IPv6-only network for containers orchestration
Summary

- What’s a container?
- How it can improve your company workflow?
- Networking of containers
- What’s Kubernetes?
- How to get an IPv6-only network?
- How to get an IPv6-only Kubernetes cluster?
What’s a container?

Containers embed Application & Dependencies in a tiny, reproducible box, isolated by the power of the Linux Kernel (cgroups, namespaces, SELinux).

Unless virtual-machines that embed a full operating system isolated by Hypervisor capacities.
#1st Demo

Launch a Wordpress website (PHP + MySQL) using Docker Compose.

```
version: '3.3'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: RANDOM
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: AGoodPassword
  wordpress:
    depends_on:
    - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: AGoodPassword
    volumes:
      db_data:
```
How it can improve your company workflow?
Containers Networking

By default containers are in a different IP space, behind a Bridge.

You can access your containers by exposing them on the host side (using IPTables).

You can use IPv6 too, by defining a fixed-cidr-v6 for your docker host.
What’s Kubernetes?

Docker is great, but what if you can deploy containers in a cluster of servers with:

- Self healing
- Auto-scaling (node & app)
- Service discovery
- Cron execution
- Storage orchestration
- Load-balancing
k8s Master
k8s Worker
Calico

Secure networking for the cloud native era

Simple

Scalable

Secure

Hundreds of enterprises trust Calico to connect and secure their cloud
(Orange, Github, Yahoo Japan, USA)
#2nd Demo

Container security on IPv6-only Kubernetes cluster.

Stars Policy Demo

The included demo sets up a frontend and backend service, as well as a client network policy on each service.

```yaml
apiVersion: extensions/v1beta1
customPolicy: NetworkPolicy
metadata:
  name: allow-mysql
  namespace: myapp
spec:
  ingress:
    - from:
      - podSelector:
        matchLabels:
          app: webapp
        ports:
          - port: 3306
            protocol: TCP
        podSelector:
          matchLabels:
            k8s-app: dbapp
```
How to get an IPv6-only network?

- You want to talk to IPv4-only server? You need NAT64
- You want to automatically set the `resolv.conf`? You need to use rdnsedd
- You want expose services to IPv4 client? You need dual-stack Proxy
- You want to secure things like NAT “does”? You need a firewall!
How to get an IPv6-only Kubernetes cluster?

- Kubernetes components are IPv6-ready
- ETCD quorum in IPv6
- Kubectl talk to IPv6 endpoints
- Kubernetes Dashboard in IPv6...
How to get an IPv6-only Kubernetes cluster?

- If you use Calico, you can use **dual stack** or **IPv6-only** on the container side.

- You can create **different IpPools** for your needs (with **calicoctl**)

- You can use **external BGP Peer** to expose publicly parts of your cluster.
How to get an IPv6-only Kubernetes cluster?

- For Kubernetes Services,
  - You can use an IPv6 pool for your internal services but this enable NAT. (ip6tables)
  - You should use `ClusterIP: None` services type to switch to DNS round robin and avoid NAT.
How to get an IPv6-only Kubernetes cluster?

- Our Setup at Nautilie.nc,
  
  Jool for NAT64, 
  Kubernetes in IPv6-only 
  4 Public Nginx ingress 
  4 Public Kube-DNS 
  GlusterFS cluster 
  BGP Announces to our core network.
More information about IPv6 with Kubernetes on my blog:

OpsNotice.xyz/kubernetes-ipv6-only