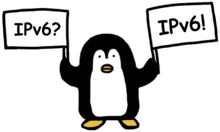
IPv6 only Kubernetes - UK IPv6 Council

ungleich



Nico Schottelius

<2020-12-07 Mon>

A long long time ago, . . .[1]

- Manually configured single servers ("the old days")
- Automatically configured / mass installation servers
- Virtualisation (Qemu/KVM, VMWare, etc.)
- Containerisation (Docker, Docker Swarm, Docker Compose)
- Kubernetes (large scale container distribution)

[1] (*) Imagine listening to Don McLean - American Pie

Motivation

- IPv6 only infrastructure at ungleich
- No internal IPv4 routing (no RFC1918, no global routes)
  - Only between upstreams and PoPs
- All workload is IPv6 only
- Workload is moving towards Kubernetes ("standardisation")

Servers, Containers, IPv6?

- How is IPv6 related to any of this?
- Network design is a major part of Kubernetes

Networking in Kubernetes

- Abstracted by CNI
  - Container network interface
- Long list of CNI plugins
  - `https://kubernetes.io/docs/concepts/cluster-administration/networking/`
- Example CNI
  - Bridge, Calico, Cilium, Flannel, Multus, Weave, ...
- Cloud/Vendor
  - Azure, AWS, CNI-Genie

CNI functionality

- "The whole networking"
- Tunneling (IP-IP, VLAN, VXLAN, . . .)
- Routing (iBGP, eBGP, Mesh)
- IPAM

Not in CNI

- DNS is served by CoreDNS (cluster builtin)
- IPAM can be part of k8s control plane
- kube-proxy: forwards requests for another node

IPv6 support in K8S

- IPv6 only support has recently been added
- Dual stack support ("multiple IPs per pod") even more recent
- Kubernetes 1.22 or higher is safe

IPv6 support in CNI

- CNIs are rather flexible
  - Even without k8s support, calico supported IPv6 already
- IPv6 compatible CNIs
  - Calico (production use)
  - Cilium (development use)
  - Probably others

Setting up k8s with IPv6

- General approach
  - Think about your network design first
  - (i,e)BGP?
  - VXLAN/IP-IP/VLAN
  - eBPF required?
- Check CNIs that match your design
  - Adjust your design as the selected CNI might not support it

ungleich IPv6 only Kubernetes example

- CNI: calico
  - No encapsulation
  - BGP with internal routers
  - nodeToNodeMeshEnabled: true
  - keepOriginalNextHop: true
- DNS Domain: c2.k8s.ooo
- Pod network: 2a0a:e5c0:10:2::/64
- Service network: 2a0a:e5c0:10:3::/108
- Rest: standard calico/k8s

IPv6 only k8s cluster: Service DNS

- DNS: every cluster has a domain
  - Usually cluster.local
  - Managed by internal coredns
- Is appended to cluster services:
  - [servicename].[namespace].svc.[clusterdomain]
- What if services are IPv6 based?
  - Globally reachable, automatic DNS!

IPv6 only k8s cluster in practice: DNS

- What if I had a short domain like... k8s.ooo?
- Use the cluster domain cX.k8s.ooo
- Services are automatically reachable, world wide!
- [servicename].default.c2.k8s.ooo = ungleich k8s test cluster

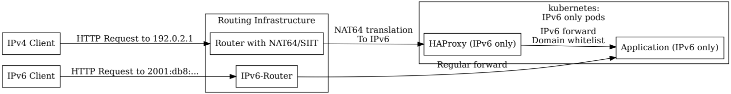
IPv6 only DNS

- Resolution only possible from IPv6 capable resolvers!
- Client can be IPv4 based for resolving, but resolvers needs to support IPv6
- Standard glue records
- Can be upgraded with NAT64 as shown below

```
c2                            NS       kube-dns.kube-system.svc.c2
kube-dns.kube-system.svc.c2 AAAA     2a0a:e5c0:13:e2::a
kube-dns.kube-system.svc.c2 A        147.78.194.9
```

IPv6 only workload in practice via proxies

- Workload is IPv6 only
- How do we handle IPv4 traffic?
- Use HTTP(s) proxies (existing concept)
- In k8s: IPv6 based proxy made reachable via NAT64



k8s: IPv4 Proxy in IPv6 only k8s

K8S concepts & IPv6

- K8S has an ingress concept
  - Ingress is usually an nginx distributing HTTP/HTTPS traffic
  - Services are by default "internal"
  - This is different in the IPv6 world
- K8s pods are conceptually "private"
  - They are not necessarily in the IPv6 (only) world
- Network policies can help
- Some existing k8s concepts have to be questioned

Summary

- Documentation for IPv6 only not yet up to speed
- IPv6 only kubernetes clusters work fine nowadays
- Multiple CNIs support IPv6
- Standard methods for handling IPv4
  - NAT64, SIIT, HTTP Proxies

More information

- Matrix-Chat:
  - #ipv6:ungleich.ch
    - aka `https://IPv6.Chat`
  - #kubernetes:ungleich.ch
- `https://redmine.ungleich.ch/projects/open-infrastructure/wiki/The_ungleich_kubernetes_infrastructure`