Red Hat OpenShift

Welcome UDN!

OpenShift's new Network Model makes it easy

Joachim von Thadden

Senior Principal Specialist Solution Architect, Red Hat

Red Hat

$ whoami

Joachim von Thadden
EMEA Senior Principal Specialist Solution Architect

- based in Germany, near Düsseldorf
- more than 25 years in IT
- more than 30 years working with Linux
- >10 years experience with OpenStack & OpenShift
- 9.5 years at Red Hat

Network: update

Dictionary

Definitions from Oxford Languages · Learn more

network

/ˈnɛtwəːk/

noun

1. an arrangement of intersecting horizontal and vertical lines.
   "a spider constructs a complex network of several different kinds of threads"

   Similar: web | criss-cross | grid | lattice | net | matrix | mesh | webbing

2. a group or system of interconnected people or things.
   "the company has a network of 326 branches"

   Similar: system | complex | complex system/arrangement | nexus | web

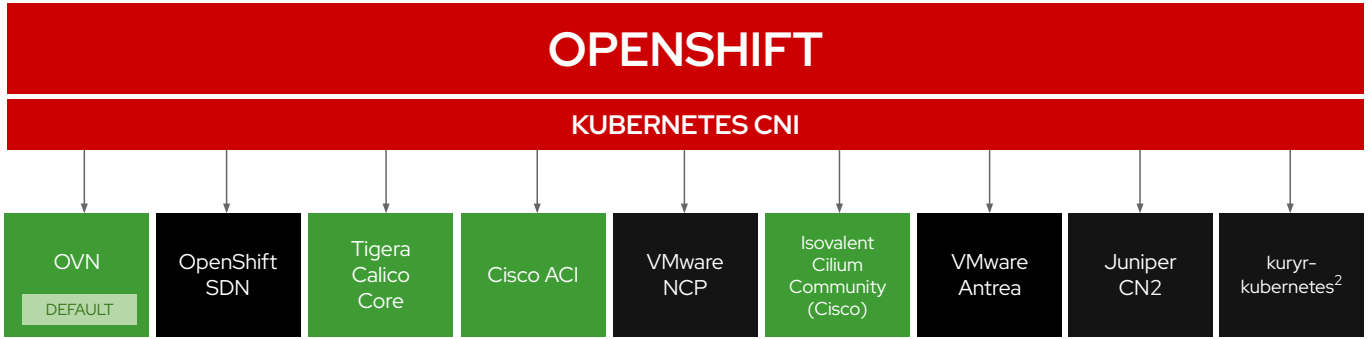
verb

1. connect as or operate with a network.
   "compared with the railways the canals were less effectively networked"
2. interact with others to exchange information and develop professional or social contacts.
   "it's so important to network when starting a new business"

Red Hat

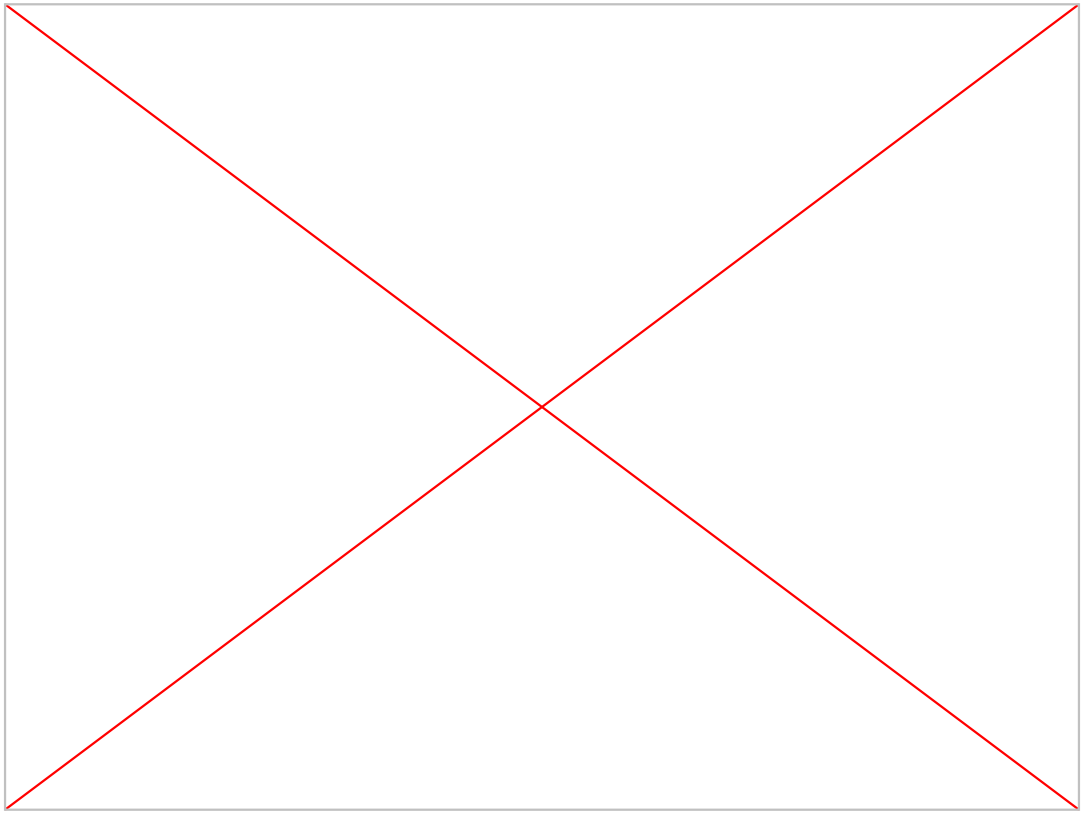
OpenShift Primary Networking CNI Plug-Ins

https://access.redhat.com/articles/5436171



Fully Supported | Deprecate Notice

Partner	Product	Latest Version(s)	Installer	OpenShift version
VmWare	NCP	3.0.2 w/NSX-T3.x+	UPI	4.4
VmWare	Antrea	1.8.0	IPI	4.11 - 4.13
Cisco	ACI (bare metal)	5.2.x, 5.3.x, 6.0.x, 6.1.x	UPI	4.10 - 4.17
Cisco	Isovalent/Cillium	1.16	UPI & IPI	4.16 - 4.17
Tigera	Calico Core	3.29	IPI	4.14 - 4.17
Juniper	Contrail (CN2)	23.1	Assisted	4.12



Red Hat

What about Multus? Why isn't it 'good enough'?

Multus is a Kubernetes meta-plugin for multi-homed pods, that will continue to serve a purpose for secondary networks going forward. Some idiot (me) even wrote a blog post about it many moons ago - Demystifying Multus

But Multus does have several downsides, including:

- Configuration **complexity**: Multus uses unstructured annotations, which can make configuration error-prone.
- **Limited flexibility**: Multus doesn't allow for dynamic network addition or removal to pods.
- Limited Kubernetes integration: Multus **lacks native support for network policies and network services**.
- **Poor observability**: It's hard to monitor and troubleshoot multi-network setups with Multus.
- **Requires iptables**: Multus uses kernel networking, but it needs to implement modules to work with user-space networking.
- Requires IPv4: Multus currently requires IPv4.

Linux Bridge vs UDN: Localnet

Table 10.1. Linux bridge CNI compared to an OVN-Kubernetes localnet topology

Feature	Available on Linux bridge CNI	Available on OVN-Kubernetes localnet
Layer 2 access to the underlay native network	Only on secondary network interface controllers (NICs)	Yes
Layer 2 access to underlay VLANs	Yes	Yes
Network policies	No	Yes
Managed IP pools	No	Yes
MAC spoof filtering	Yes	Yes

Source: https://docs.redhat.com/en/documentation/openshift_container_platform/4.18/html/virtualization/networking#secondary-network-config

UDN

User Defined Networks: Introduction

Flexible Network Configurations for the users

Workload/Tenant Isolation

Ability to group different types of applications in different isolated networks that cannot talk to each other

Flexible Network Topologies

Ability to create **layer3** or **layer2** or **localnet** type networks which can act as either **primary** and **secondary** networks for your pods

Overlapping podIPs

Ability to create Multiple Networks in your cluster with same pod Subnet range thereby possible to have copies of setups!

Kubernetes APIs supported!

Primary UDNs will have full support for **services**, **network policies**, **admin network policies**, **egressIPs**, and **ocp-routes**

In Essence:
We want true SDN!

In Essence:
We want true SDN!
With and without Routing!

In Essence:
We want true SDN!
With and without Routing!
And direct Network Access!

User Defined Networks API

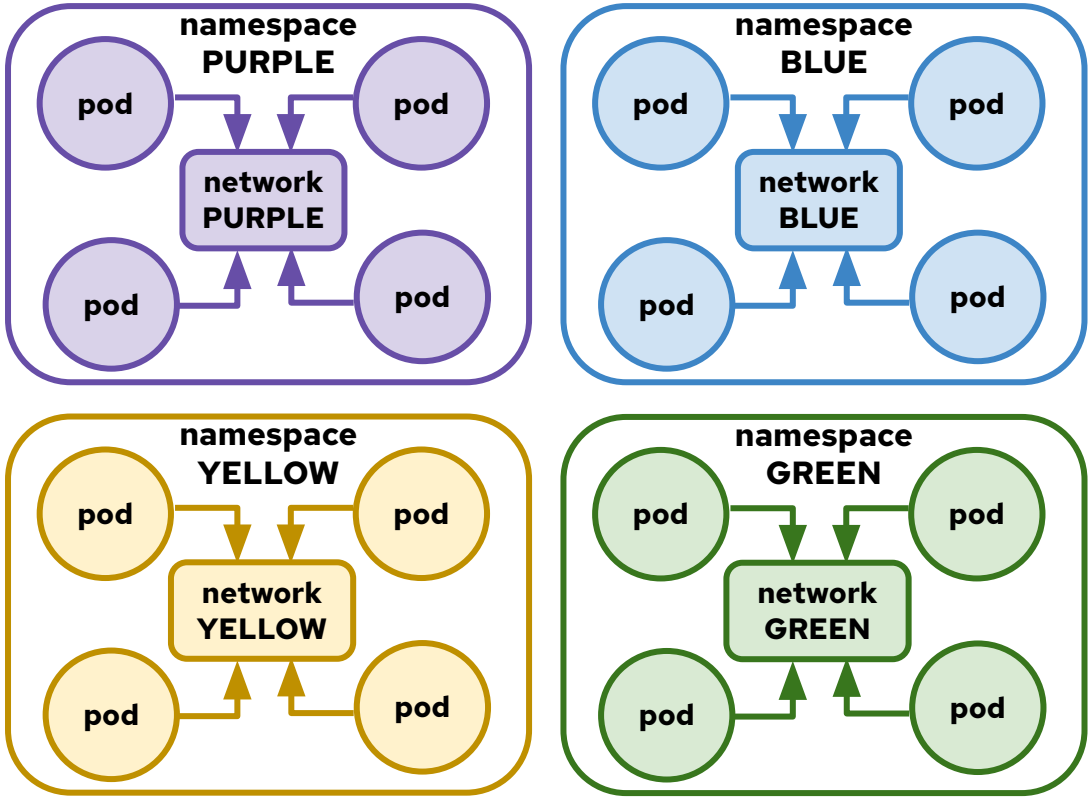
Solution: The WHAT?

- Each of these CRDs has a field called `role` which supports two options:
  - **Primary (**Also known as P-UDN -> Primary **UserDefinedNetwork):** This means the network will act as the primary network for the pod and all default traffic will pass through this network**
  - **Secondary (**Also known as S-UDN -> Secondary **UserDefinedNetwork):** This means the network will act as only a secondary network for the pod and only pod traffic that is part of the secondary network may be routed through this interface

** Except for Kubelet health checks

User Defined Networks Use Case

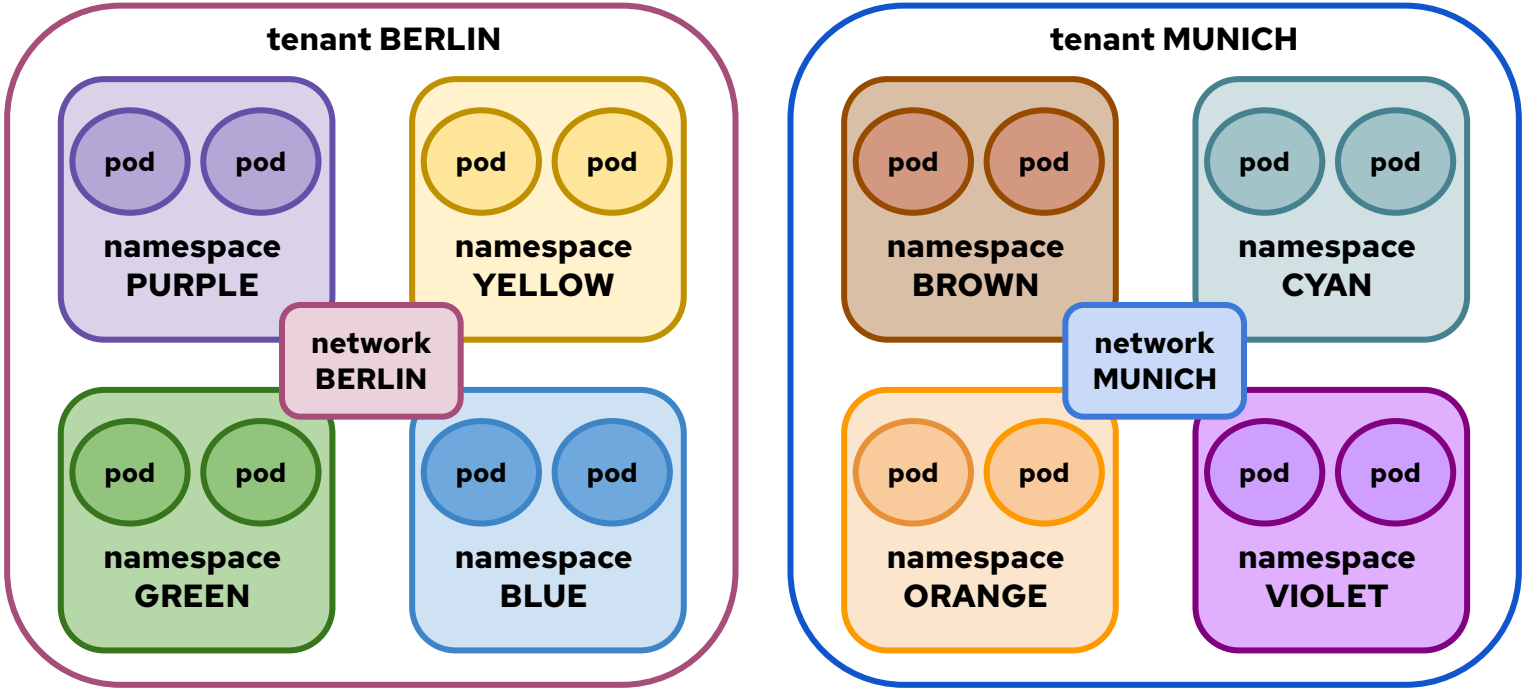
Native Namespace Isolation: Each namespace can be 1 unique UDN



Networks purple, blue, yellow and green are disconnected networks (logically isolated islands) which guarantees native isolation of these namespaces

User Defined Networks Use Case

Native Tenant Isolation: Multiple namespaces can be connected to be part of the same UDN

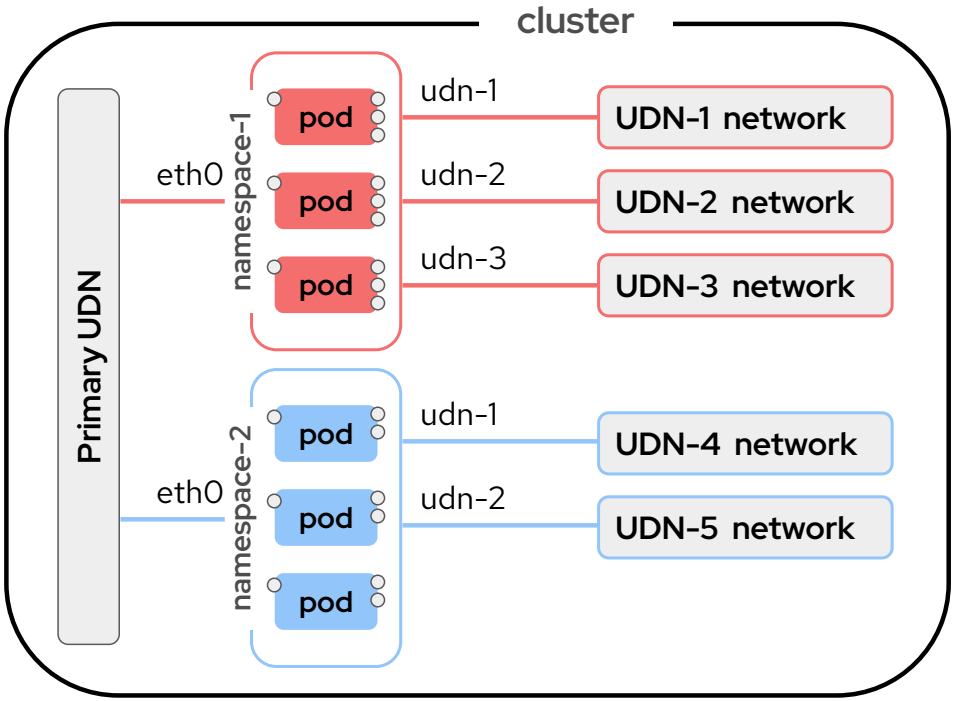


Tenants BERLIN and MUNICH are disconnected networks that guarantees native isolation

Namespaces purple, yellow, green and blue can talk to each other since they are all connected to BERLIN network but cannot talk with tenant MUNICH's namespaces

Primary and Secondary UDNs

Every pod/VM can have only one Primary UDN, but can have multiple Secondary UDNs

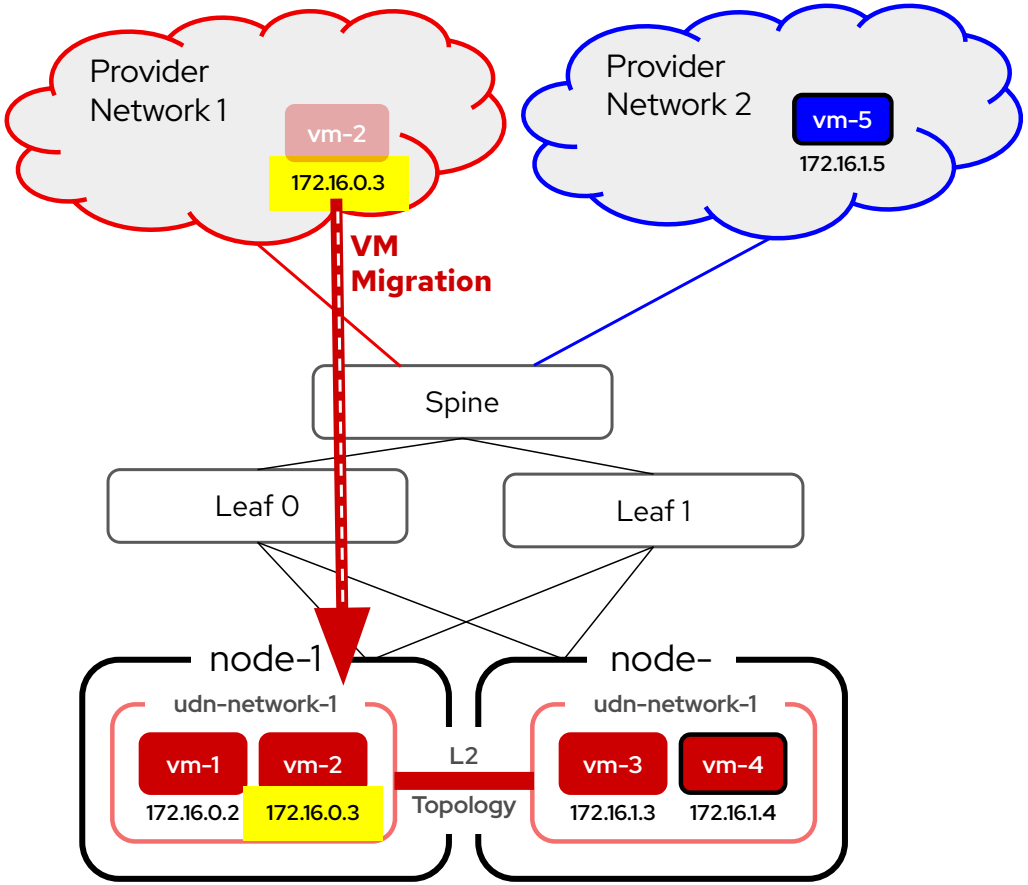


- Pods and VMs designate a **primary** UDN, which is where all traffic flows by default (*default-route*).
- There can only be a single **primary** UDN per namespace.
- Any number of **secondary** UDNs can be defined for a specific purpose of your designation.
- The default pod network (connected to the pod's eth0 interface) is currently only used for Kubernetes pod healthchecks.

User Defined Network API

- Each of these CRDs has a field called `topology` in its SPEC which supports three options:
  - **`Layer3:`** OVN-Kubernetes will end up creating a layer3 type logical network topology in OVN
    - Can be created as primary or secondary network roles for a pod
    - Available via UDNs and CUDNs and NADs
  - **`Layer2:`** OVN-Kubernetes will end up creating a layer2 type logical network topology in OVN
    - Can be created as primary or secondary network roles for a pod
    - Available via UDNs and CUDNs and NADs
  - **`Localnet:`** OVN-Kubernetes will end up creating a localnet type logical network topology in OVN
    - Can be created only as a secondary network role for a pod
    - Available via NADs; **Only available via CUDNs from OCP 4.19**

ROADMAP: OVN-K8s Support for BGP+EVPN



In-Progress: BGP+EVPN for exposing a VM into a provider's network.

- BGP as a routing protocol for UDNs
- EVPN, a common data center networking fabric that relies on BGP for dynamically exposing cluster scoped network entities into a provider's network, as well as program BGP-learned routes from the provider's network into OVN
- **Use case:** Extend UDN into provider networks, so a VM can be directly referenced by its (static) L2 network address, rather than requiring NAT translation at the cluster edge
- **Use case:** Live migrate a VM between a provider network and an OCP cluster

*This slide contains forward-looking roadmap information and as such should not be taken as an absolute implementation plan or outcome.

Q & A

FIN