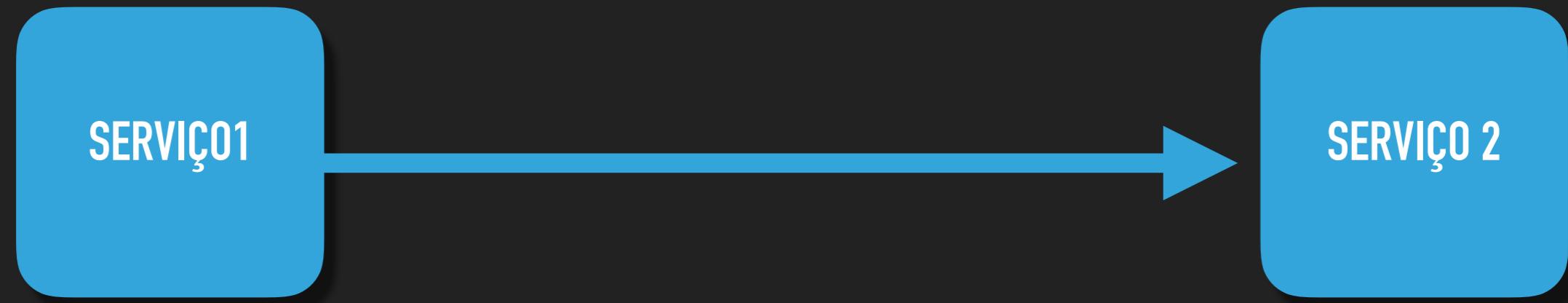
SERVIÇO



(MICRO)?SERVIÇO



**COMO DESCUBRO O
ENDEREÇO?**



O QUE AS SOLUÇÕES DEVERIAM ME OFERECER?

- ▶ Resiliência
- ▶ Desempenho
- ▶ Simplicidade
- ▶ Elasticidade



BALANCEADOR DE CARGA

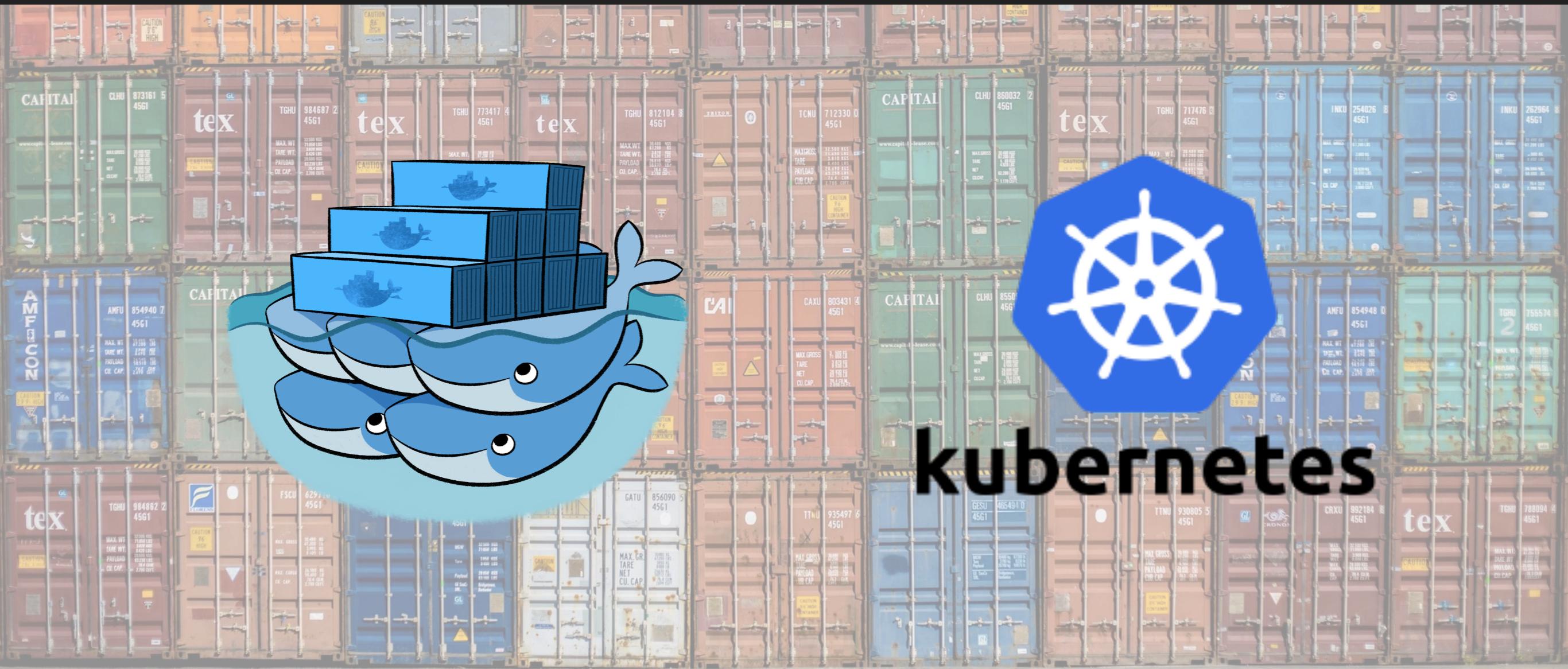
CENTRALIZADORES DE CONFIGURAÇÃO



FERRAMENTAS DE AUTOMAÇÃO



ORQUESTRAÇÃO DE CONTAINERS



O CLÁSSICO ARQUIVO DE CONFIGURAÇÃO

```
11
12     ....location /papers {-
13         .....alias /Users/tyler/works;
14
15         .....if ($request_filename ~* ".pdf") {
16             .....expires 30d;
17         }
18     }
19
20 » # if you don't like seeing a
21 » location = /favicon.ico { acc...
22
23 » # if you don't like seeing e
24 » location = /robots.txt { acc...
25
26     ....#error_page 404
27
28     ....# redirect server error pages
29     ....#
local/nginx/lab.tylergaw 1,1 Plain Text
```

**TEMOS MAIS UMA
OPÇÃO!**

O BOM E VELHO DNS

DNS

exemplo.com.br. 86400 IN SOA ...

exemplo.com.br. 86400 IN NS ns1.exemplo.com.br.

exemplo.com.br. 86400 IN A 192.0.2.1

exemplo.com.br. 86400 IN MX 1 m.exemplo.com.br

DNS

DNS

NSEC3PARAM TALINK NSAP UID OPT NIMLOC TSIG NAPTR MF ISDN MD DHCID SRV CDS GID NSEC UINFO A6 NSEC3 X25 AFSDB LOC MX CAA EID DNSKEY SINK ATMA L32 IPSECKEY SPF TA NSAP-PTR MAILA PTR CERT L64 TXT KX IXFR EUI64 DS SSHFP SIG KEY AXFR HINFO CSYNC UNSPEC AVC AAAA NINFO SMIMEA GPOS CNAME RRSIG MAILB RKEY DNAME LP NID

DNS

NSEC3PARAM
TALINK
NIMLOC
TSIG
DHCID
UINFO
MR
LOC
EUI48
APL
L32
SPF
NSAP
UID
OPT
NAPTR
MF
ISDN
CDS
GID
NSEC3
X25
AFSDB
MX
CAA
DNSKEY
SINK
ATMA
IPSECKEY
RKEY
NULL
TA
NS
PTR
CERT
NXT
EUI64
RT
A
WKS
DLV
MB
HIP
AXFR
UNSPEC
EID
MG
AVC
DNAME
LP
NID
AAAA
CNAME
RRSIG
URI
OPENPGPKEY
NSAP-PTR
MINFO
SOA
PX
IXFR
SSHFP
DS
CDNSKEY
SIG
KEY
HINFO
CSYNC
NINFO
SMIMEA
GPOS

SRV

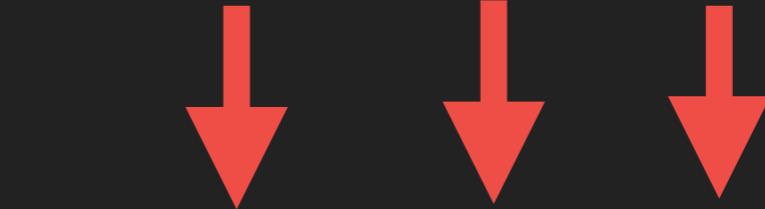
DNS

\$ORIGIN exemplo.com.br.

_https._tcp 86400 IN SRV 0 5 443 s1.servico



NOME DO
SERVIÇO PROTOCOLO



PRIORIDADE PESO PORTA

DNS

\$ORIGIN exemplo.com.br.

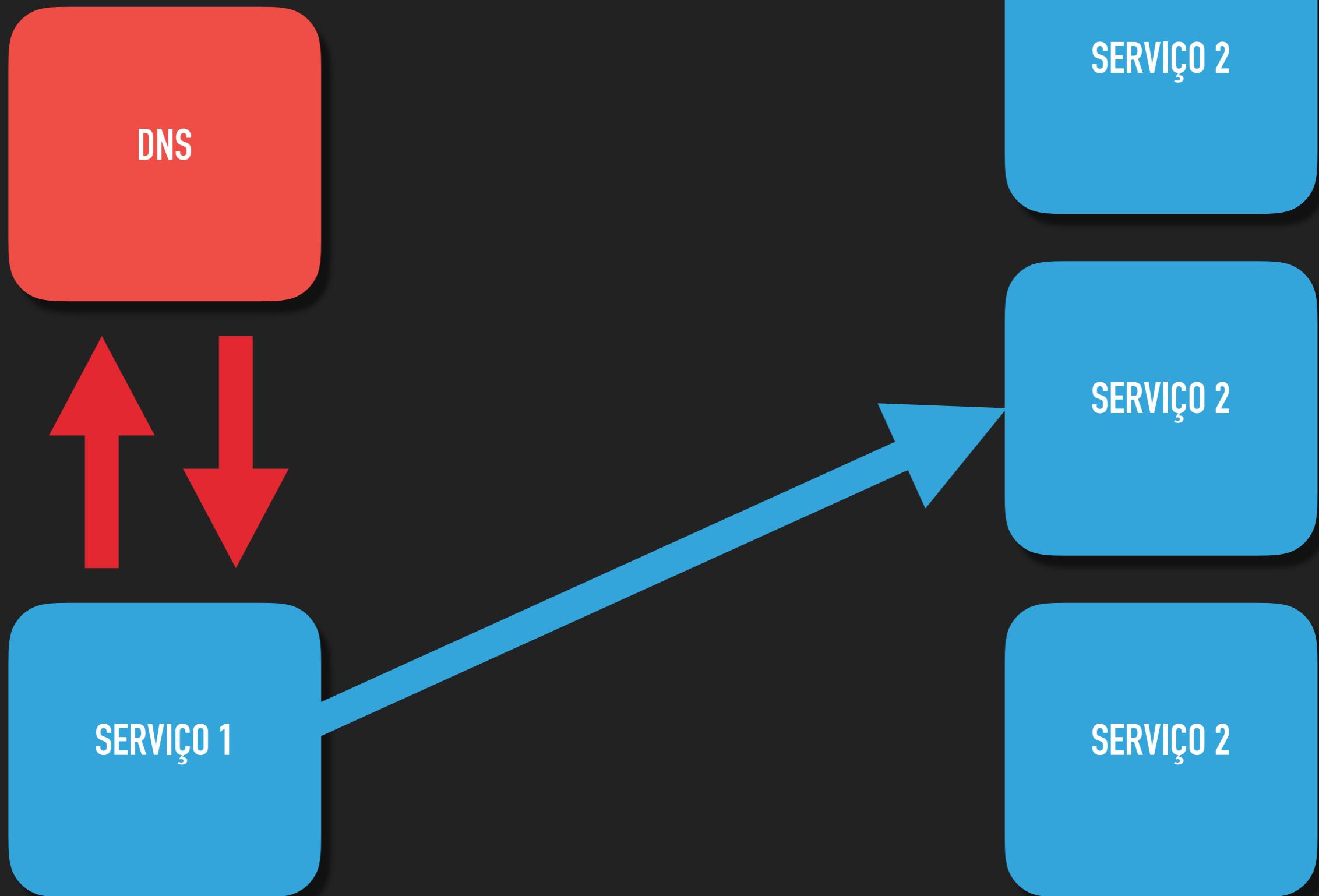
_https._tcp 86400 IN SRV 0 5 443 s1.servico

_https._tcp 86400 IN SRV 0 7 443 s2.servico

_https._tcp 86400 IN SRV 1 9 443 s3.servico

DNSDISCO

SERVIÇO



CARACTERÍSTICAS

- ▶ Busca de serviços no DNS
- ▶ Health check
- ▶ Algoritmo de escolha baseado na RFC 2782
- ▶ Flexibilidade

COMO USAR?

```
package main

import (
    "fmt"

    "github.com/rafaeljusto/dnsdisco"
)

func main() {
    target, port, err := dnsdisco.Discover("jabber", "tcp", "registro.br")
    if err != nil {
        fmt.Println(err)
        return
    }

    fmt.Printf("Target: %s\nPort: %d\n", target, port)
}
```

COMO USAR?

```
package main

import (
    "fmt"

    "github.com/rafaeljusto/dnsdisco"
)

var discovery dnsdisco.Discovery

func init() {
    discovery = dnsdisco.NewDiscovery("jabber", "tcp", "registro.br")
}

func main() {
    if err := discovery.Refresh(); err != nil {
        fmt.Println(err)
        return
    }

    target, port := discovery.Choose()
    fmt.Printf("Target: %s\nPort: %d\n", target, port)
}
```

COMO USAR?

```
func main() {
    if err := discovery.Refresh(); err != nil {
        fmt.Println(err)
        return
    }

    finish := discovery.RefreshAsync(5 * time.Second)
    defer close(finish)

    for {
        if errs := discovery.Errors(); len(errs) > 0 {
            fmt.Println(errs)
            return
        }

        target, port := discovery.Choose()
        fmt.Printf("Target: %s\nPort: %d\n", target, port)
        time.Sleep(time.Second)
    }
}
```

FLEXIBILIDADE - HEALTH CHECKER

```
func healthChecker(target string, port uint16, _ string) (bool, error) {
    r, err := http.Get(fmt.Sprintf("http://%s:%d/teste", target, port))
    return err == nil && r.StatusCode == http.StatusNoContent, err
}

func main() {
    // ...
    discovery.SetHealthChecker(dnsdisco.HealthCheckerFunc(healthChecker))
    // ...
}
```

FLEXIBILIDADE - BALANCEADOR

```
type loadBalancer struct {
    servers []*net.SRV
}

func (l *loadBalancer) ChangeServers(servers []*net.SRV) {
    l.servers = servers
}

func (l loadBalancer) LoadBalance() (target string, port uint16) {
    if len(l.servers) > 0 {
        target = l.servers[0].Target
        port = l.servers[0].Port
    }
    return
}

func main() {
    // ...
    discovery.SetLoadBalancer(new(loadBalancer))
    // ...
}
```

FLEXIBILIDADE - OBTENTOR

```
func retriever(service, proto, name string) ([]*net.SRV, error) {
    name = strings.TrimRight(name, ".")
    z := fmt.Sprintf("_%s._%s.%s.", service, proto, name)

    var request dns.Msg
    request.SetQuestion(z, dns.TypeSRV)
    request.RecursionDesired = true

    response, err := dns.Exchange(&request, "8.8.8.8:53")
    if err != nil {
        return nil, err
    }

    var servers []*net.SRV
    // conversão de response.Answers para []*net.SRV
    return servers, nil
}

func main() {
    // ...
    discovery.SetRetriever(dnsdisco.RetrieverFunc(retriever))
    // ...
}
```



github.com/rafaeljusto/dnsdisco

EXEMPLO

<https://gist.github.com/rafaeljusto/ec64b408ec240df52f77>

The screenshot shows a terminal window with four panes, each displaying a command-line interface for the DNSDISCO project.

- Panels 1 and 2:** Show the client side. The left panel has the command `rafael@dnsdisco-client:~|=> vim db.dnsdisco.com`. The right panel has the command `rafael@dnsdisco-server1:~|=> ./dnsdisco-server -name s1`.
- Panels 3 and 4:** Show the server side. The left panel has the command `rafael@dnsdisco-server2:~|=> ./dnsdisco-server -name s2`. The right panel has the command `rafael@dnsdisco-server3:~|=> ./dnsdisco-server -name s3`.

The bottom-left pane displays the contents of the `db.dnsdisco.com` file, which contains the following DNS records:

```
dnsdisco.com. 86400 IN SOA ns1.dnsdisco.com. adm.rafael.net.br. (
    2016022400 ; serial
    86400       ; refresh
    86400       ; retry
    86400       ; expire
    10 )        ; minimum

dnsdisco.com. 86400 IN NS ns1.dnsdisco.com.

_test._tcp.dnsdisco.com. 10 IN SRV 1 10 7777 s1.dnsdisco.com.
_test._tcp.dnsdisco.com. 10 IN SRV 10 90 7777 s2.dnsdisco.com.
;_test._tcp.dnsdisco.com. 10 IN SRV 2 30 7777 s3.dnsdisco.com.

s1.dnsdisco.com. 86400 IN A 192.168.122.52
s2.dnsdisco.com. 86400 IN A 192.168.122.110
s3.dnsdisco.com. 86400 IN A 192.168.122.173
ns1.dnsdisco.com. 86400 IN A 127.0.0.1

"db.dnsdisco.com" 17L, 568C
```

The bottom center of the terminal window shows the status bar with "1.1" and "All".