KubeVirt VMs all the way down

June 16th, 2023, DevConf CZ

Felix Enrique
ellorent@redhat.com
https://github.com/qinqon/

Miguel Duarte Barroso
mdbarroso@redhat.com
https://github.com/maiqueb

KubeVirt

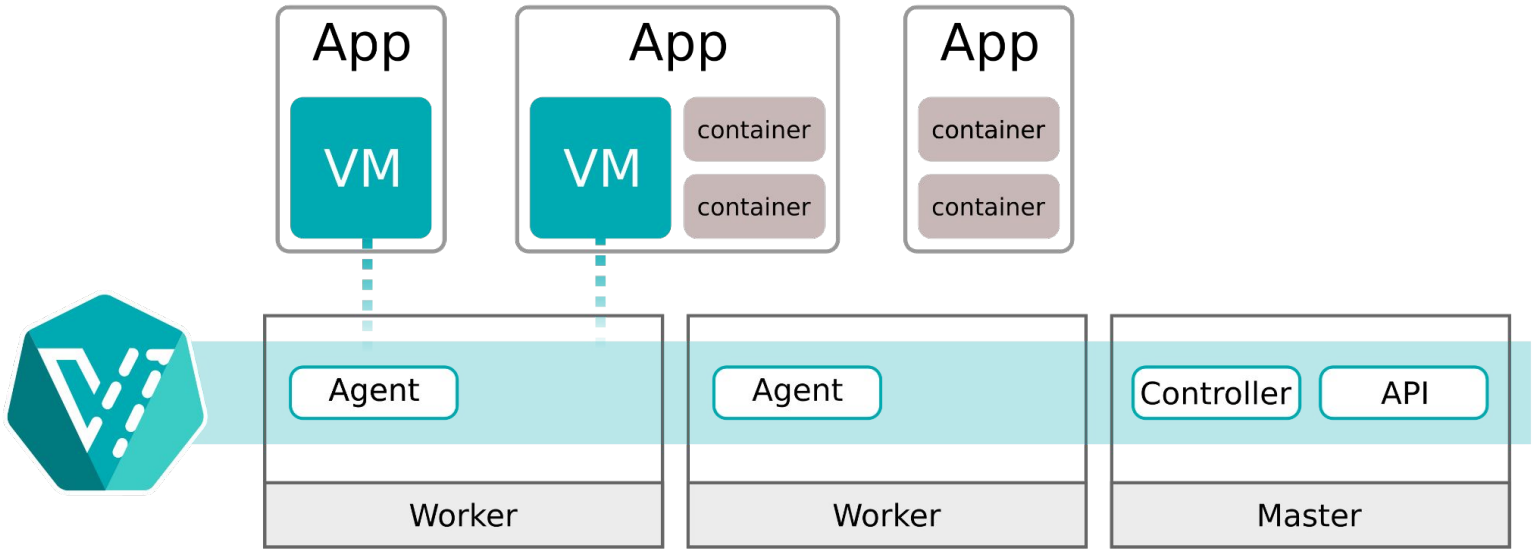
Agenda

- Intro
  - KubeVirt
  - cluster-api-provider-kubevirt (aka capk)
  - OVN-Kubernetes
- Motivation
- Goals
- Solution
- Demo
- Conclusions

KubeVirt

- Kubernetes plugin
  - Try it out
- Runs VMs alongside pods in the same platform (Kubernetes)
- Each pod runs libvirt + qemu process for the VM
  - Cattle vs pets ?...
  - Pet is alive within the cow
  - VM => stateful
  - Pod => stateless
- VM networking requirements >> pod networking requirements

KubeVirt


App
VM
App
VM
container
container
App
container
container
Agent
Agent
Controller
API
Worker
Worker
Master

CAP-K: Cluster API provider KubeVirt

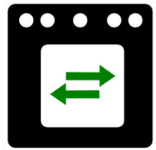
- Cluster API: declarative Kubernetes-style APIs to cluster creation, configuration and management
  - Consistent and repeatable cluster deployments across various infrastructure envs.
  - Different types of providers (AWS / GCP / Azure / …)
- CAP-K: cluster-API KubeVirt **provider**
  - Kubernetes nodes are KubeVirt VMs
- … why would you !?
  - Cluster scale
  - Cheap cluster provisioner / On demand cluster creation
  - Cross cloud portability
  - CI
- More info => checkout last year's KubeVirt summit capk presentation

https://github.com/kubernetes-sigs/cluster-api/blob/main/logos/kubernetes-cluster-logos_final-02.svg

OVN / OVN-K

- OVN provides a higher-layer of abstraction than Open vSwitch
  - SDN
  - Open vSwitch orchestrator
  - Logical routers / logical switches, ACLs, etc rendered to openflow

- OVN-Kubernetes => CNI plugin **for** Kubernetes
  - Opinionated topology
  - Translates Kubernetes objects to OVN logical entities



Motivation

- Decouple mgmt cluster node updates from tenant cluster VMs via live-migration
  - Without live migration the hosted cluster workloads will be disrupted by upgrades to the mgmt cluster
- What we have currently does not provide live-migration
  - Masquerade binding: masquerade inside pod
  - Bridge binding: GW config must be updated when migrating
- OVN provides live-migration for other projects - e.g. Openstack
  - Very minimal downtime - around 0.1 seconds - using latest improvements

Goals

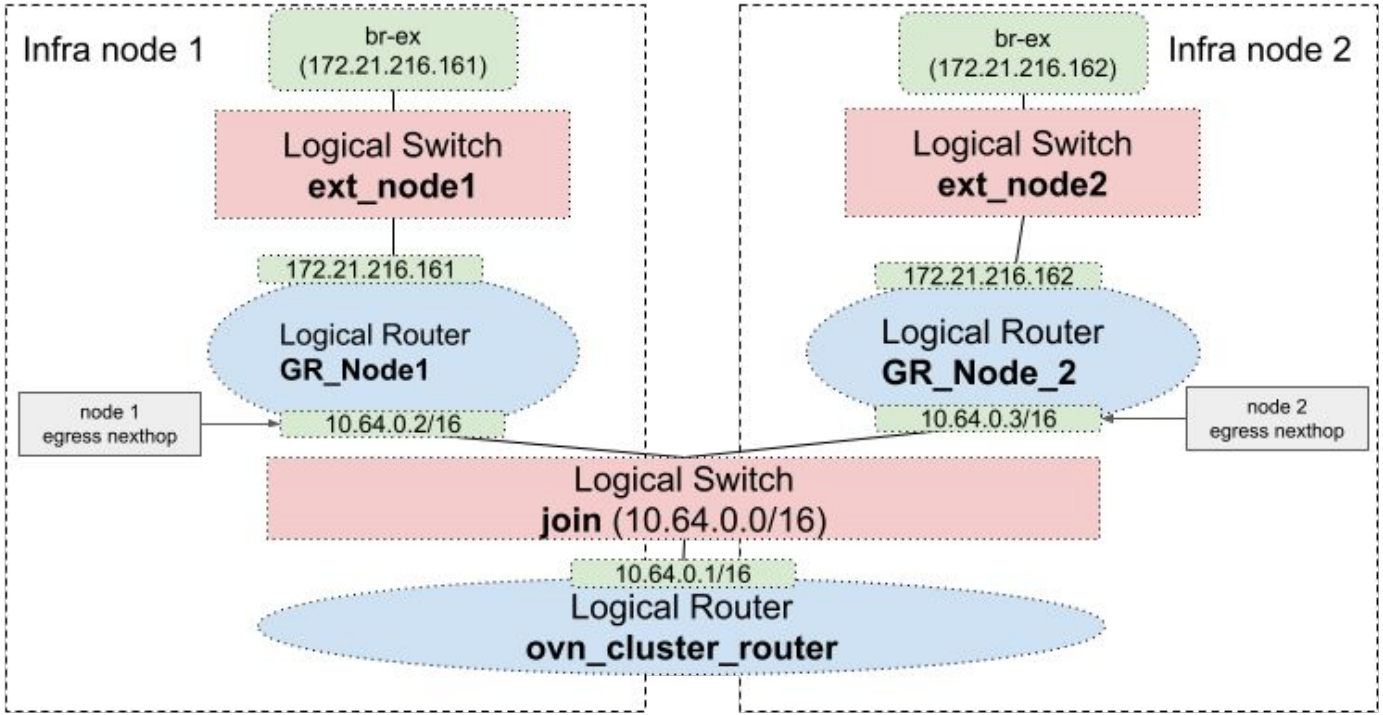
- TCP connections on the Kubernetes node **must survive** live migration
  - Kubelet …
  - Workloads !!!
  - Minimal downtime
- Consistent IP / GW config for worker nodes (VMs) during Live Migration
  - … Kubelet is bound to it
- Hosted Cluster Network isolation
  - Hosted clusters can only access other hosted clusters via public LBs
  - Hosted clusters can only access infra components via public LBs
- Expose workers as Kubernetes services
  - NodePort / LoadBalancer

The solution

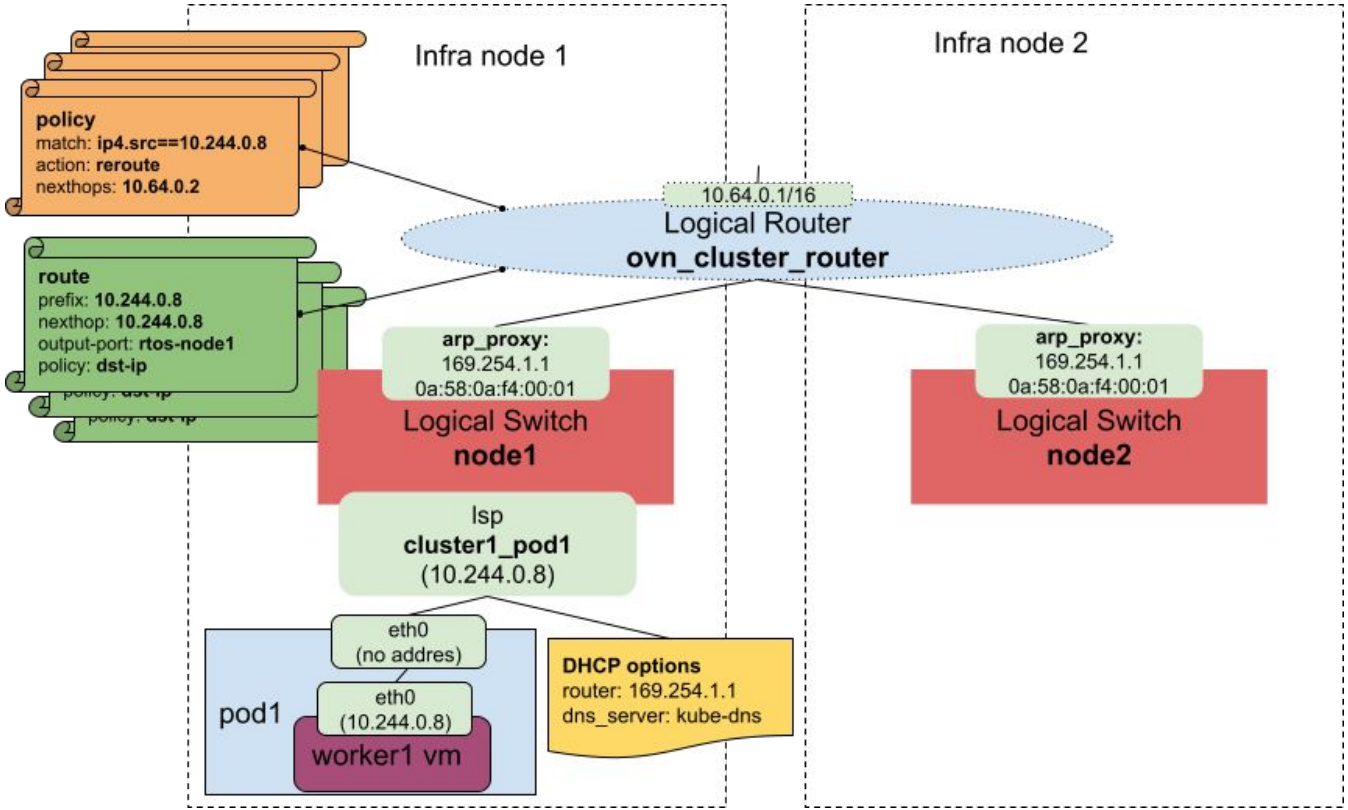
Implementation

- Use point-to-point routing everywhere
  - Use the **cluster default network**
  - CNI will **not** set IP addresses in the veth end in the pods
    - Configure DHCP Options at LSP
  - Use an stable IP and MAC addresses across nodes (169.254.1.1) using ARP proxy
    - Had to be implemented in OVN

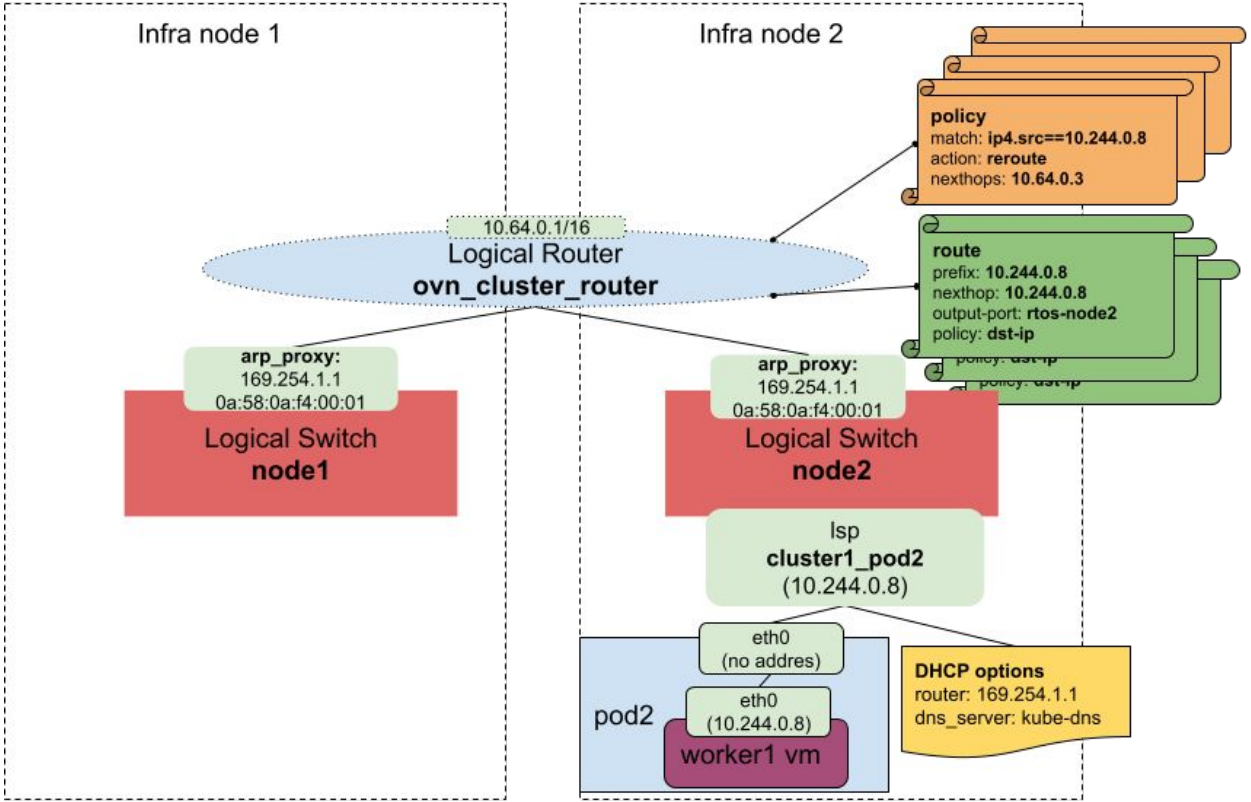
point-to-point routing, north: nexthop



point-to-point routing, south: before live migration

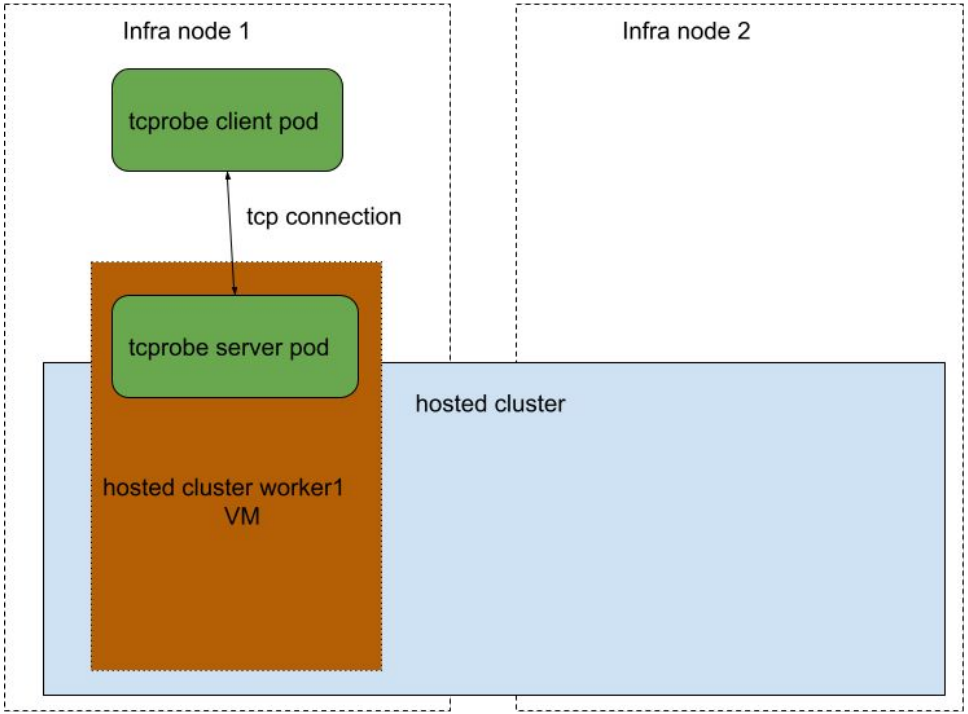


point-to-point routing, south: after live migration

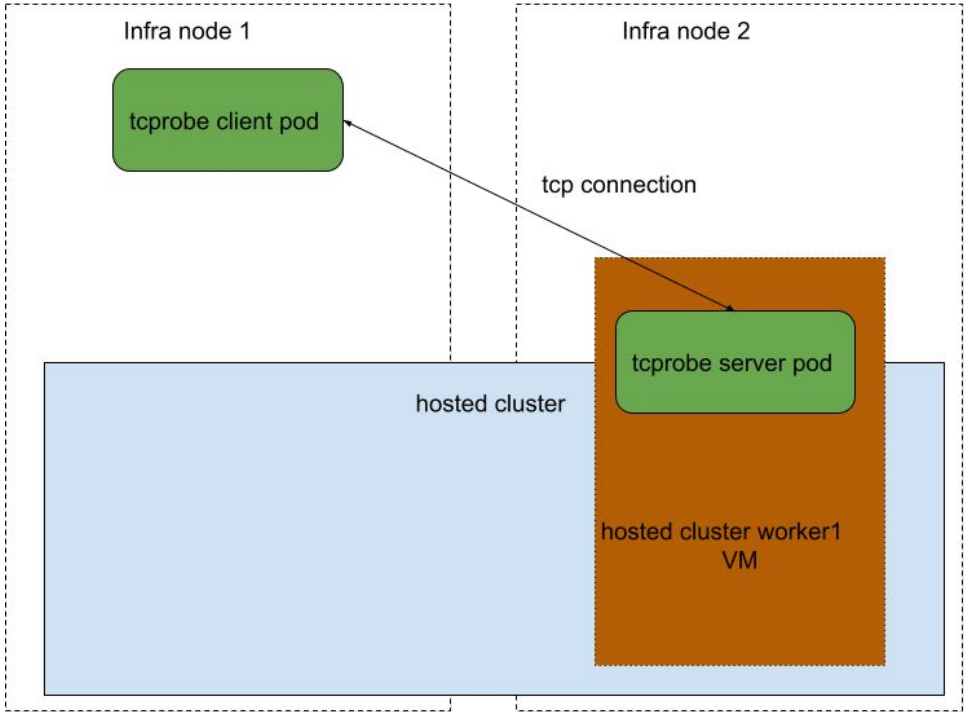


Live migration demo

Demo: before live migration



Demo: after live migration



Demo

Node migration: https://asciinema.org/a/hEqzoFveibsuHKynQxZ3RxCHZ

kubevirt summit 2022 capk demo: https://youtu.be/8sEs2ExtwDY?t=1425

Conclusions

- Using the pod’s default network provides following features "for free"
  - Hosted Cluster Network isolation
  - Expose hosted cluster workloads via services
  - Rich service implementation (node port / load-balancer / clusterIP)
- Using point-to-point routing on primary interfaces allows for
  - Consistent IPs during migration
  - Established TCP connections on the infra nodes survive live-migration
- Now we know what features to request from OVN-K

Thank you !!!

Questions ?