represented on

Make On-Prem Bare-Metal Kubernetes Network Stack Telco Ready

Marcel Fest & Christopher Dziomba | DT Technik GmbH | May 18, 2022

Agenda

1. Introduction
2. What is Das SCHIFF?
3. The Narrative
4. Conclusion

represented on

01
Introduction

Make On-Prem Bare-Metal Kubernetes Network Stack Telco Ready

- 🏠 Located in Düsseldorf, Germany
- ☕ Coffee drinker
- 🚴 Adventuresome, Casual cycling
- 🏂 Amateur snowboarder

Marcel Fest
Devops Engineer
Deutsche Telekom Technik GmbH

You can find me on:

Make On-Prem Bare-Metal Kubernetes Network Stack Telco Ready

- 🏠 Located in Bonn, Germany
- 👨‍🍳 Loves to cook
- 👩‍💻 Software & Hardware tinkerer at night
- 🏂 Passionate skier

Christopher Dziomba
Devops Engineer
Deutsche Telekom Technik GmbH

You can find me on:

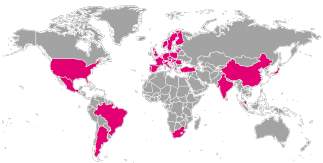
Organization

Organization

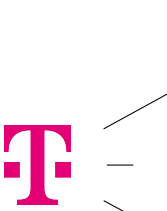
Telekom Deutschland GmbH

T-Systems International GmbH

T-Mobile US Inc.



Organization



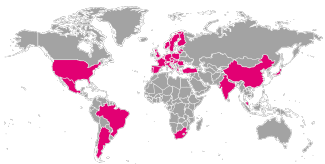
Telekom Deutschland GmbH

T-Systems International GmbH

T-Mobile US Inc.

Deutsche Telekom Technik GmbH

Das SCHIFF



Organization

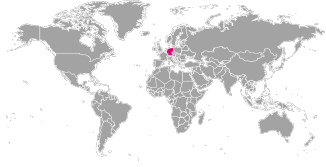
Telekom Deutschland GmbH

T-Systems International GmbH

T-Mobile US Inc.

Deutsche Telekom Technik GmbH

Das SCHIFF



Techn1k in Numbers

Mobile

- **>33K** cell towers.
- **>20K** 5G Ready.
- **>75%** connected via fiber.

- **>1.5K/yr** cell towers in the next years.
- *2025*: **99%** citizenship coverage for mobile Broadband with **100Mbit/s**.
- *2025*: **90%** of Germany covered with **5G**.

Broadband/Fixed

- **>650K** kilometers of optical fiber.
- **>34M** fixed lines *Vectoring/FTTB/FTTH*.
- **>2.6M** masts for above ground Landlines.

- **FTTH-Factory** Automatic Fiber planning.
- **>2.5M/yr** households in the next years.
- *2030*: FTTH for **every** German household.

Organization

Telekom Deutschland GmbH

T-Systems International GmbH

T-Mobile US Inc.

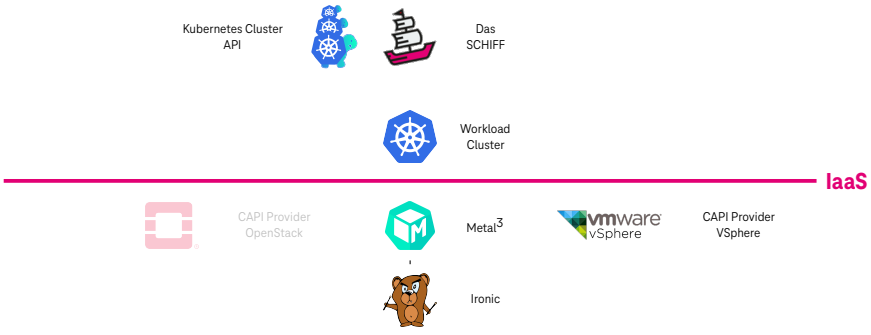
Deutsche Telekom Technik GmbH

Das SCHIFF

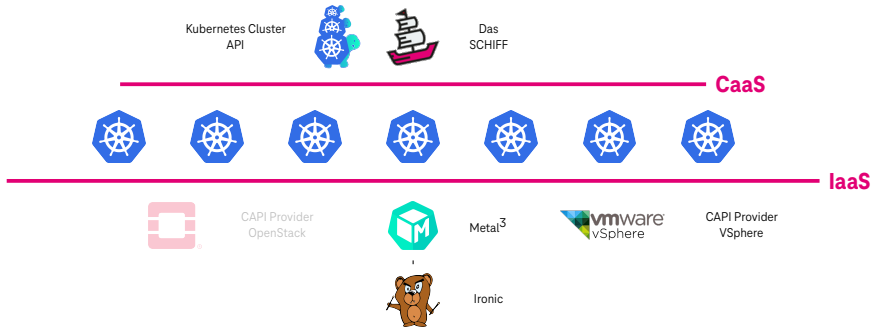
02
What is Das SCHIFF?

“An internal, GitOps based Kubernetes Cluster as a Service Platform almost exclusively built using open source components.”

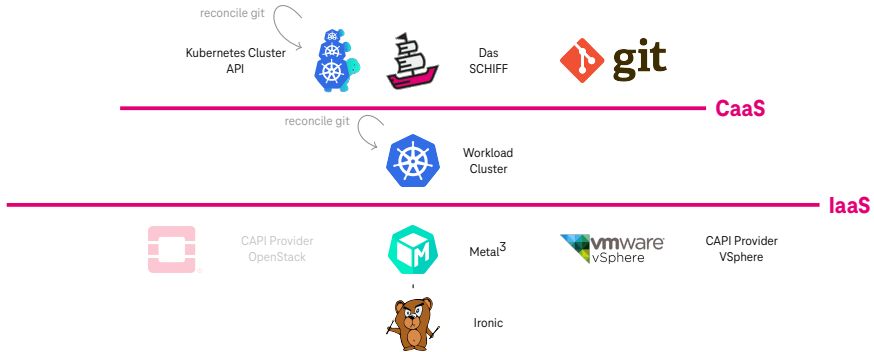
Das SCHIFF


Kubernetes Cluster API
Das SCHIFF
Workload Cluster
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

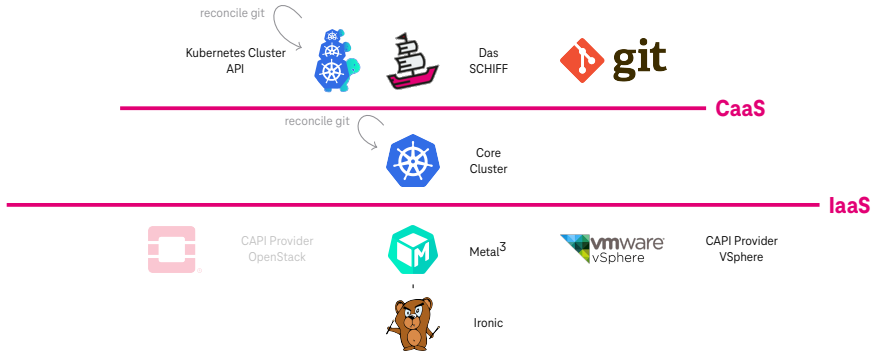
Das SCHIFF


Kubernetes Cluster API
Das SCHIFF
CaaS
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

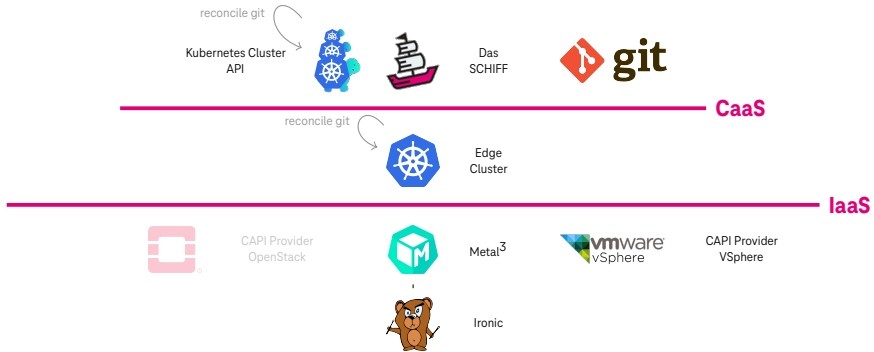
Das SCHIFF


reconcile git
Kubernetes Cluster API
Das SCHIFF
git
CaaS
reconcile git
Workload Cluster
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

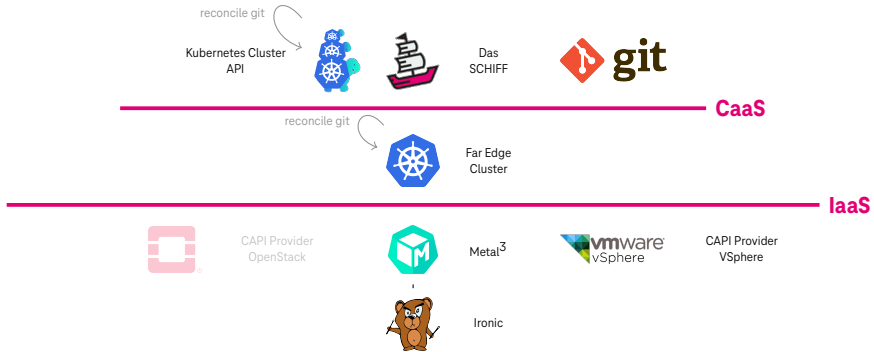
Das SCHIFF


reconcile git
Kubernetes Cluster API
Das SCHIFF
git
CaaS
reconcile git
Core Cluster
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

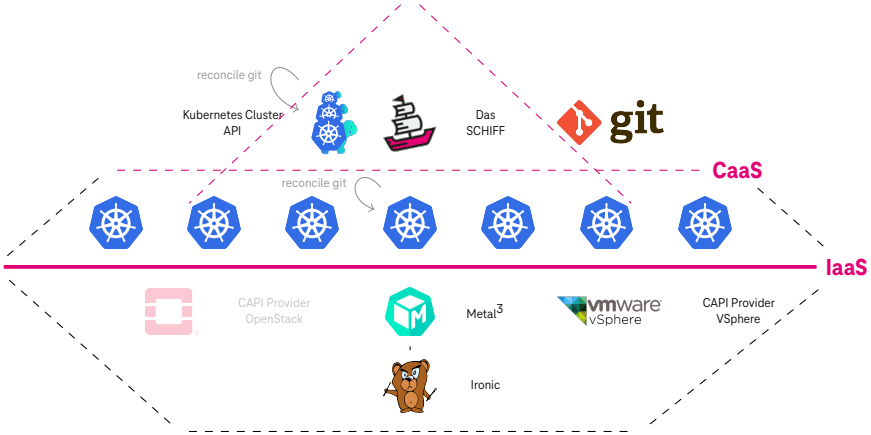
Das SCHIFF


reconcile git
Kubernetes Cluster API
Das SCHIFF
git
CaaS
reconcile git
Edge Cluster
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

Das SCHIFF


reconcile git
Kubernetes Cluster API
Das SCHIFF
git
CaaS
reconcile git
Far Edge Cluster
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

Das SCHIFF


reconcile git
Kubernetes Cluster API
Das SCHIFF
git
CaaS
reconcile git
IaaS
CAPI Provider OpenStack
Metal3
vmware vSphere
CAPI Provider VSphere
Ironic

Cloud-Native in a Telco World

Cloud-Native in a Telco World

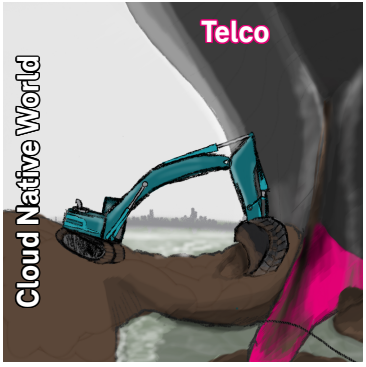


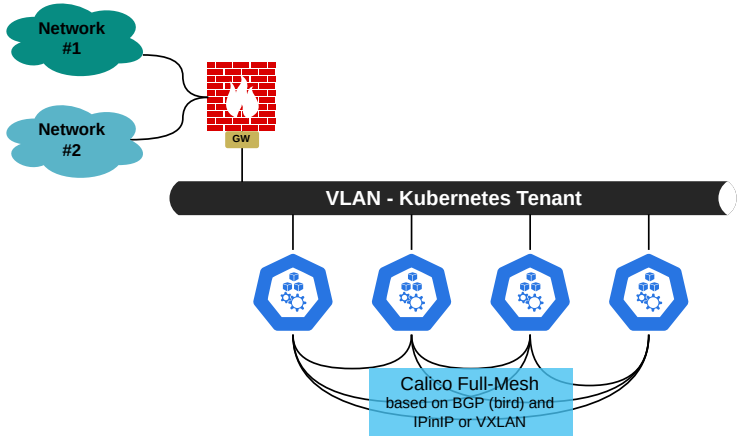
Figure 1: How Telco looks like in the current Cloud Native World [8]

Mission

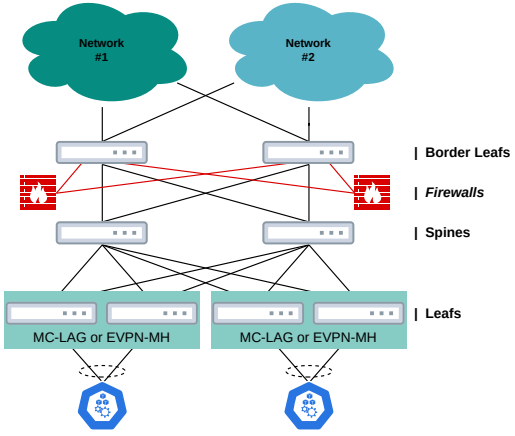
1. Reliable Kubernetes Clusters with well defined API Contract for internal customers.
2. Telco Grade Network Integration for Cloud Native Network Functions (CNFs).
3. Contribute to Upstream OpenSource projects to provide enhancements for our niche use cases.
4. Add the “Telco” view to the CNCF by providing Feedback.

03
The Narrative

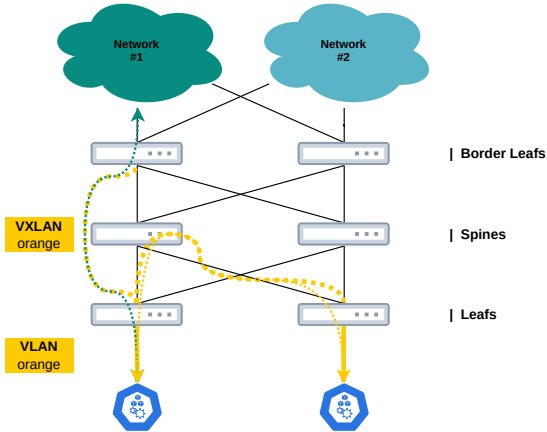
Problem Statement



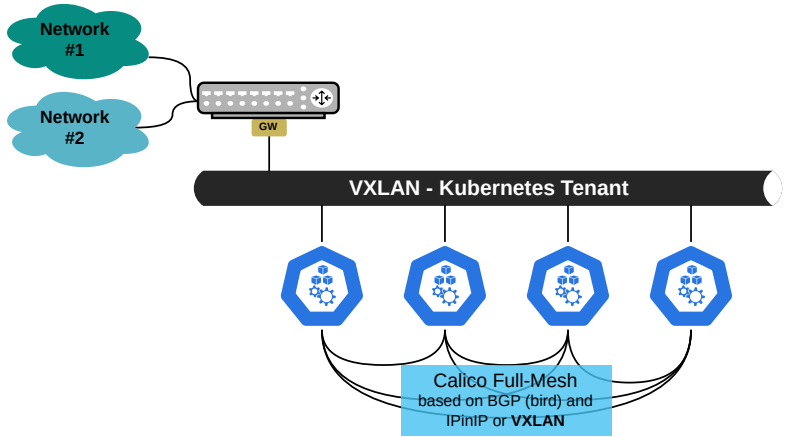
Introducing IP Fabrics



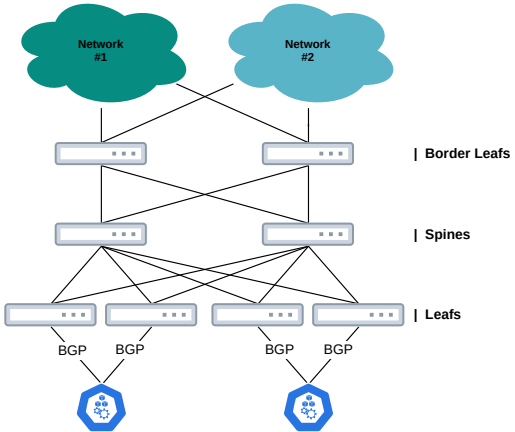
Traffic Flow in an IP Fabric



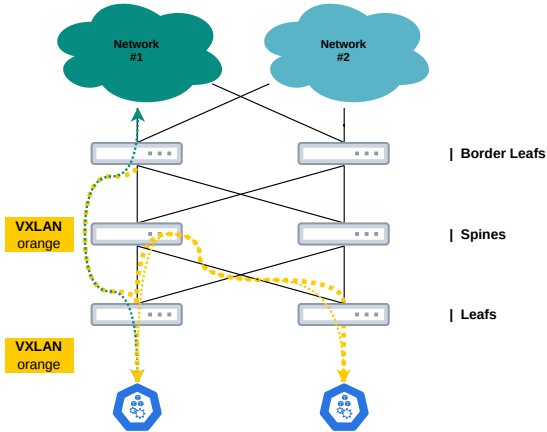
Layer 2 View


Network #1
Network #2
GW
VXLAN - Kubernetes Tenant
Calico Full-Mesh
based on BGP (bird) and
IPinIP or VXLAN

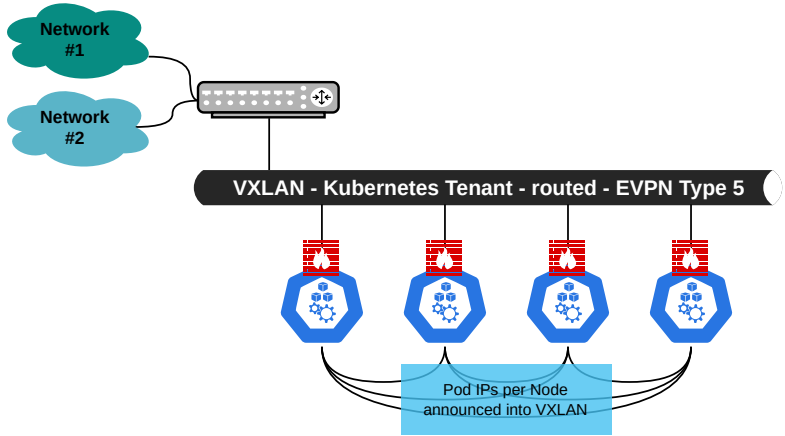
Making the Host part of our IP Fabric



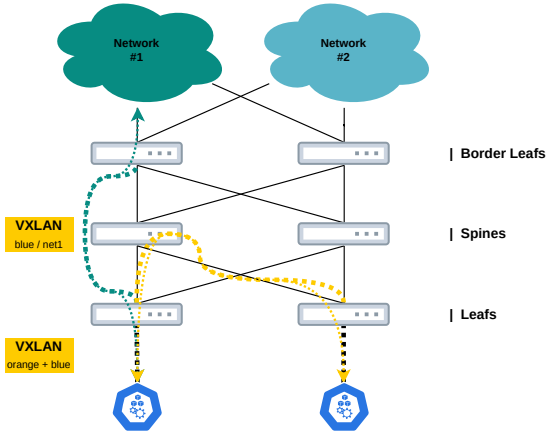
Traffic Flow



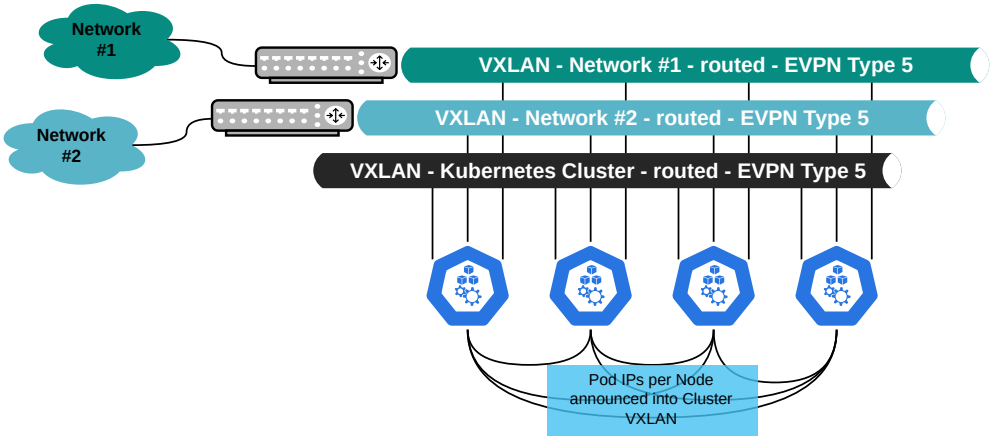
VXLAN View


Network #1
Network #2
VXLAN - Kubernetes Tenant - routed - EVPN Type 5
Pod IPs per Node announced into VXLAN

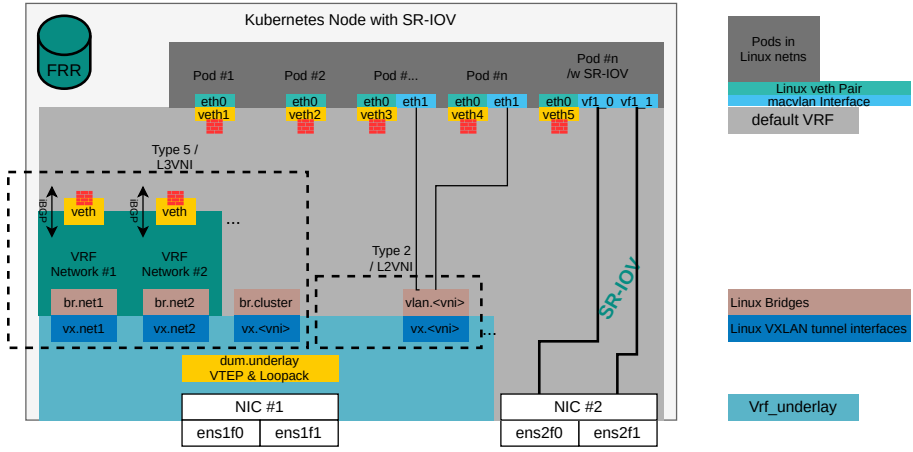
Solution


Network
#1
Network
#2
| Border Leafs
| Spines
| Leafs
VXLAN
blue / net1
VXLAN
orange + blue

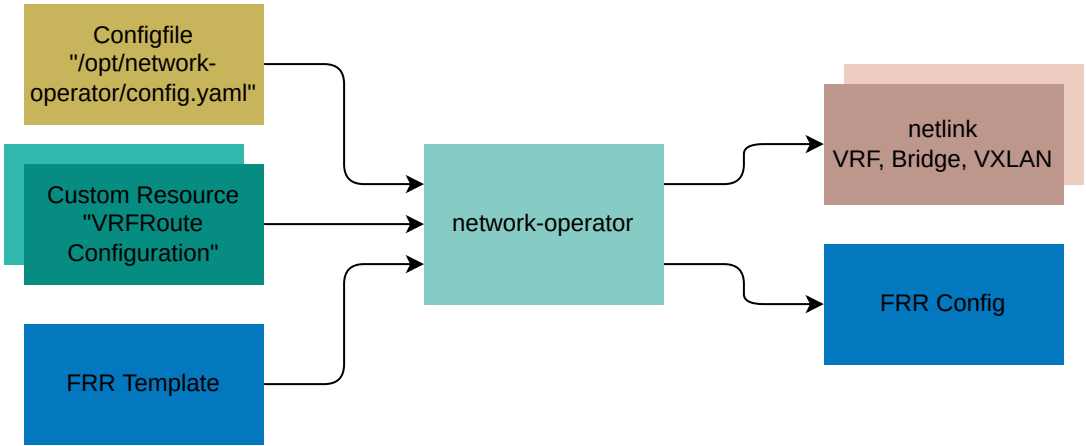
Network Connectivity from the Host



VRFs on a Kubernetes Host



Network-Operator I

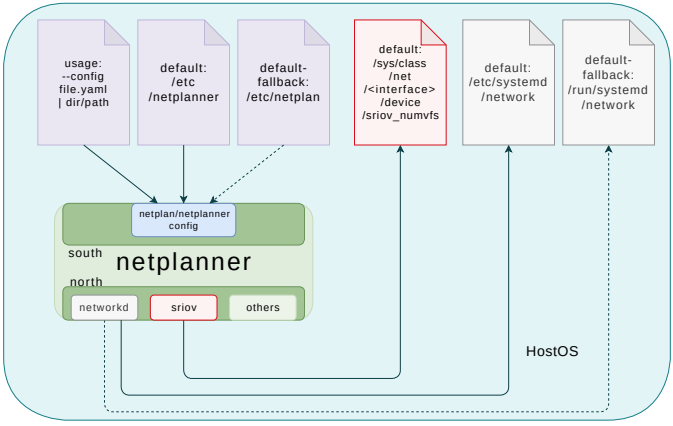


Network-Operator II

- **network-operator** translates Kubernetes CRs into netlink and FRR configuration
- Creates *VRF*, *L3VNI*, *VXLAN* interface and *FRR* configuration.
- Includes some workarounds for *L2VNI* with *macvlan* interfaces, *FRR* does not announce *permanent* FDB entries without adding SVI IP

```
apiVersion: ...
kind: VRFRouteConfiguration
metadata:
  name: bb-m2m-01
spec:
  vrf: "bb_m2m"
  import:
  - le: 32
    cidr: 0.0.0.0/0
    action: permit
  export:
  - cidr: 10.1.1.1/32
    action: permit
  seq: 1
```

Netplanner I



Netplanner II

- SRIOV support for the standard Linux **/sys/class/net** interface.
- **Partially** compatible with **netplan**
- Imperative definition of linux netdev wiring.
- **networkd** as the provider backend.
- **No** DHCP4 or DHCP6 support.
- **No** Wifi support.




```
network:
  renderer: networkd
  version: 3
  dummies: {...}
  ethernets: {...}
  bridges: {...}
  vxlans: {...}
  bonds: {...}
  vlans: {...}
  vrfs: {...}
  veths: {...}
  additionals: {...}
```

Caveats

- Linux VRFs are a pain
  - Very soft seperation because of *ip rule*
  - Route Leaking is pretty much disfunctional when trying to reach local endpoints
  - Introducing *veth* (and iBGP across them) between VRFs is working but iptables is applied twice
  - Easiest solution is a simple *tc-filter* BPF program
- All caveats we've discovered are part of **network-operator**
- If no broadcast or multicast is required, additional Multus interface could transition to being fully-routed as well
- SR-IOV is an outlier, only option to configure it is *tc-flower*, partly Open vSwitch (which we've omitted on purpose) or move to SmartNIC/DPU/IPU
  - mlx5 OFED is also checking routes in default VRF only, missing VRFs in our underlay VRF (yes, phys_dev of VXLAN interface is in underlay VRF)

04
Conclusion

Conclusion

- Kubernetes Hosts can directly participate in an IP / EVPN Fabric
- Added flexibility
- Linux VRFs are a pain but can work if used with caution
- SR-IOV is an open question, can vendors standardize around eBPF or memif (of VPP)?
- **netplanner** solves lack of VXLAN, VRF and VETH support in **netplan**
- **network-operator** serves as the glue between *Kubernetes* and *FRR*
- DPUs might be an option to offload the VRF peering on the network card

END

Questions?

Looking for a Job?

References I

[1] Datadog AG. Vector logo. https://github.com/vectordotdev/vector/blob/9658477bf36d7972a5da1e861730bf7701997911/website/static/img/logos/vector.svg, August 2021. Accessed on 2022-05-04.

[2] Russell Bryant. Metal3 icon. https://cncf-branding.netlify.app/img/projects/metal3/icon/color/metal3-icon-color.png, May 2022. Accessed on 2022-05-02.

[3] CNCF. Prometheus logo. https://cncf-branding.netlify.app/img/projects/prometheus/icon/color/prometheus-icon-color.png, Jul 2017. Accessed on 2022-05-04.

[4] CNCF. Thanos logo. https://cncf-branding.netlify.app/img/projects/thanos/icon/color/thanos-icon-color.png, May 2018. Accessed on 2022-05-04.

[5] CNCF. Cluster api logo. https://raw.githubusercontent.com/kubernetes-sigs/cluster-api/main/logos/kubernetes-cluster-logos_final-02.svg, May 2019. Accessed on 2022-05-04.

[6] CNCF. Kyverno logo. https://github.com/cncf/artwork/blob/c190ddedde70ce80101cf800f043094b28045ba2/projects/kyverno/icon/color/kyverno-icon-color.png, December 2020. Accessed on 2022-05-04.

[7] CNCF. Metallb logo. https://github.com/cncf/artwork/blob/c190ddedde70ce80101cf800f043094b28045ba2/projects/metallb/icon/color/metallb-icon-color.png, January 2020. Accessed on 2022-05-04.

[8] Hannah Fest. Telco in a cloudnative world. Private handdrawn Picture under Creative Commons v2 License, May 2022. Accessed on 2022-05-03.

[9] Daniel Finneran. Kube-vip logo. https://github.com/kube-vip/kube-vip/blob/ec0014d9ea96cd482088df60c317040886c445ac/kube-vip.png, December 2021. Accessed on 2022-05-04.

[10] Flux logo. https://cncf-branding.netlify.app/img/projects/flux/icon/color/flux-icon-color.png, March 2021. Accessed on 2022-05-04.

[11] OpenStack Foundation. Openstack logo. https://object-storage-ca-ymq-1.vexxhost.net/swift/v1/6e4619c416ff4bd19e1c087f27a43eea/www-images-prod/openstack-logo/OpenStack-Logo-Mark.png, April 2017. Accessed on 2022-05-04.

[12] Lucas Alvares Gomes. Ironic drummer bear. https://wiki.openstack.org/w/images/d/df/Ironic_mascot_color.png, December 2014. Accessed on 2022-05-02.

References II

[13] Tim Hockin. Kubernetes logo. https://cncf-branding.netlify.app/img/projects/kubernetes/icon/color/kubernetes-icon-color.png, May 2016. Accessed on 2022-05-04.

[14] GitHub Inc. Github mark favicon. https://github-media-downloads.s3.amazonaws.com/GitHub-Mark.zip, April 2022. Accessed on 2022-04-29.

[15] GitHub Inc. Github octocat. https://github-media-downloads.s3.amazonaws.com/Octocats.zip, April 2022. Accessed on 2022-04-29.

[16] LinkedIn Inc. Linkedin brand logo. https://content.linkedin.com/content/dam/me/brand/en-us/brand-home/downloads/LinkedIn-Logos.zip, April 2022. Accessed on 2022-04-29.

[17] Twitter Inc. Twitter brand logo. https://about.twitter.com/content/dam/about-twitter/en/brand-toolkit/downloads/twitter-logo-01282021.zip, January 2021. Accessed on 2022-04-29.

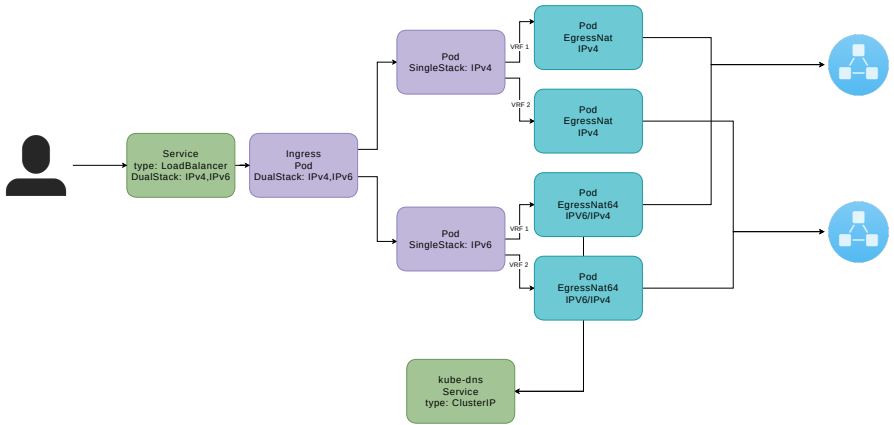
[18] Michael Krotschek. Ironic drummer bear metal. https://wiki.openstack.org/w/images/6/60/Bear_metal.svg, September 2015. Accessed on 2022-05-02.

[19] Jason Long. Git scm logo. https://git-scm.com/images/logos/downloads/Git-Logo-2Color.png, May 2012. Accessed on 2022-05-02.

[20] Tigera. Calico felix logo. https://github.com/projectcalico/calico/blob/df5e99e7430a758a5785c1e8e083de699a885899/calico/images/felix.png, August 2016. Accessed on 2022-05-04.

[21] Trent White. Grafana logo. https://github.com/grafana/grafana/blob/main/public/img/grafana_icon.svg, Jan 2016. Accessed on 2022-05-04.

NAT64 for the legacy world



CaaS: Components


grafana-operator


flux2


coil


metalLB


thanos


rbac-manager


calico


kube-vip


prometheus-operator


kyverno


velero


vector