
DigitalOcean

Global Container Networks on Kubernetes

Andrew Sy Kim - Software Engineer at DigitalOcean

 @kimandrewsy  @andrewsykim

Kubernetes at DigitalOcean

First Kubernetes cluster on v1.1

Challenges Early On

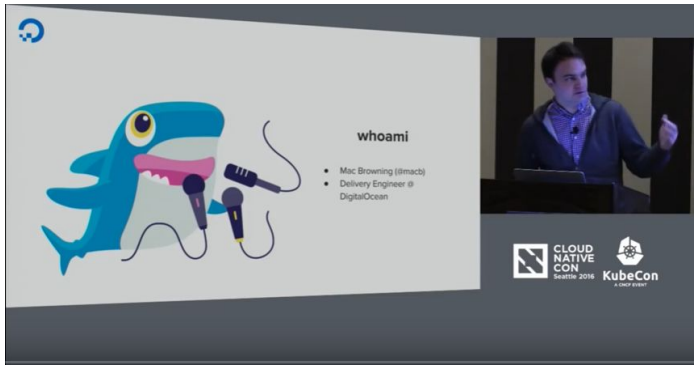
- Kubernetes is a moving target.
  - Best practices and features evolving
- Kubernetes has all the features
  - And exposes knobs for all of them
- Too much YAML!

DOCC

Internal PaaS

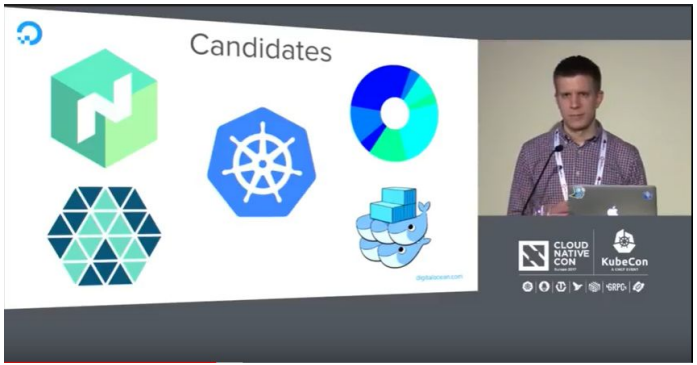
- Built on top of Kubernetes
- Multitenant
- Built with simplicity
- Promotes (and sometimes enforces) best practices

Check out some previous talks!



Delivering Services at DigitalOcean
https://youtu.be/K5WRJvMx4us

Kubernetes at DigitalOcean:
Building a Platform for the Future
https://youtu.be/Jhfd5FjYimU



DOCC at Scale

- **850** applications
- **2000** pods
- **3000** docker containers
- **3500** deploys per month
- **15+** clusters across many regions
- **3500+** cores
- **10TB** of memory
- **50+** service owners

The Good!

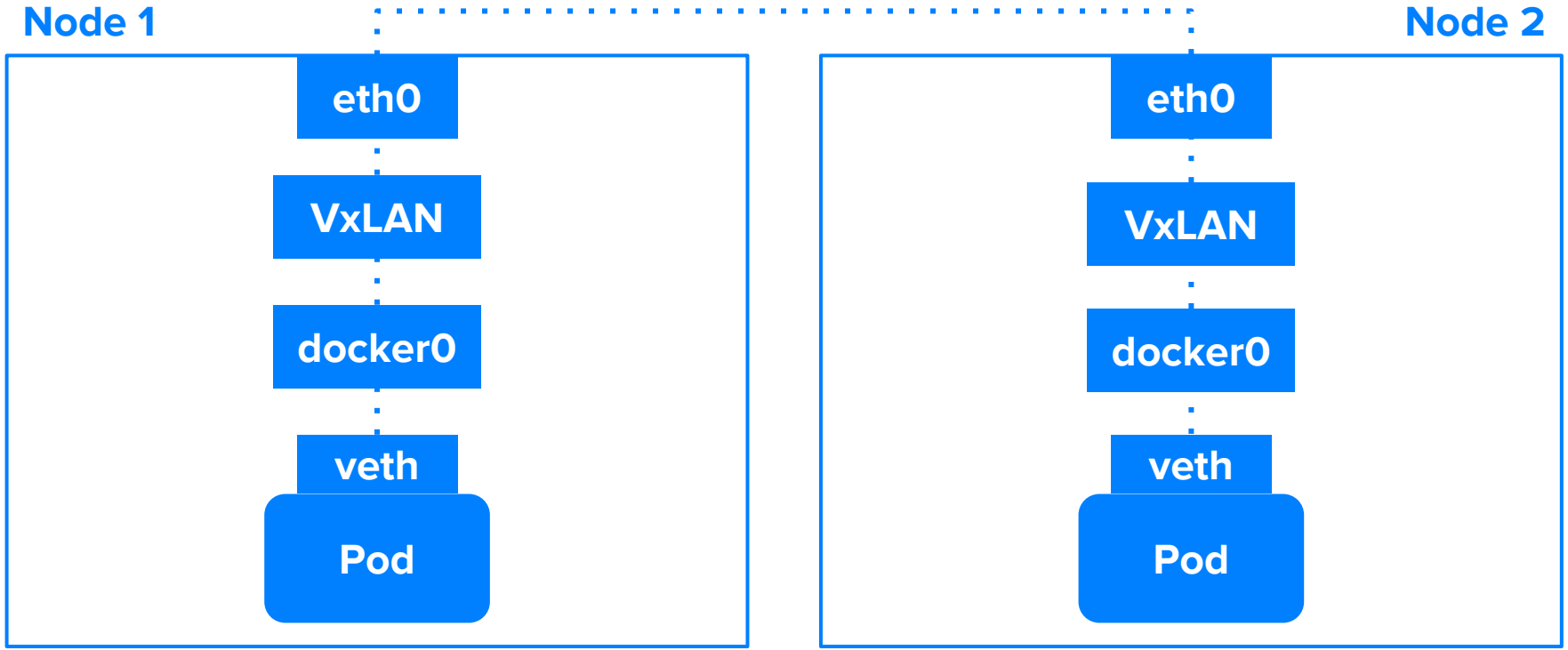
- Provided an excellent multi-tenant environment
- Production-grade scheduling
- Excellent guarantees for uptime

Challenges

- Cluster network was not fast enough for latency sensitive applications
- Networking abstractions like IP masquerading/NAT is not intuitive and difficult to monitor.

Cluster Networking

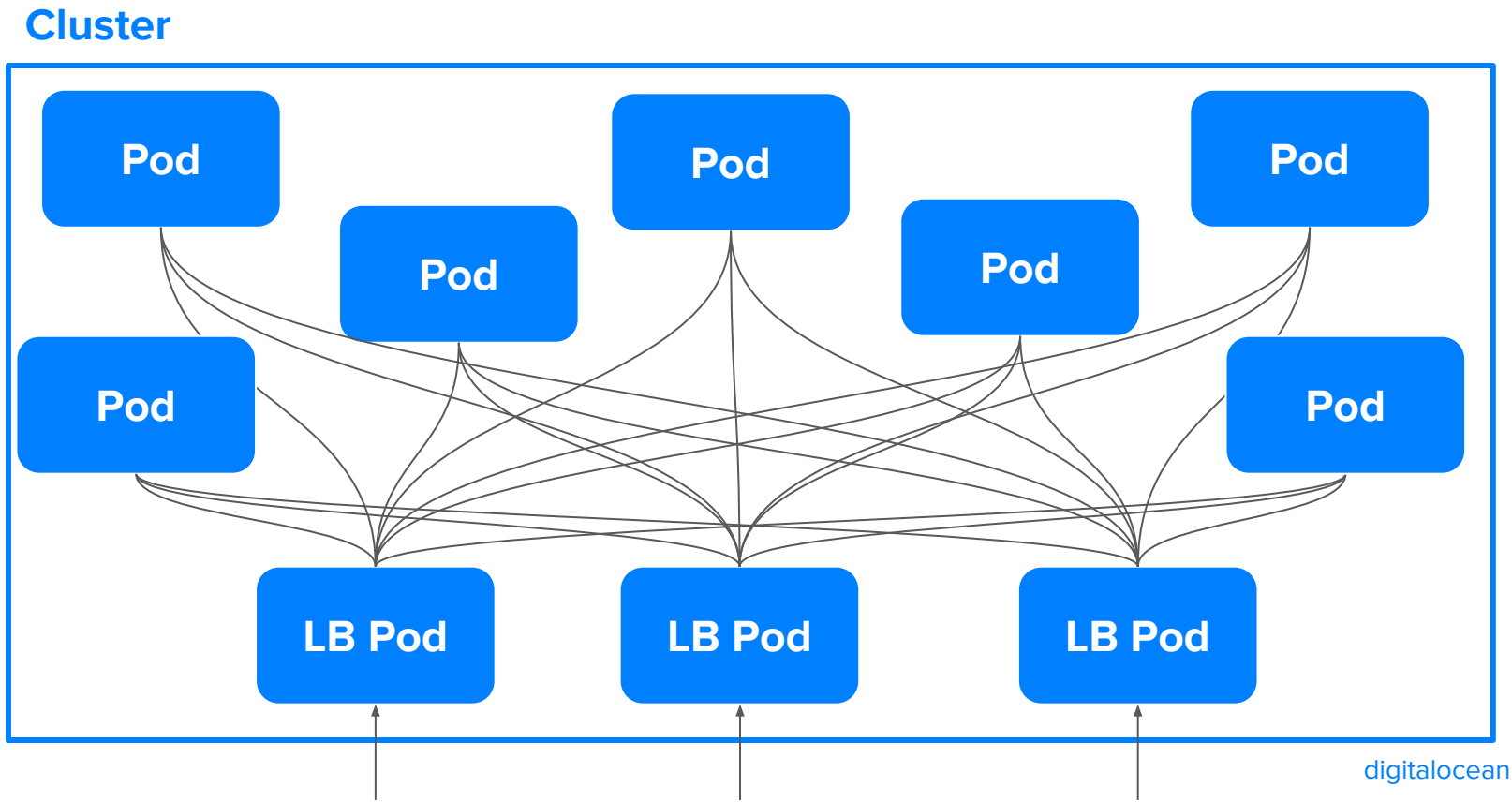
Pod Network


Node 1
eth0
VxLAN
docker0
veth
Pod
Node 2
eth0
VxLAN
docker0
veth
Pod

Pod Network

- Subnets are rigid
- Lots of packet overhead
- Requires node convergence for routes
- IP Masquerading

Ingress Controllers



Ingress Controller

- Standardized ingress traffic
- SNI based TLS pass through
- Highly Available - 3 nodes with automatic failover
- Automatic DNS failover

Labelcontroller

- Ensures enough nodes have the LB label
- Matches nodes with labels to DNS
- “Label development”

```
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: loadbalancer
  labels:
    app: loadbalancer
spec:
    …
    nodeSelector:
      loadbalancer: "true"
```

Ingress Failure

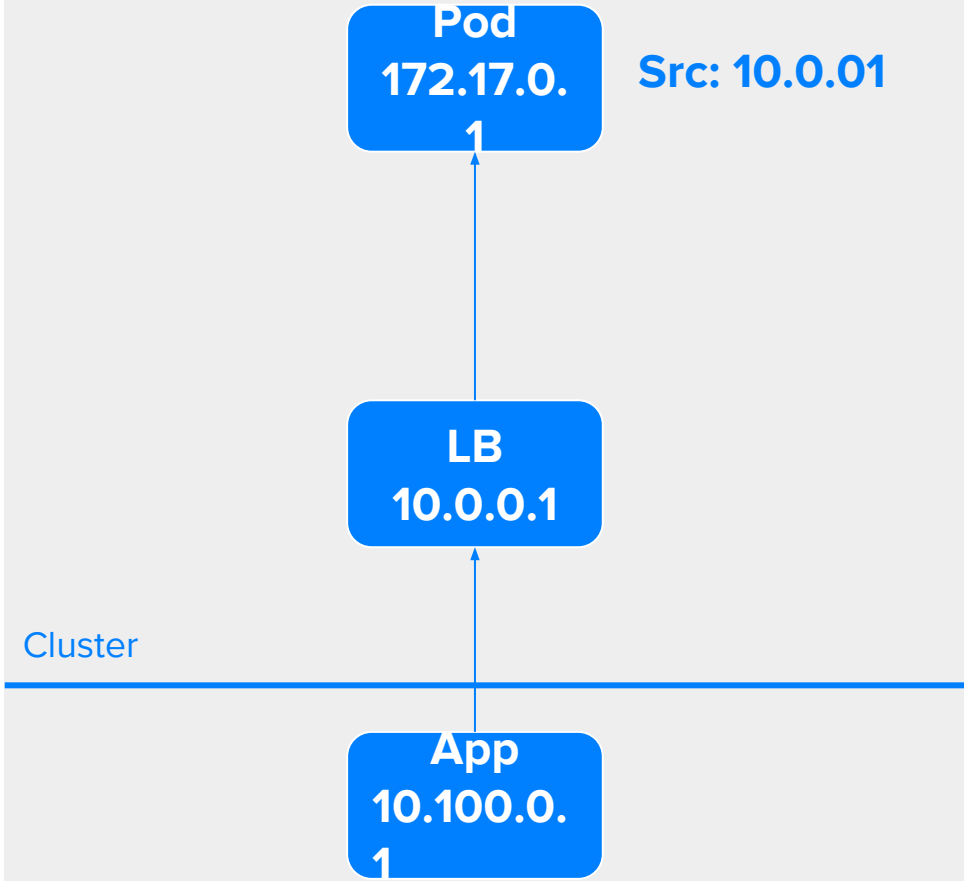
```
kubectl get pods -l app=loadbalancer

loadbalancer-lmqsb  2/2   Running   0   7d
loadbalancer-rgkpr  2/2   Running   0   7d
loadbalancer-v6gh9  1/2   Error     0   7d
```

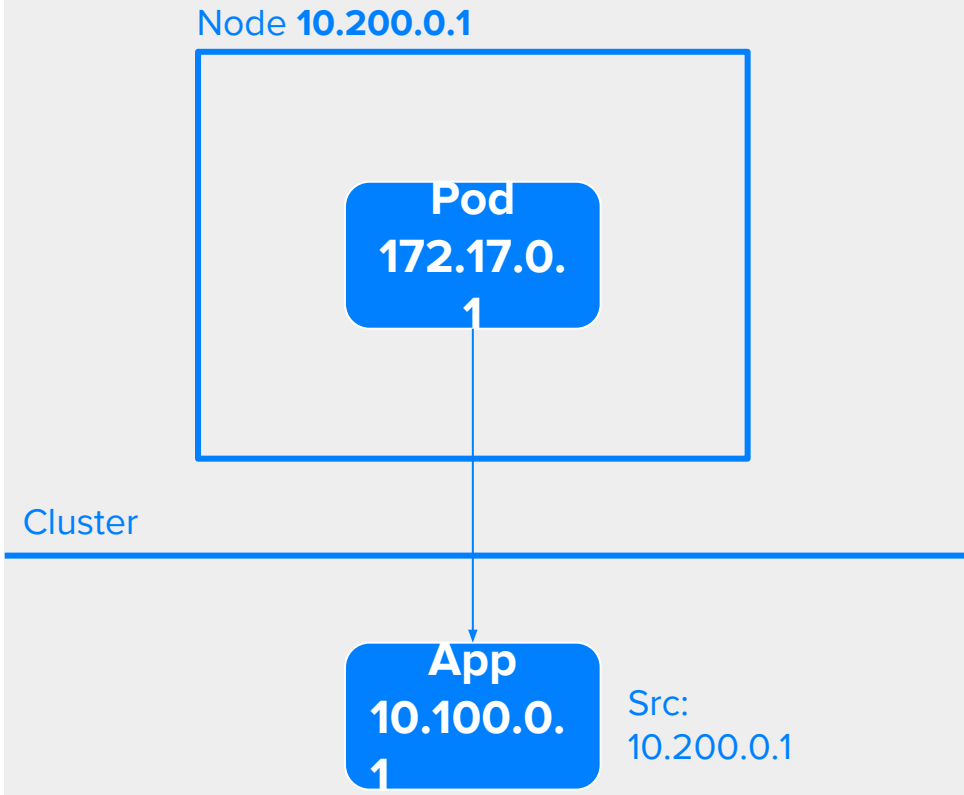
```
kubectl get nodes -l 'loadbalancer=true'

node01   Ready      node   246d   v1.10.0
node02   Ready      node   246d   v1.10.0
node03   NotReady   node   246d   v1.10.0
```

IP Masquerading & NAT



IP Masquerading & NAT



What Was Lacking?

- Too much overhead from packet encapsulation
- Operational complexity - managing node labels and DNS
- Not reliable enough at larger scale (> 100 nodes)
- Lack of visibility for src/dst IPs

What We Wanted

- Simple to operate
- Better visibility for src/dest IPs
- Flexible IP management
- Scales with the cluster
- Fast - significantly less overhead per packet

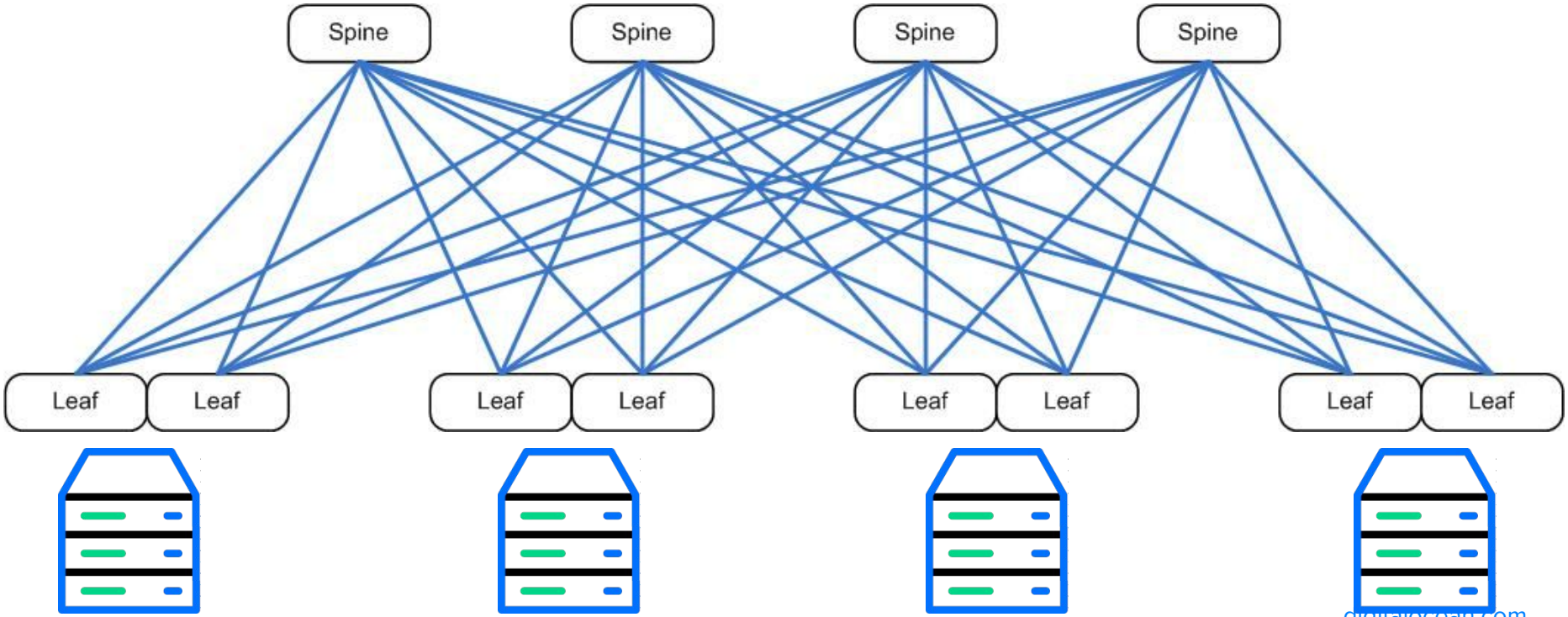
Discovery

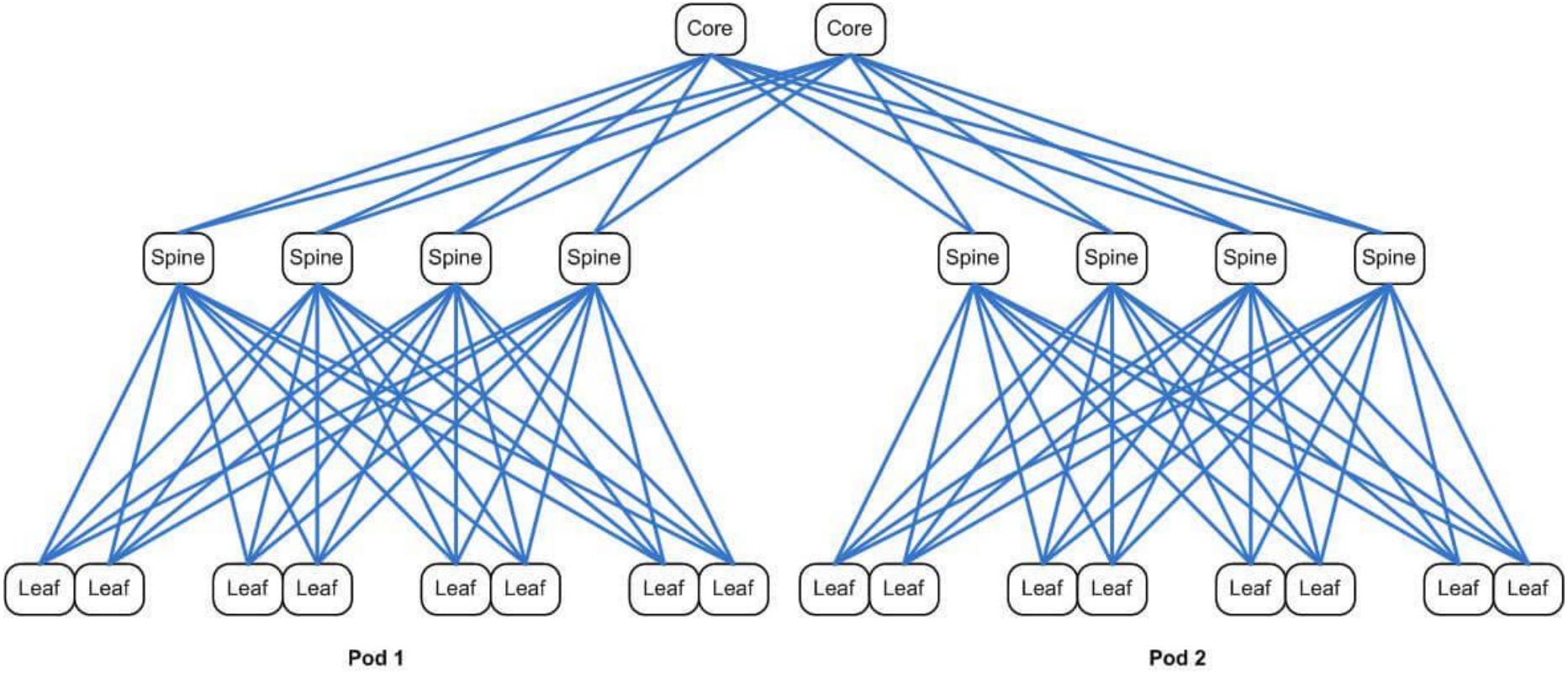
Get it right the first time!

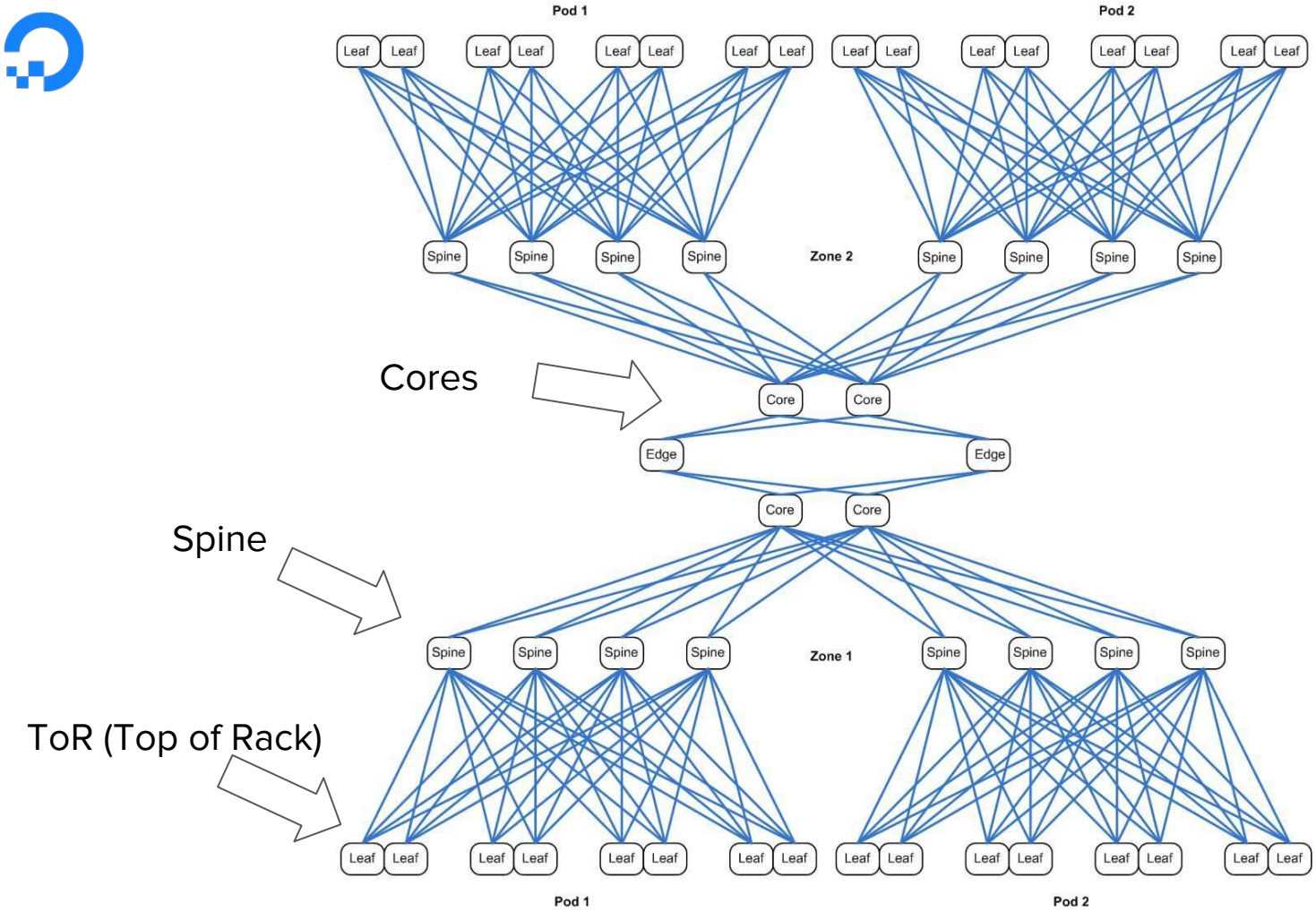
“What if you can directly connect to pod or service IPs from anywhere on our network?”

– Mac Browning

Data Center Networking & Clos Topology


Spine
Spine
Spine
Spine
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf


Core
Core
Spine
Spine
Spine
Spine
Spine
Spine
Spine
Spine
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Pod 1
Pod 2



digitalocean.com

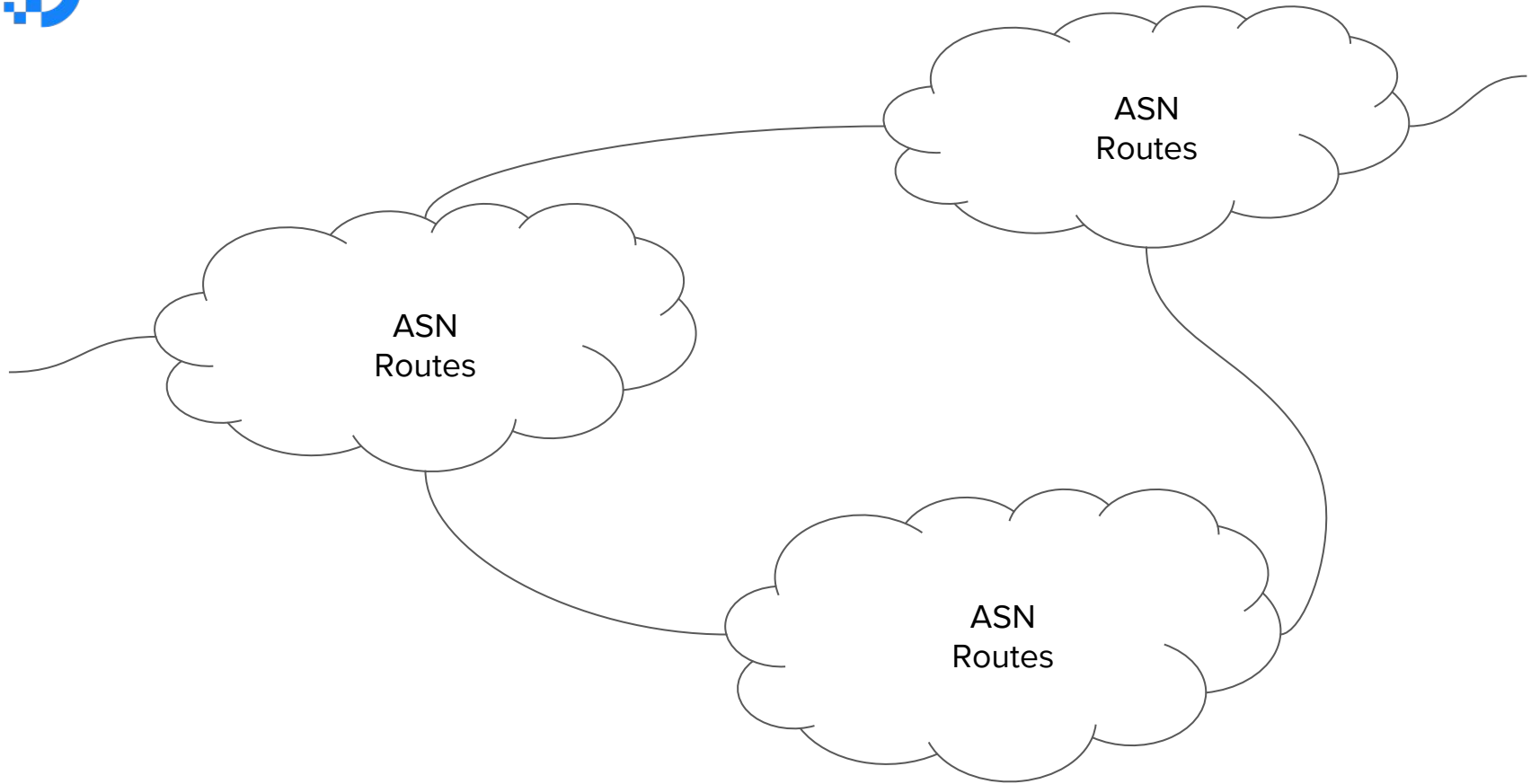
Benefits

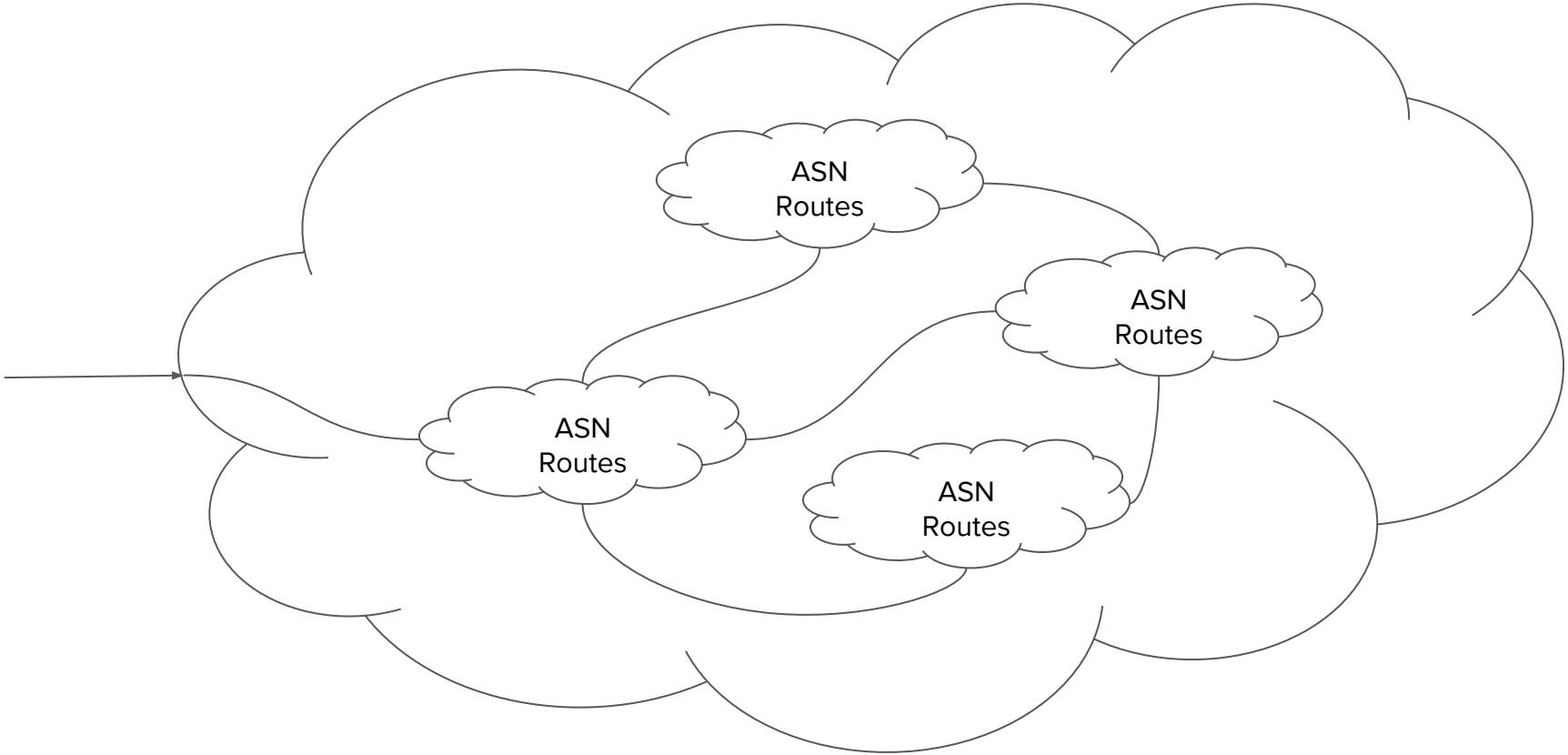
- All routing managed by datacenter switches
- No route conversion on nodes necessary
- Datacenter tech is blazingly fast!

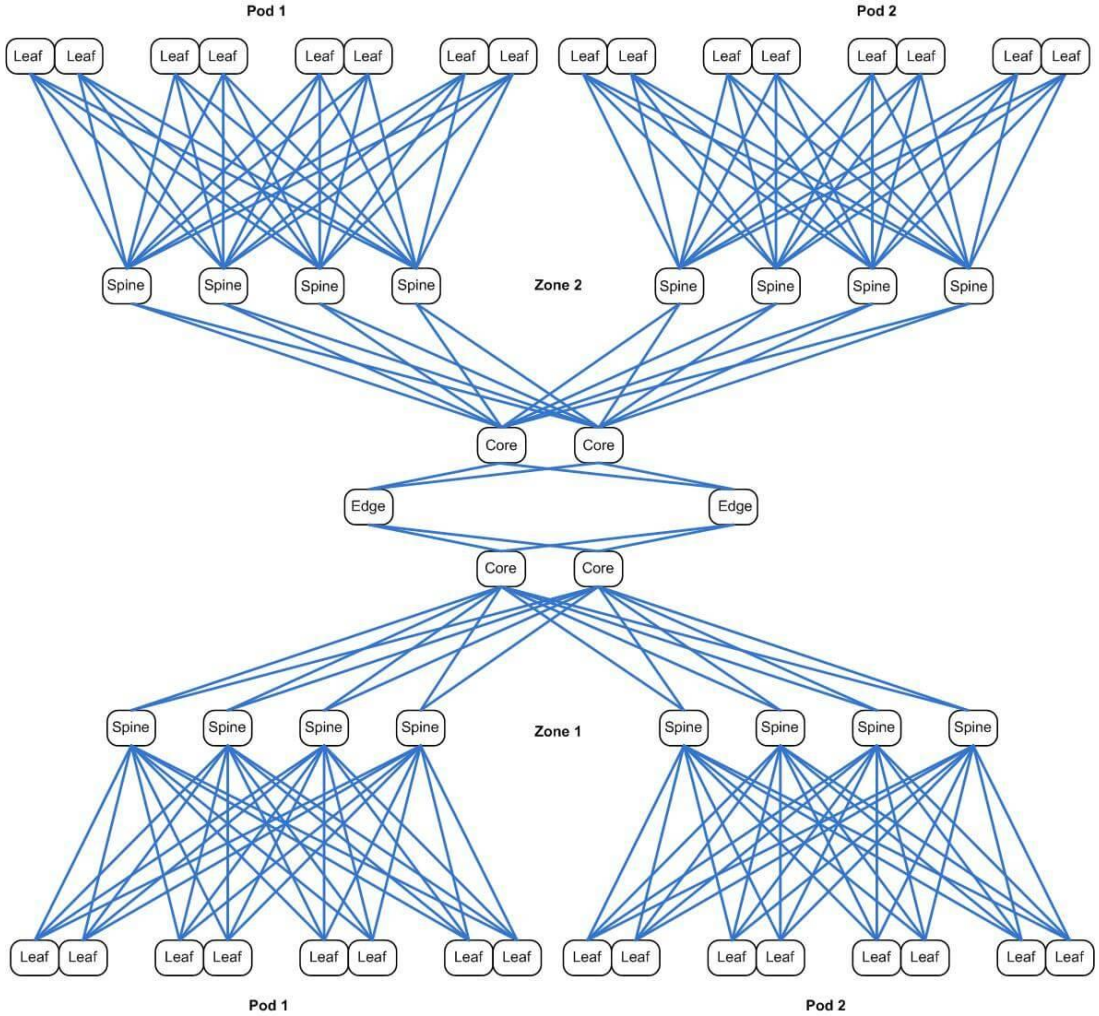
BGP

“The Border Gateway Protocol (BGP) is an inter Autonomous System routing protocol”

– IETF RFC 4271

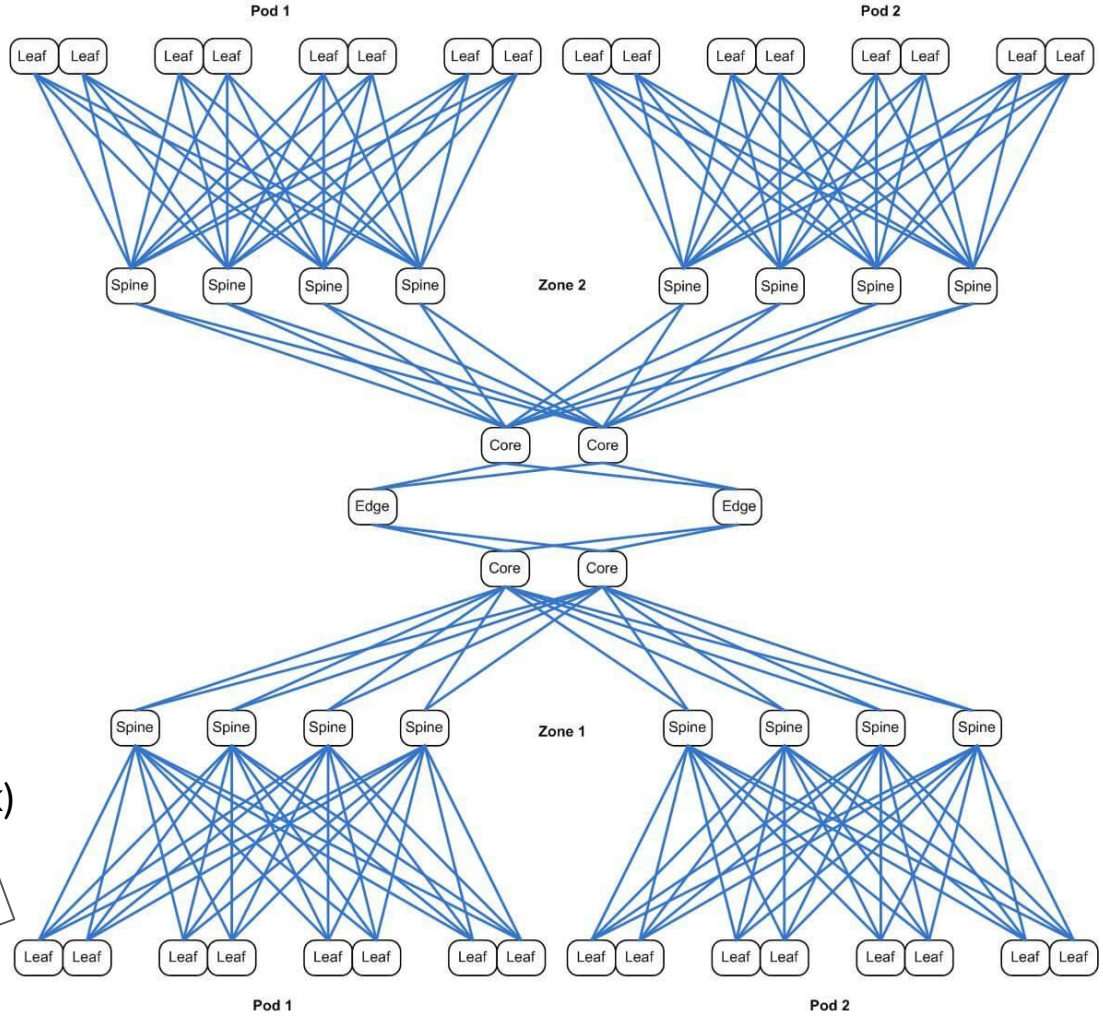

ASN
Routes
ASN
Routes
ASN
Routes


ASN
Routes
ASN
Routes
ASN
Routes
ASN
Routes


Pod 1
Pod 2
Leaf
Spine
Zone 2
Core
Edge
Zone 1
Pod 1
Pod 2

digitalocean.com

Kubernetes Clusters as BGP Autonomous Systems

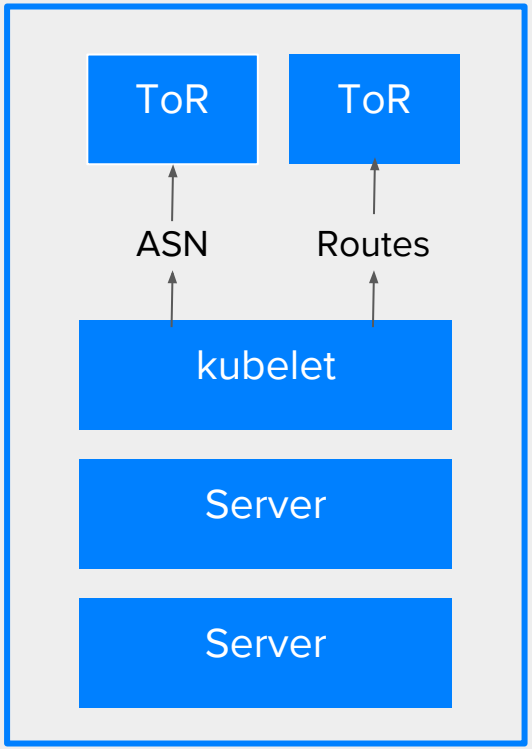


ToR (Top of Rack)

digitalocean.com

Requirements for BGP peering

- Internal ASN per cluster
- ToRs configured to peer with cluster ASN



https://github.com/cloudnativelabs/kube-router

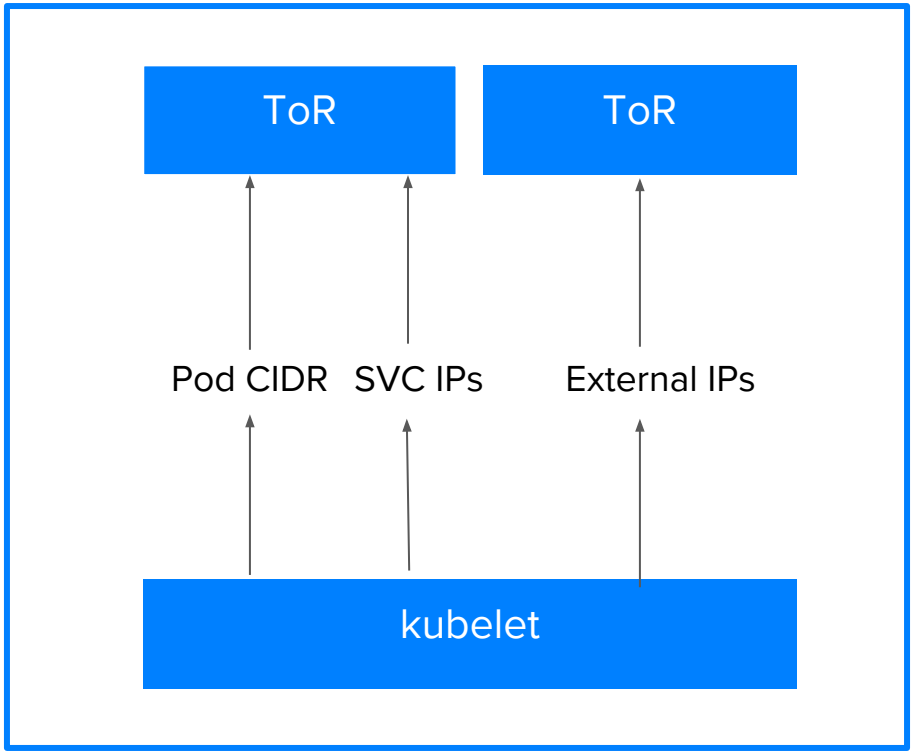
Maintained by **@murali-reddy**

- Supports iBGP / eBGP peering
- Automatic BGP peering of pod and service subnets
- Bonus: IPVS/DSR support, network policies, BGP route reflectors

kube-router

- IP and ASN to peer with
- BGP peering required on all nodes

```
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: kube-router
  labels:
    k8s-app: kube-router
spec:
    ...
    containers:
      args:
        - --peer-ips=<top-of-rack-ip>
        - --peer-asns=<top-of-rack-asn>
```


ToR
ToR
Pod CIDR
SVC IPs
External IPs
kubelet

Anycast IPs

- automatically accessible from any internal network
- automatic health checking capabilities
- managed by service owners

```
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
    clusterIP: 172.25.7.35
    ...
```

Use cases

Globally Distributed Kubernetes Services

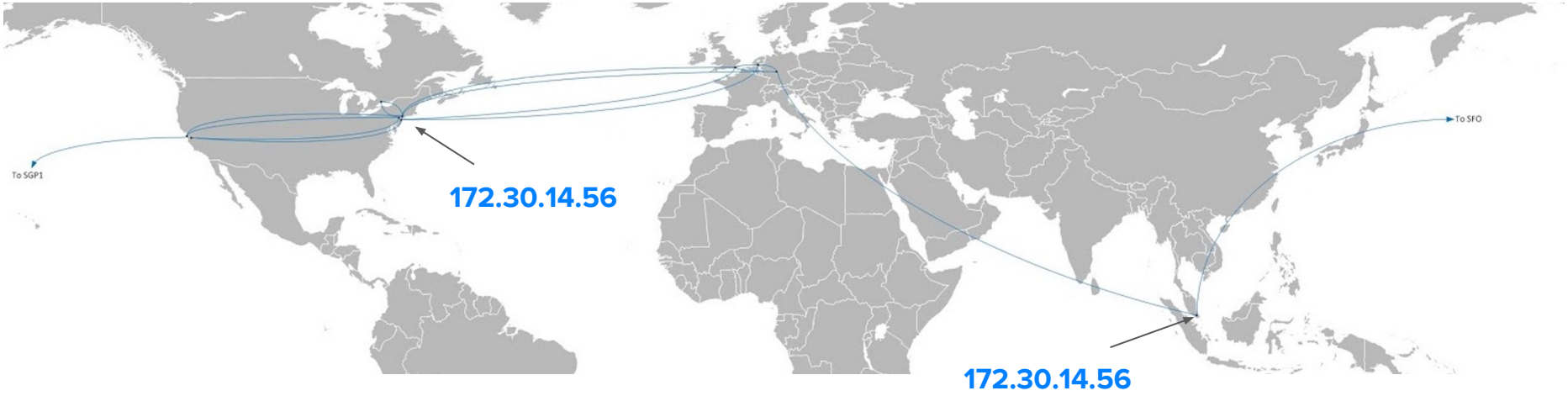
```
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
    region: nyc3
spec:
    clusterIP: 172.25.7.35
    ...
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
    region: sgp1
spec:
    clusterIP: 172.25.152.38
    ...
```


To SGP1
To SFO
172.25.7.35
172.25.152.38

Globally Distributed Services with External IPs

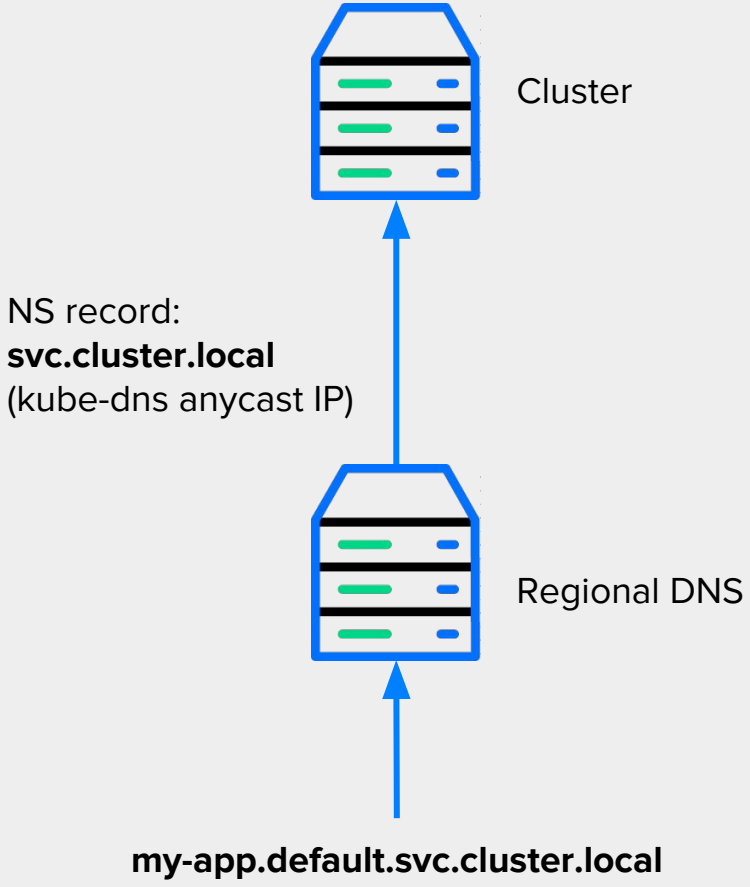
Shared IPs across all data centers for external IPs

```
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
    externalIPs:
      - 172.30.14.56
    ...
```


To SGP1
172.30.14.56
To SFO
172.30.14.56

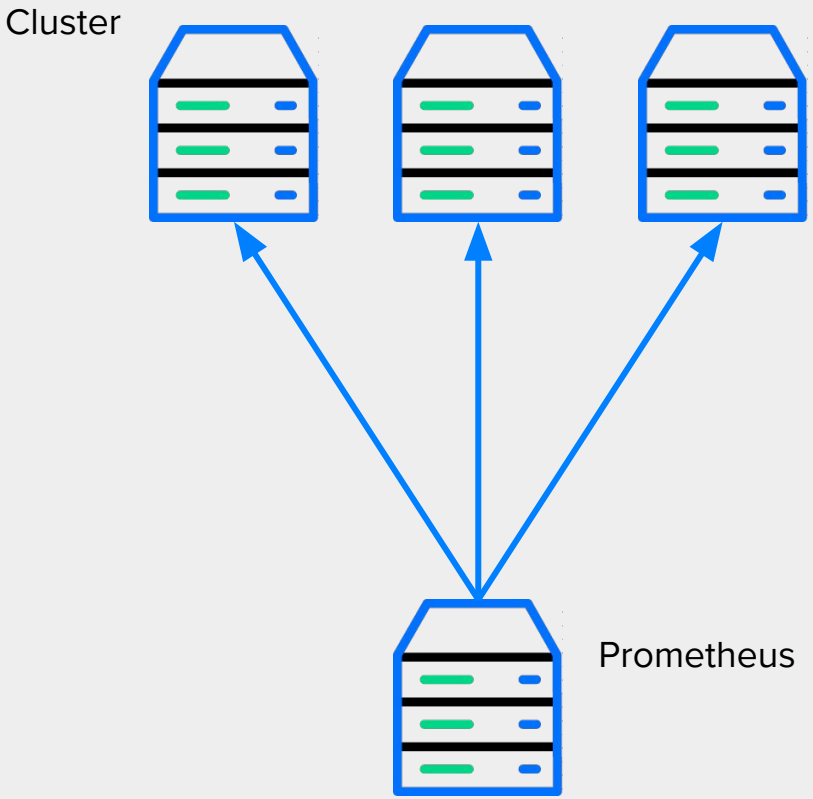
DNS

NS records to delegate cluster domains from regional DNS servers

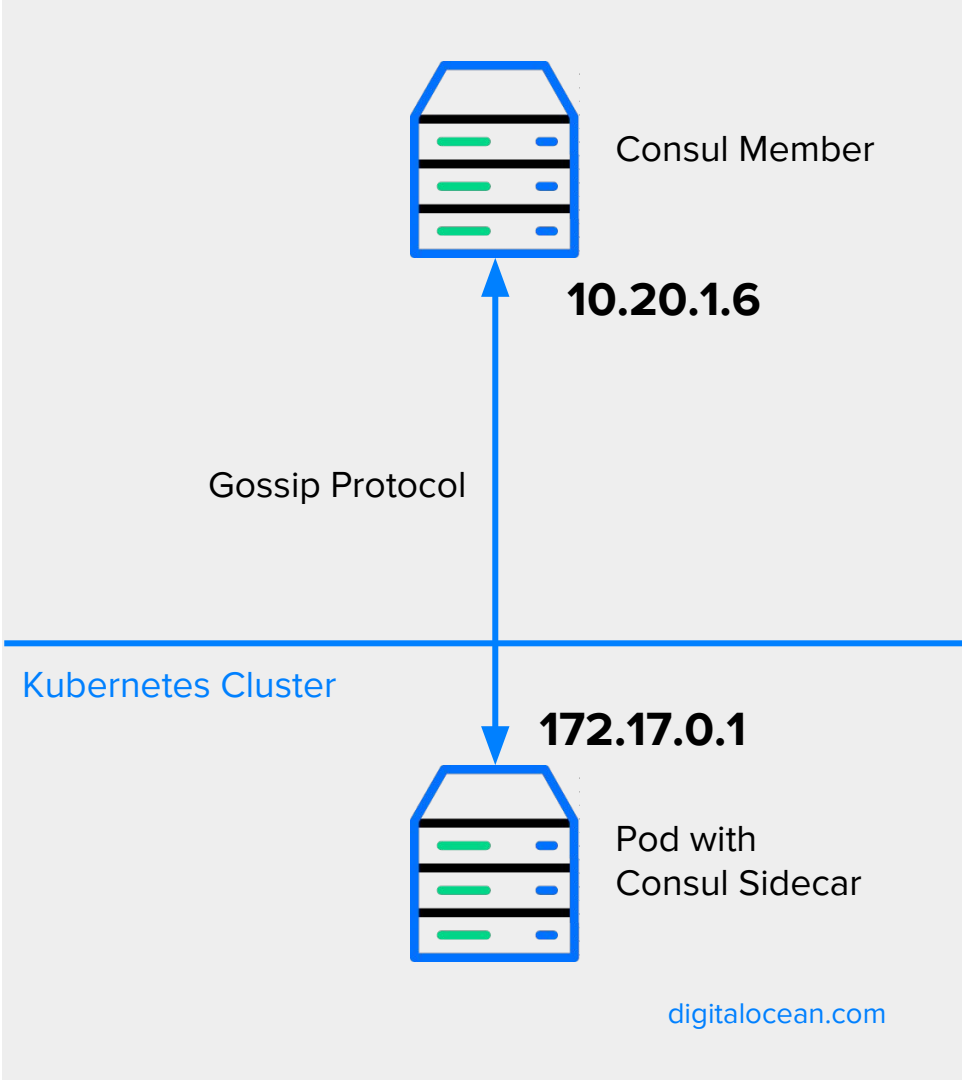


Prometheus Scraping

Metrics scrapping from an external prometheus cluster



Better Integration with other Systems

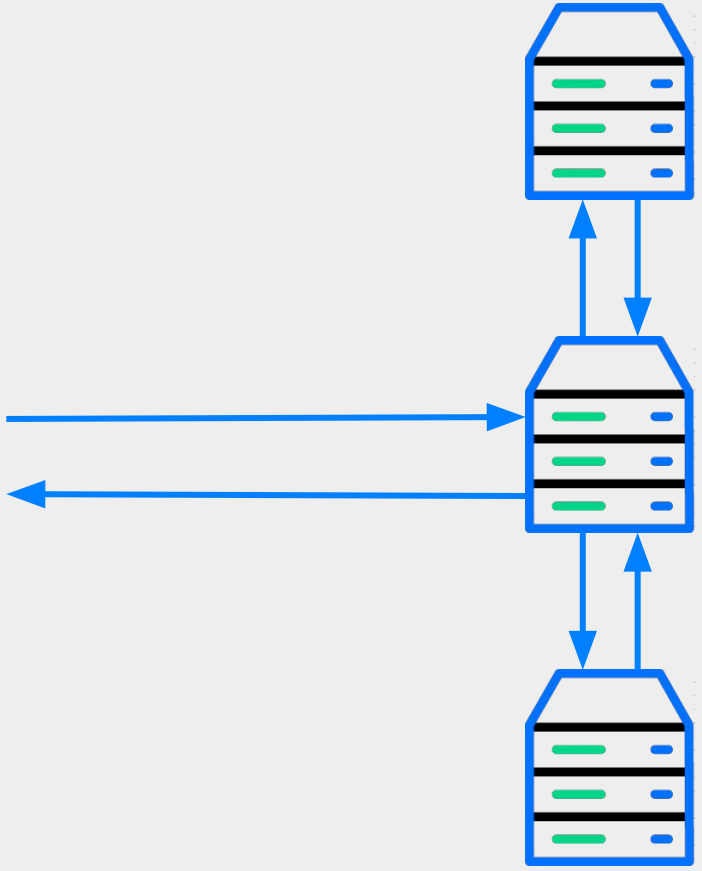


Re-architecting Ingress Controllers

- Daemonsets ➔ Deployment
- Node IPs ➔ Service IP for wildcard DNS
- No need for Labelcontroller

Challenges & Trade Offs

The Proxy Mesh

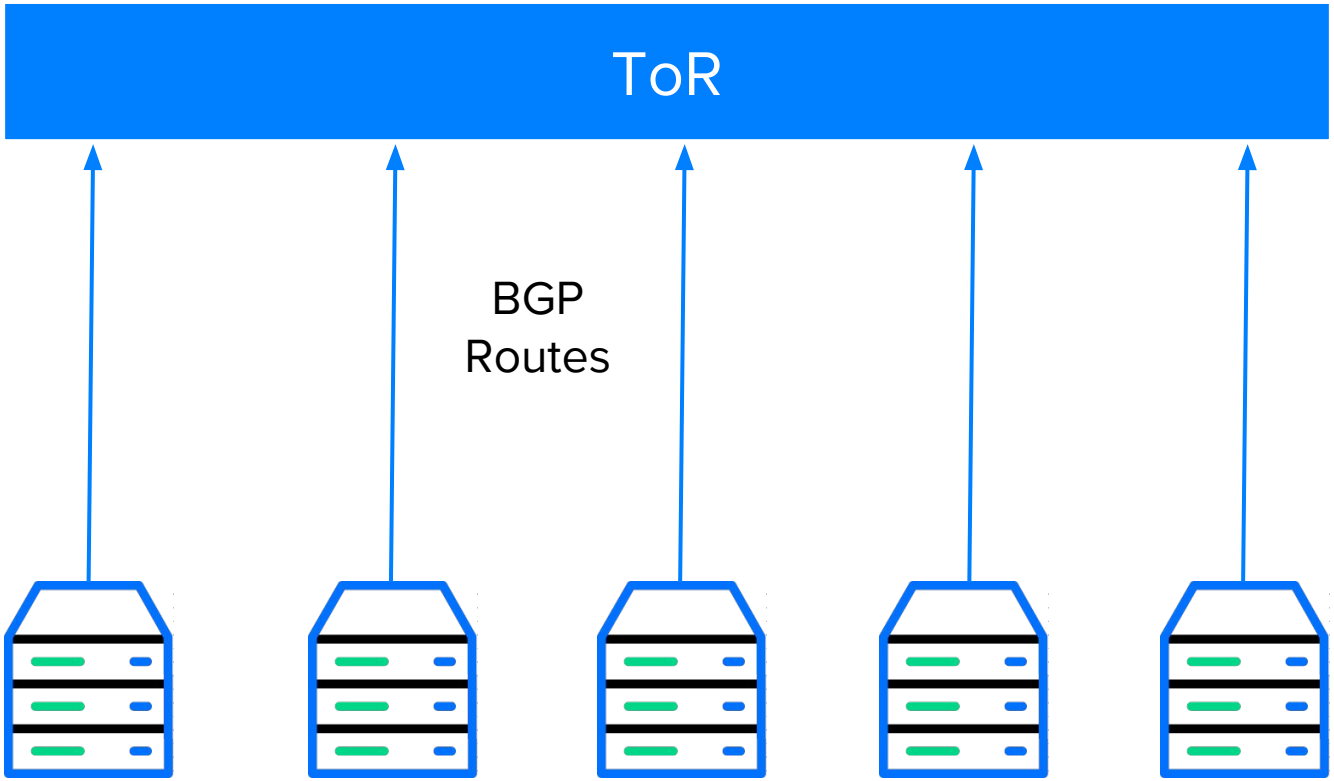


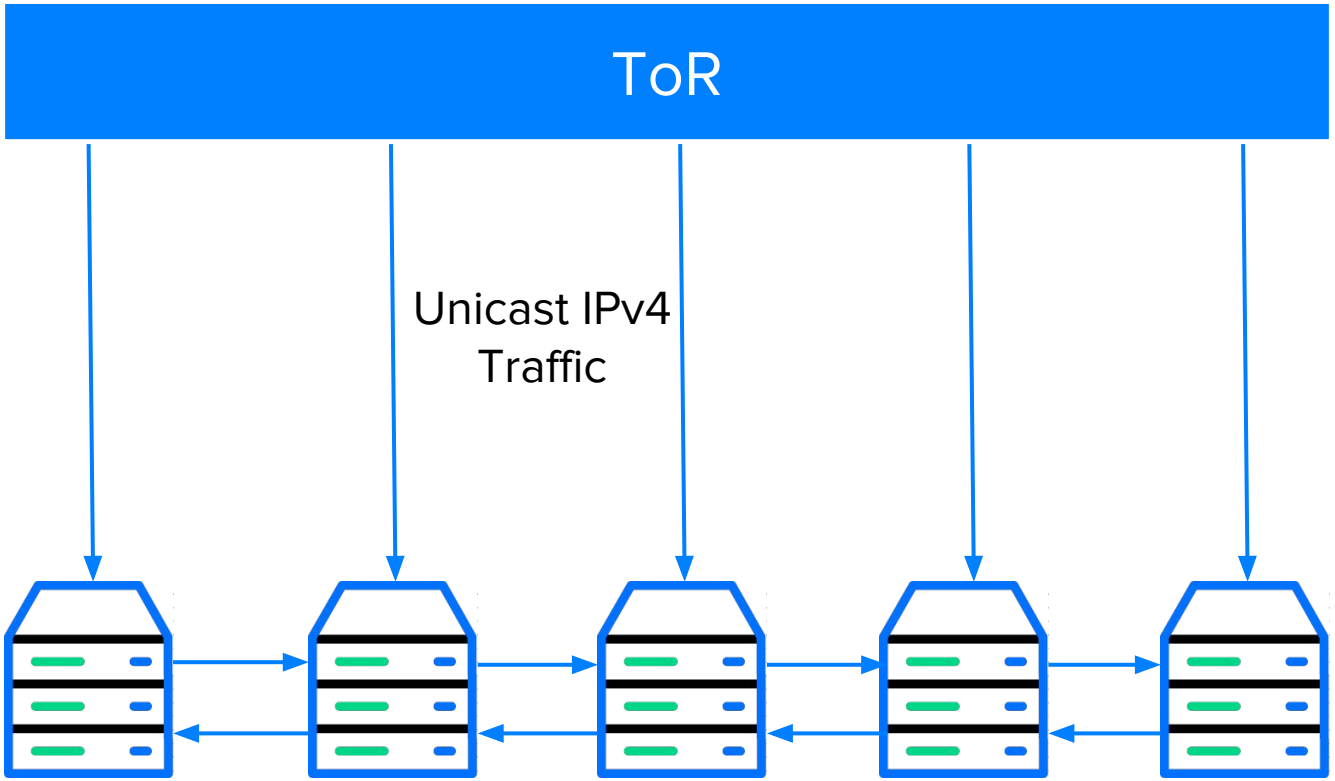
Uniqueness Requirements for IPs

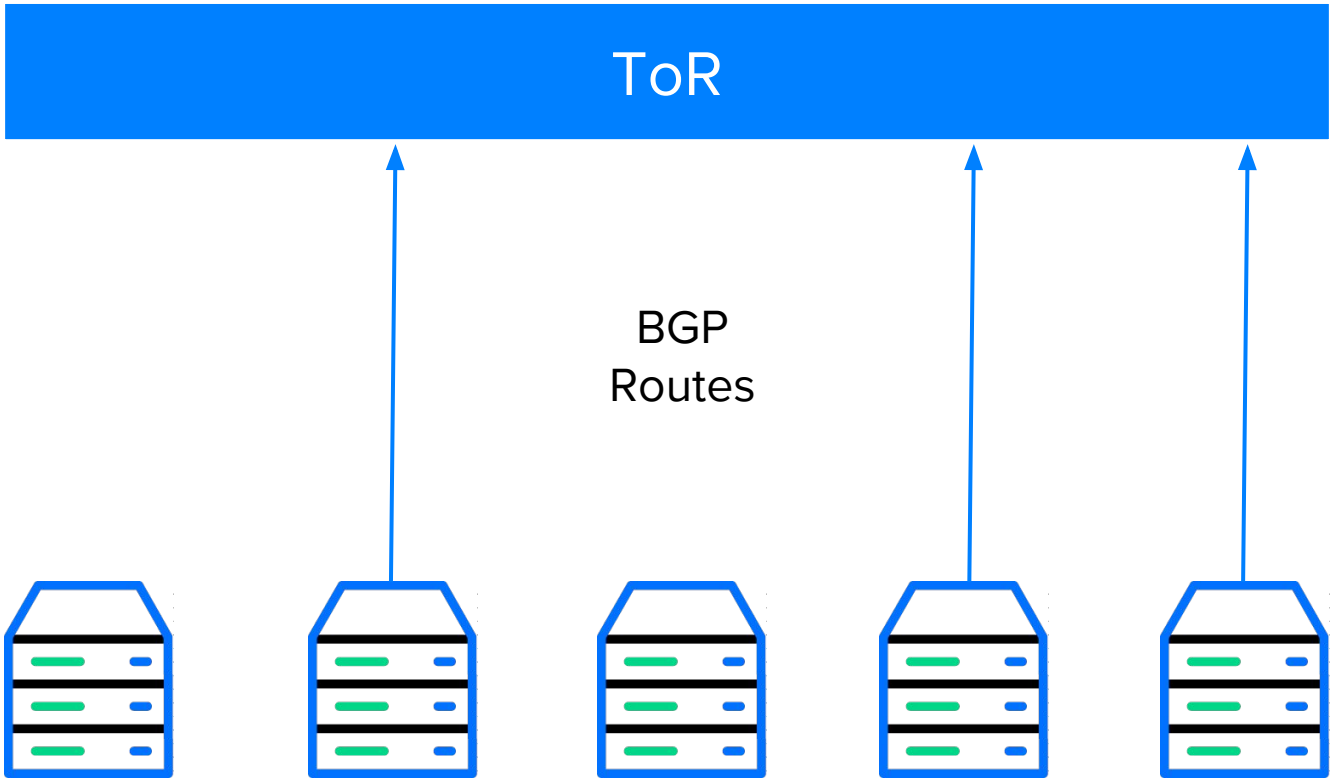
- Pod and Service IPs must be unique across all data centers and environments
- Routes can conflict in local development environments

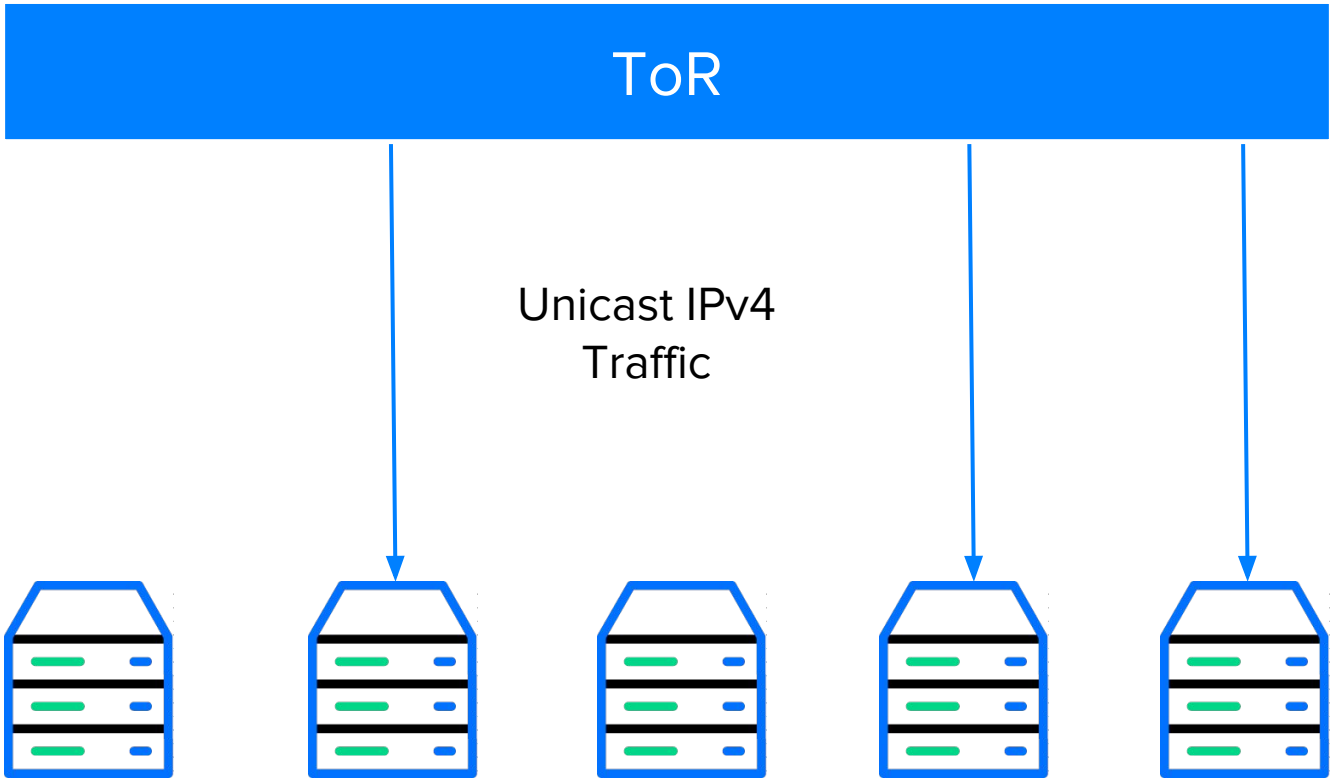
BGP Route Convergence & Rebalancing

Future Considerations & Improvements


ToR
BGP
Routes

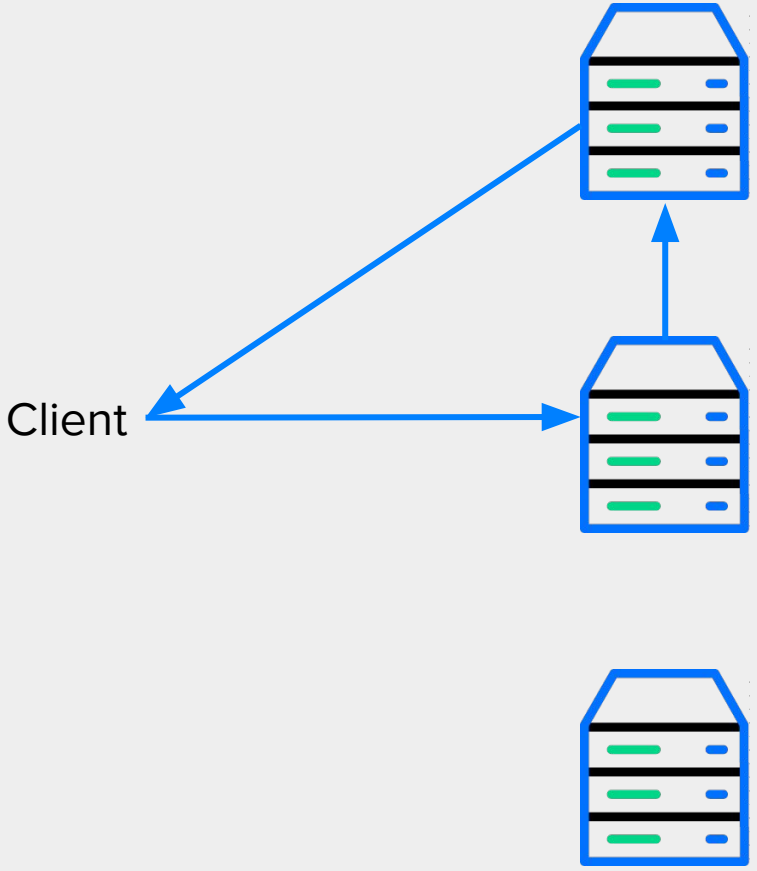

ToR
Unicast IPv4
Traffic


ToR
BGP
Routes


ToR
Unicast IPv4
Traffic

Direct Server Return (DSR)

- Should be configurable based on application needs



Adopting IPVS

- Faster convergence of endpoints
- Faster kernel level loadbalancing than iptables proxy
- Advanced loadbalancing schedulers (round robin, least connections, source/destination hashing, etc)

Exposing Kubernetes Services to the Public Internet

```
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
    externalIPs:
      - 1.2.3.4
    ...
```


To SGP1
1.2.3.4
To SFO
1.2.3.4

Container Networking on IPv6!

Recap!

BGP + Kubernetes

Additional Topics

- BGP + Kubernetes on the Cloud!
- BGP route reflectors
- Equal Cost Multi Path (ECMP)
- Security Considerations of using BGP

Thank you!

 @kimandrewsy  @andrewsykim