

# IPv6 in Kubernetes & OpenShift

IPv6 Council #13

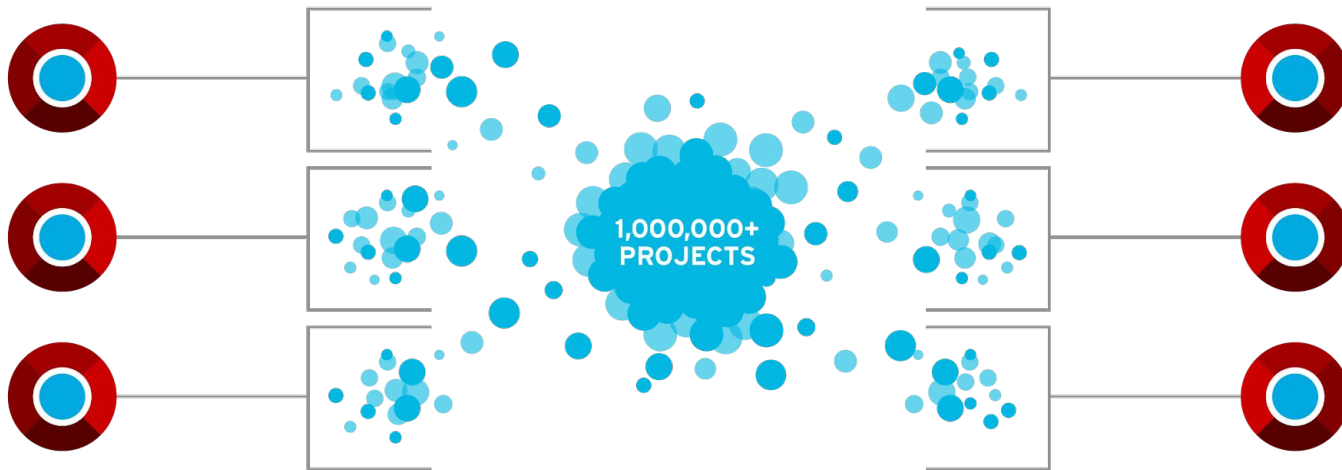
April 1st, 2022

Pieter Lewyllie - Solution Architect, Red Hat  
- IPv6 Council Be Co-Chair

# Disclaimer

These slides represent my personal views. Not all of the following slides have been vetted by Red Hat.

# Product development model



## Participate

We participate in and create community-powered upstream projects.

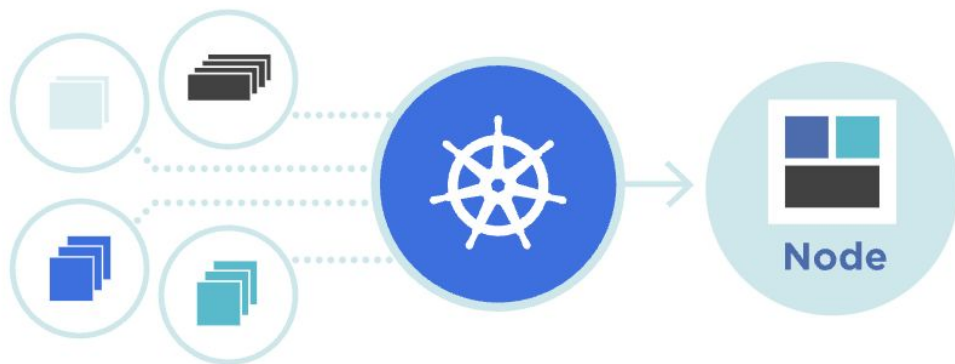
## Integrate

We integrate upstream projects, fostering open community platforms.

## Stabilize

We commercialize these platforms together with a rich ecosystem of services and certifications.

**Kubernetes**, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.



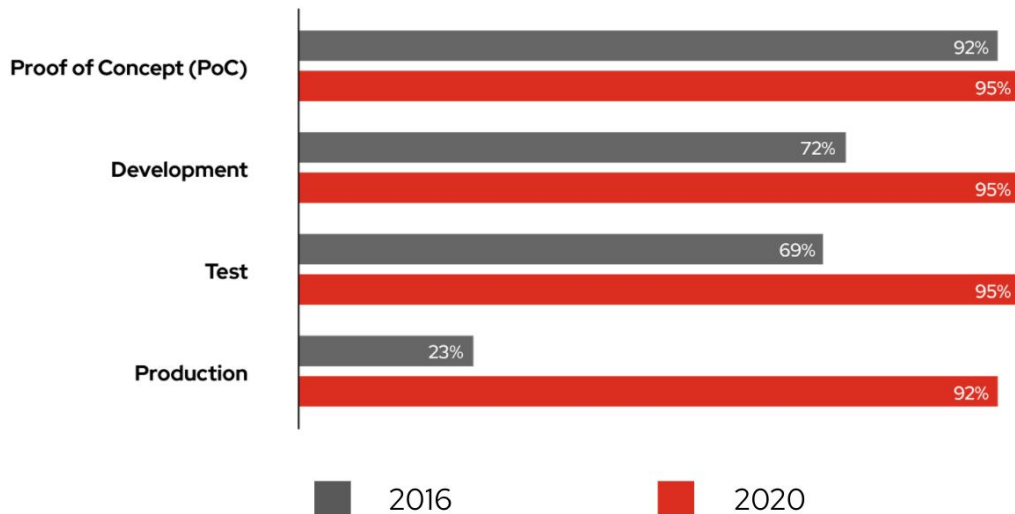
# 91%

ARE USING KUBERNETES

Businesses are using Kubernetes to

- Drive automation
- Improve efficiencies
- Increase application agility

## Container adoption is accelerating



# IPv6 & containers...

# APNIC

 **kubernetes**

[Documentation](#) [Blog](#) [Partners](#) [Community](#) [Case Studies](#) [English](#) ^

Concepts

 **kubernetes**

[Documentation](#) [Kubernetes Blog](#) [Training](#) [Partners](#) [Community](#) [Case Studies](#) [Versions](#) ▾ [English](#) ▾

## IPv6 and containers a horror story

By [Matt Pal...](#)

🔍 Search

[Home](#)  
[Getting started](#)  
[Concepts](#)

[Kubernetes Documentation](#) / [Concepts](#)  
/ [Services, Load Balancing, and Networking](#) / [IPv4/IPv6 dual-stack](#)

## IPv4/IPv6 dual-stack

FEATURE STATE: [Kubernetes v1.23](#) [stable]

# Kubernetes Warms Up to IPv6

25 Feb 2019 11:55am, by [Mary Branscombe](#)

<https://blog.apnic.net/2018/03/22/ipv6-and-containers-a-horror-story/>

<https://thenewstack.io/kubernetes-warms-up-to-ipv6/>

[https://docs.openshift.com/container-platform/4.2/release\\_notes/ocp-4-2-release-notes.html](https://docs.openshift.com/container-platform/4.2/release_notes/ocp-4-2-release-notes.html)

<https://kubernetes.io/docs/concepts/services-networking/dual-stack/>

GIN >

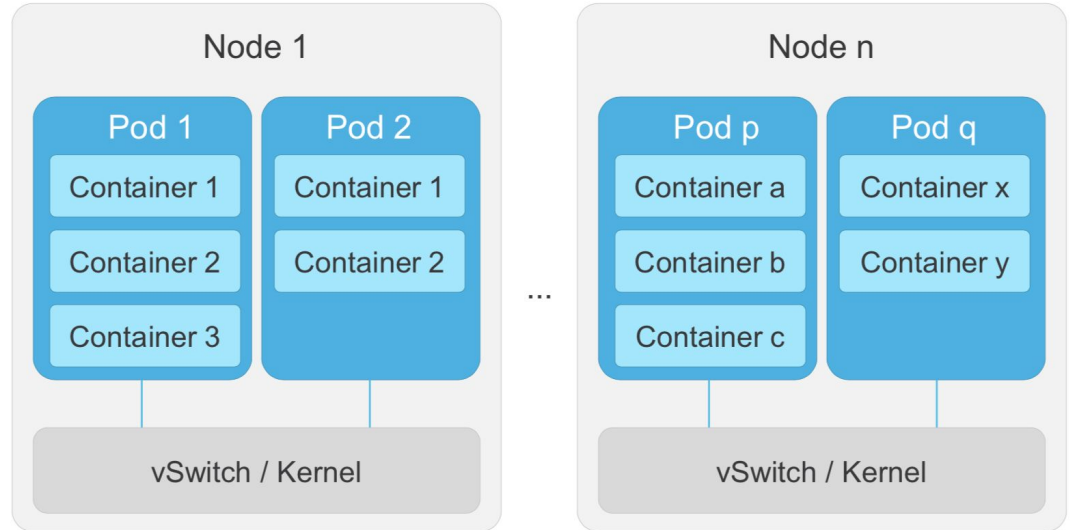
[an edit](#)

# Why IPv6 in Kubernetes?

- ✓ Cleaner design
- ✓ Easier troubleshooting
- ✓ Not easy to find remaining (private!) IPv4 space in organization
  - ✓ Multi cluster
  - ✓ VNFs: Mobile packet core, 5G...

# Nodes, Pods, Containers

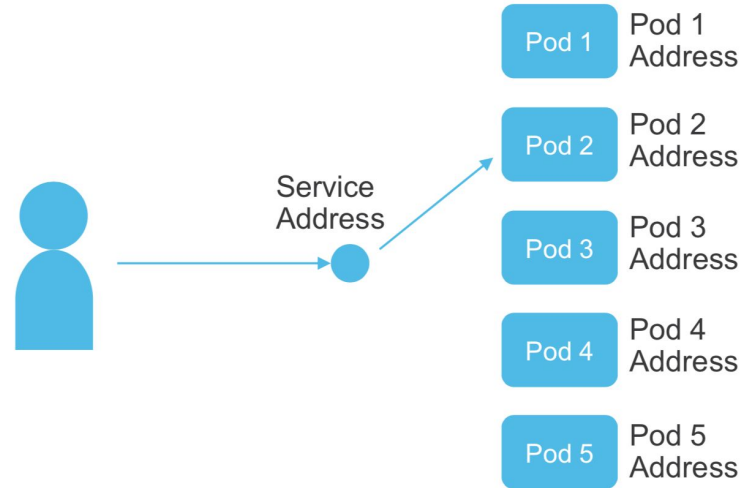
- Node:
  - A server
- Cluster:
  - Collection of nodes
- Pod:
  - Collection of containers;
  - Nodes can run multiple Pods





# Services overview

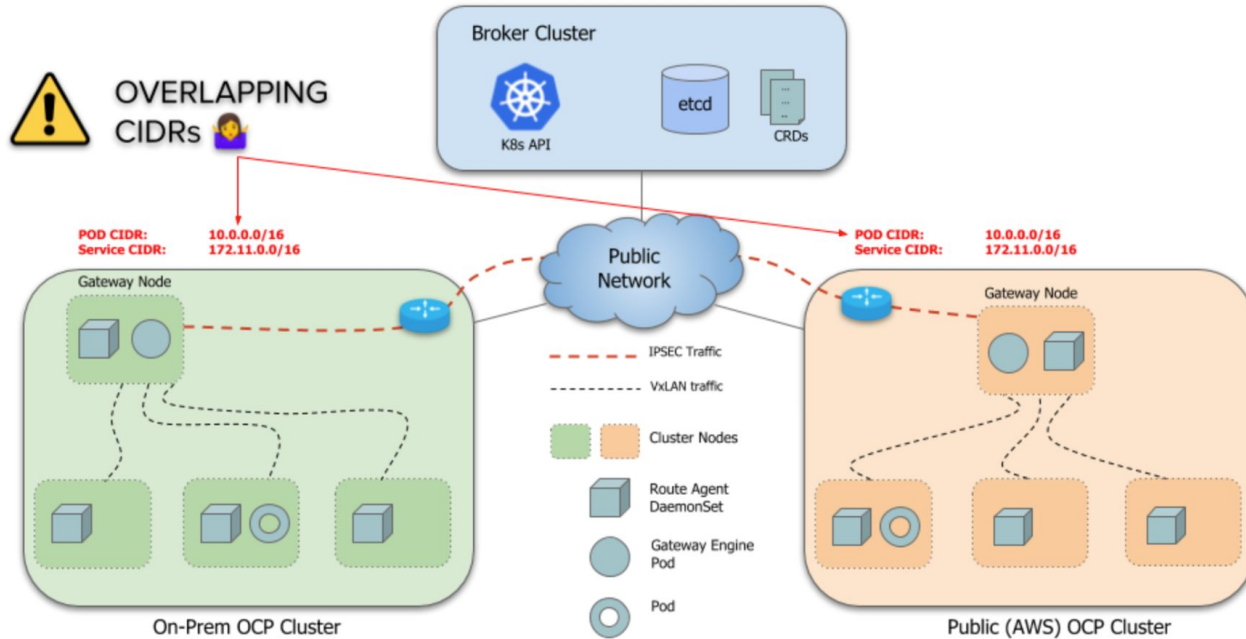
- “Pods can come and go, services stay”
- Define a single IP/Port combination that provides access to a pool of pods
- By default a service connects the client to a Pod in a round- robin fashion
- This solves the dilemma of having to keep up with every transient IP address



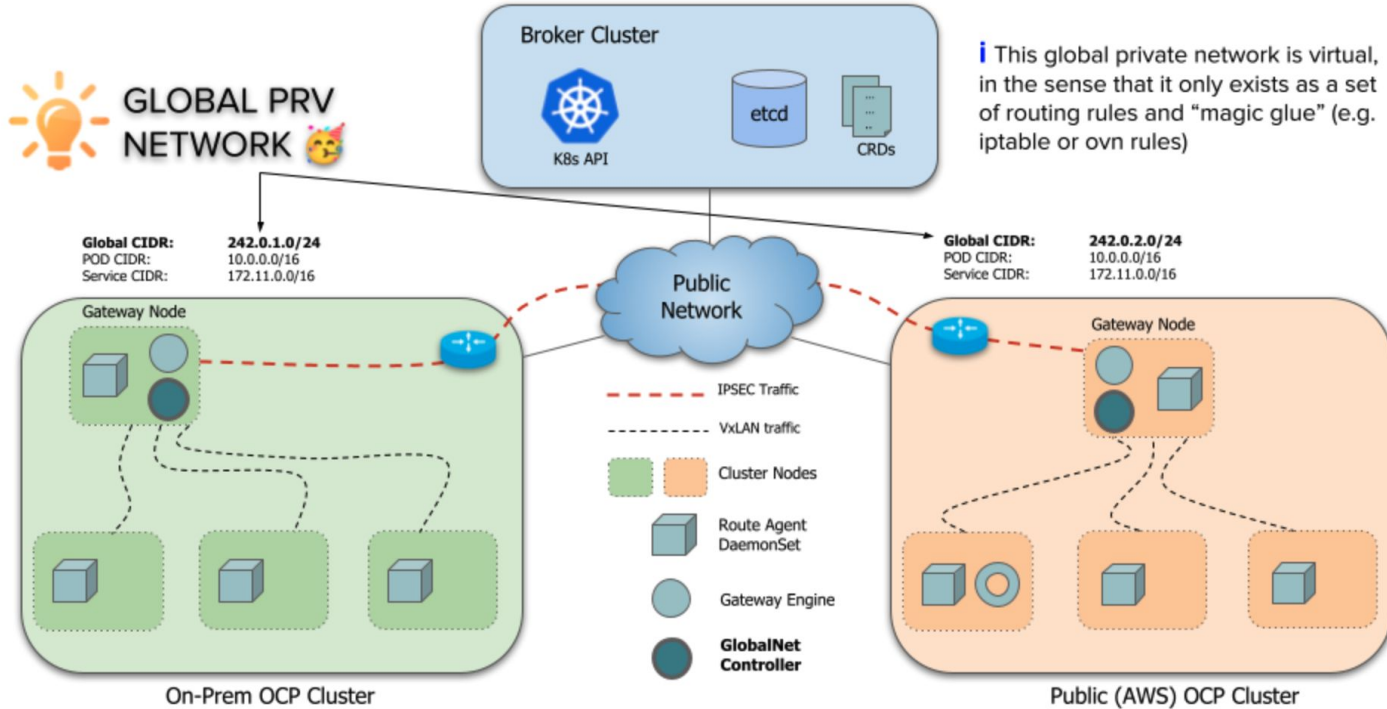
# Some IPv4 examples...

Let's start with Submariner

# Multicluster networking with Submariner.io



# Hijacking public ranges



# Stories from the field

100.64.0.0/16

# Evolution to IPv6 in Kubernetes

# IP Addresses in Kubernetes

- Originally Kubernetes was IPv4-only
- Moving to dual stack (simultaneous IPv4+IPv6) is hard; lots of places that support only 1 IP need to be updated...
  - `pod.status.podIP`
  - `service.status.clusterIP`
  - `kubelet --node-ip`
  - `node.status.addresses` ... already plural but some existing clients expect to only see IPv4 addresses there!
- “Single-stack IPv6” introduced as a transitional mechanism

# Single-Stack IPv6

- Instead of every IP address in the cluster being IPv4, every address is IPv6 instead.
- Alpha in kubernetes 1.9, beta in 1.18. Turns out to not actually be very useful...
  - **Every** address has to be IPv6
  - Pods *only* have IPv6 addresses, and hence only IPv6 routing to the outside world.
  - It turns out lots of people want 99% IPv6 and 1% IPv4, but kubernetes single-stack IPv6 can't do that.



# Dual Stack

- Alpha since kubernetes 1.11
- Every pod gets both an IPv4 address and an IPv6 address
  - added `pod.status.podIPs` in addition to old `pod.status.podIP`
- In the original alpha API, nodes and services were still single stack
  - Nodes might have an IPv6 IP, but Kubernetes wouldn't know about it.
  - Services could be defined as either IPv4-only (`ipFamily: IPv4`) or IPv6-only (`ipFamily: IPv6`) and would get one IP.

# Dual Stack

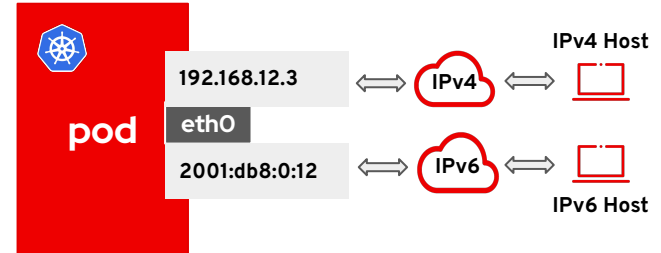
- SIG Network decided we really needed to change Services to be fully dual-stack
  - `ipFamilyPolicy` to request `SingleStack` / `PreferDualStack` / `RequireDualStack` on service creation
  - `ipFamilies` to indicate the IP family or families to support
  - `clusterIPs` replacing/extending `clusterIP`
- Also improving the situation around node IPs, load balancers, etc

# Dual Stack

- Since the old API was only alpha, the old single-valued `ipFamily` field was dropped entirely in favor of `ipFamilies`, rather than being deprecated.
- Thus the kube 1.20 dual-stack API is not compatible with the kube 1.19 dual-stack API.
- IPv4/IPv6 dual-stack networking is enabled by default for your Kubernetes cluster starting in 1.21, allowing the simultaneous assignment of both IPv4 and IPv6 addresses.

# IPv4/IPv6 Dual Stack Support Configuration

- IPv6 single/dual stack is supported in OpenShift 4.8 (k8s 1.21) with OVN.
- **Single Stack**
  - Either an IPv4 or IPv6 address is assigned to the pod interface
- **Dual Stack**
  - Both IPv4 and IPv6 addresses assigned to the interface
- Simple install-time configuration
  - Modify `"install-config.yaml"` to specify IPv6 subnets in addition to IPv4.
- Post-install configuration:
  - Edit `"network.config.openshift.io"` config to add secondary `"(machine|cluster|service)Network"` values, and they will get rolled out correctly.
- Restrictions / Caveats / Notes
  - OVN *only*, no plans to support in openshift-sdn
  - Supported platform at GA: Bare Metal IPI (other platforms TBD)



# IPv6 Enablement

# Kubernetes Networking CIDR Ranges

- **Machine CIDR**
  - IP address range for machines or cluster nodes
- **Service CIDR**
  - IP address range for services
- **Pod (Cluster) CIDR**
  - IP address range for pods
- **Host Prefix**
  - The host prefix determines the pod IP address pool for each machine. If the host prefix is set to /64, each machine is assigned a /64 subnet from the pod CIDR address range.

# Enabling IPv6 Single Stack in OpenShift Clusters

- Initiate an **OVN** install, creating an `install-config.yaml`:
  - `openshift-install create install-config`
- In the `install-config.yaml` file:
  - For OVN, change the `networkType` to `OVNKubernetes`
  - For single-stack IPv6, set the following fields to IPv6 values in `install-config.yaml`:
    - `clusterNetwork`
    - `serviceNetwork`
    - `machineNetwork`
- Continue with the cluster installation process:
  - `openshift-install create cluster`

```
networking:  
  networkType: OVNKubernetes  
  machineCIDR: 2001:db8:0:1202::/64  
  clusterNetwork:  
  - cidr: fd01::/48  
    hostPrefix: 64  
  serviceNetwork:  
  - fd02::/112
```

# Enabling IPv6 Dual Stack in OpenShift Clusters

- Initiate an **OVN** install, creating an `install-config.yaml`:
  - `openshift-install create install-config`
- In the `install-config.yaml` file:
  - For OVN, change the `networkType` to `OVNKubernetes`
  - For dual-stack, set appropriate dual-stack values for `clusterNetwork`, `serviceNetwork`, and `machineNetwork`. For example:

```
clusterNetwork:  
- cidr: 10.128.0.0/14  
  hostPrefix: 23  
- cidr: fd01::/48  
  hostPrefix: 64
```

- Generate the manifest files:
  - `openshift-install create manifests`



# Deploying Services

```
"apiVersion": "v1",
"kind": "Service",
"metadata": {
  "labels": {
    "name": "test-service"
  },
  "name": "test-service"
},
"spec": {
  "ports": [
    {
      "name": "http",
      "port": 27017,
      "protocol": "TCP",
      "targetPort": 80
    }
  ],
  "ipFamilyPolicy": "RequireDualStack",
  "selector": {
    "name": "test-pods"
  }
}
```

