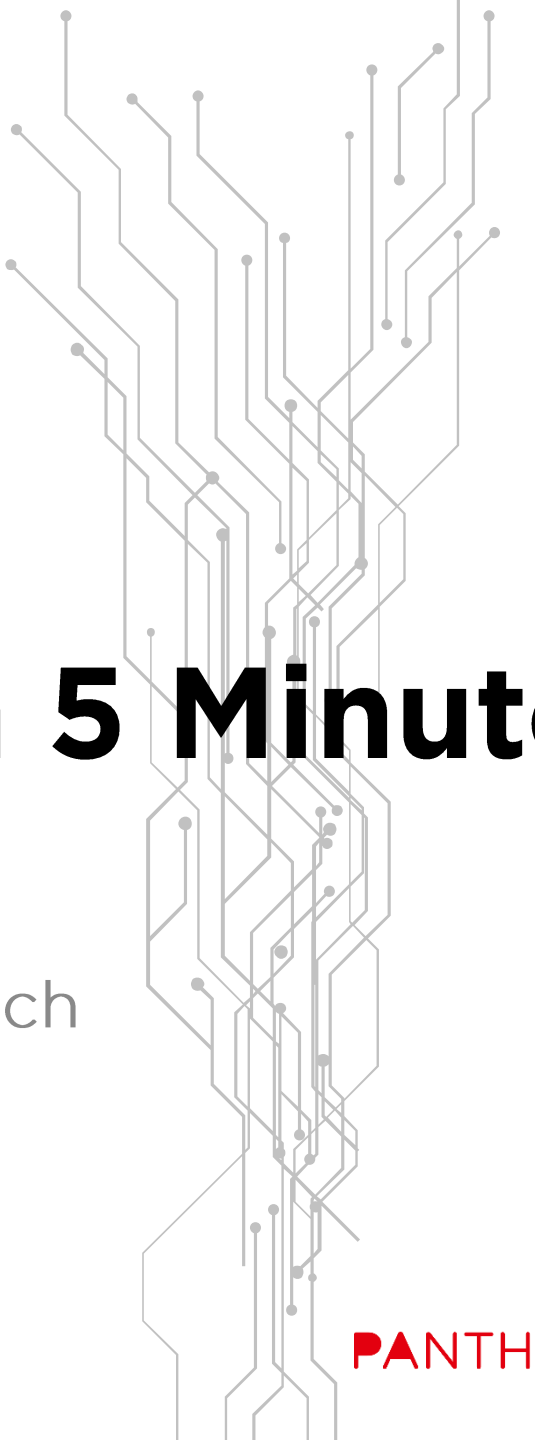
Build & Deploy a CNF in 5 Minutes

Rastislav Szabó, PANTHEON.tech



PANTHEON.tech

What is a CNF?

- Cloud-native Network Function
- Software implementation of network functionality (e.g. IPv4/v6 router, L2 switch, VPN gateway, firewall)
- Built & deployed in a cloud-native way:
  - Packaged into a Docker container
  - Deployed & orchestrated in Kubernetes
  - Configurable using cloud-native APIs (K8s CRDs, gRPC, ...)
  - Chained with other CNFs to provide more complex network functionality (microservices pattern)

How to Build a CNF?

- Data plane: FD.io VPP
- Cloud-native management/control plane: Ligato.io
- (Special) Networking interconnect between CNFs: NetworkServiceMesh.io

PANTHEON .tech

The Simplest CNF / ligato/vpp-agent

```
$ kubectl run cnf --image=ligato/vpp-agent --env="ETCD_CONFIG="

$ kubectl exec -it cnf -- vppctl -s:5002

 _______             _        _    _____  ___
 __/ __/ _ \ (_)__    | |  / / _ \/ _ \
 _/ _// // / / / _ \   | |/ / ___/ ___/
 /_/ /____(_)_/\___/   |___/_/  /_/

vpp# sh inter
              Name               Idx    State  MTU (L3/IP4/IP6/MPLS)
local0                            0     down          0/0/0/0
vpp#
```

PANTHEON.tech

Ligato.io –based CNFs & NSM

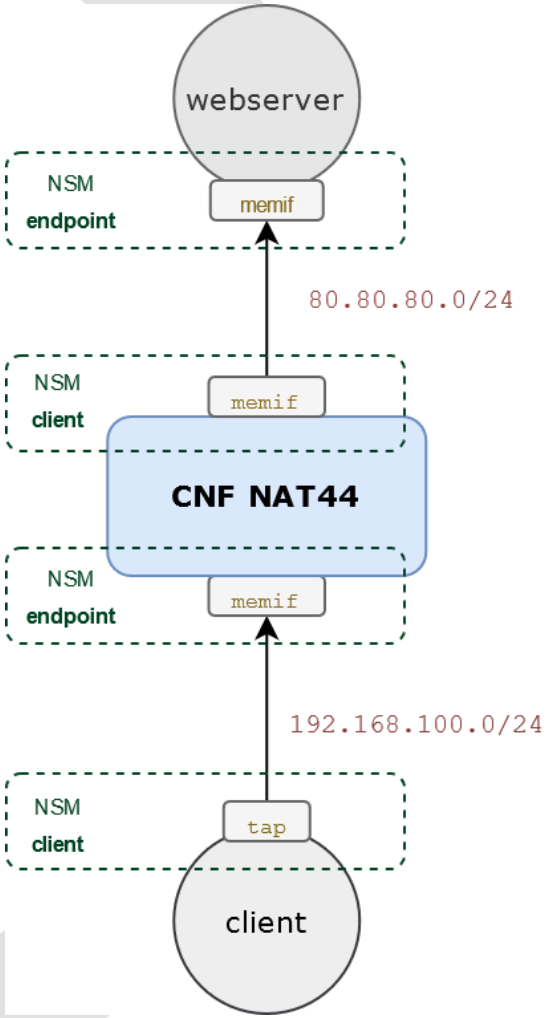
Solution: NSM plugin for Ligato.io CNFs:

- Seamless integration of Ligato.io -based CNFs with NSM
- Connections between CNFs can be defined fully declaratively, without the need for any NSM wiring code in CNFs
- https://github.com/PANTHEONtech/cnf-nsm

- Universal VPP + VPP Agent Docker image: pantheontech/nsm-agent-vpp

PANTHEON.tech

DEMO / NSM Routing




```
apiVersion: networkservicemesh.io/v1alpha1
kind: NetworkService
metadata:
  name: cnf-nat-example
spec:
  payload: IP
  matches:
    # connect client to the (local side of) cnf-nat44
    - match:
      sourceSelector:
        app: client
      route:
        - destination:
          destinationSelector:
            app: nat44
    # connect the (external side of) cnf-nat44
      to the webserver
    - match:
      sourceSelector:
        app: nat44
      route:
        - destination:
          destinationSelector:
            app: webserver
```

PANTHEON.tech

DEMO / NSM & CNF Config

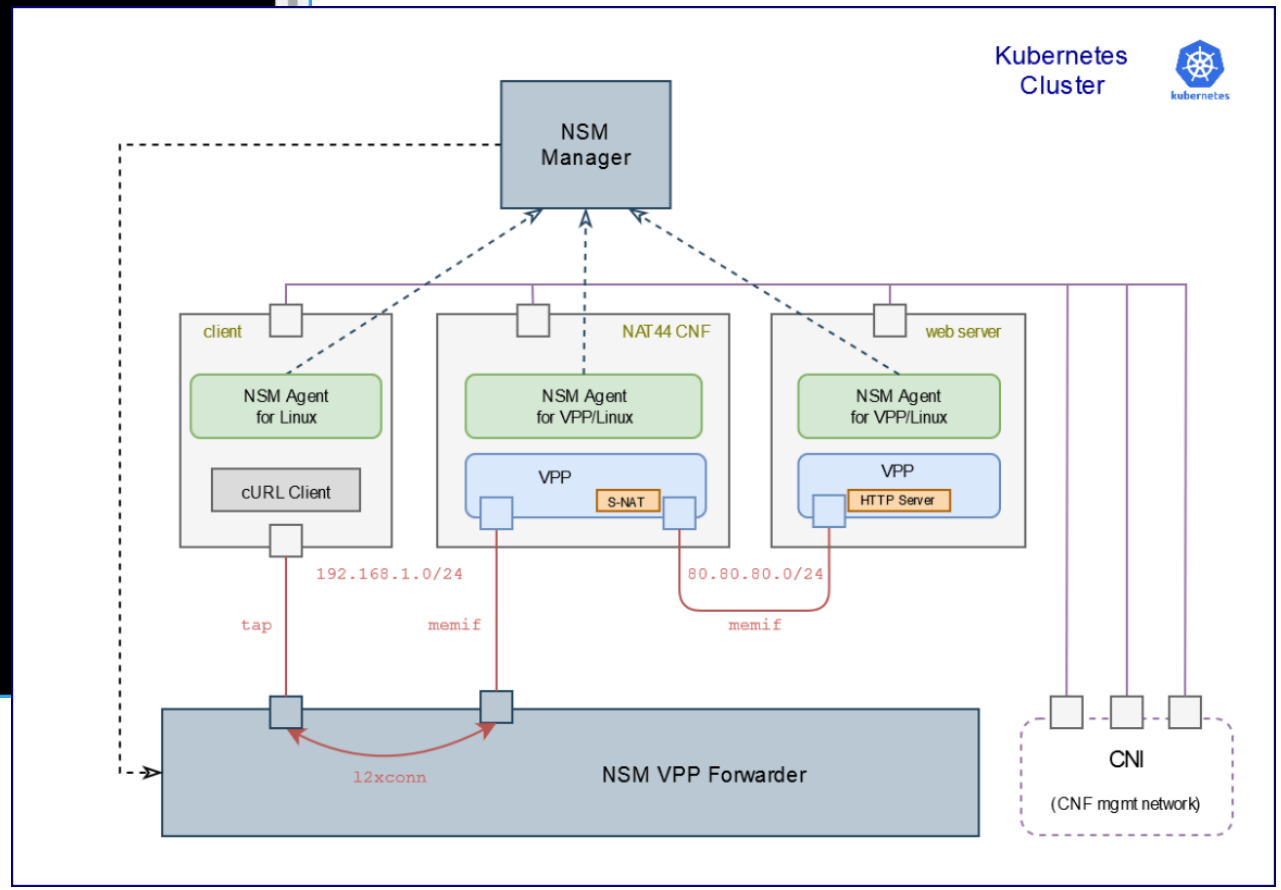
```
apiVersion: pantheon.tech/v1
kind: CNFConfiguration
metadata:
  name: cnf-nat44
spec:
  microservice: cnf-nat44
  configItems:
    - module: cnf.nsm
      version: v1
      type: endpoint
      data: |-
        network_service: cnf-nat-example
        advertised_labels:
          - key: app
            value: nat44
        interface_name_prefix: memif
        interface_type: MEM_INTERFACE
        single_client: true
        ipAddresses:
          - "192.168.100.1/24"

    - module: cnf.nsm
      version: v1
      type: client
      data: |-
        name: access-to-external-network
        network_service: cnf-nat-example
        outgoing_labels:
          - key: app
            value: nat44
        interface_name: memif1
        interface_type: MEM_INTERFACE
        ipAddresses:
          - "80.80.80.100/24"
```

```
    - module: vpp.nat
      type: nat44-interface
      data: |-
        name: memif1
        nat_outside: true
        output_feature: true

    - module: vpp.nat
      type: nat44-interface
      data: |-
        name: memif0
        nat_inside: true

    - module: vpp.nat
      type: nat44-pool
      data: |-
        first_ip: 80.80.80.100
        last_ip: 80.80.80.105
```

PANTHEON.tech

DEMO / Networking

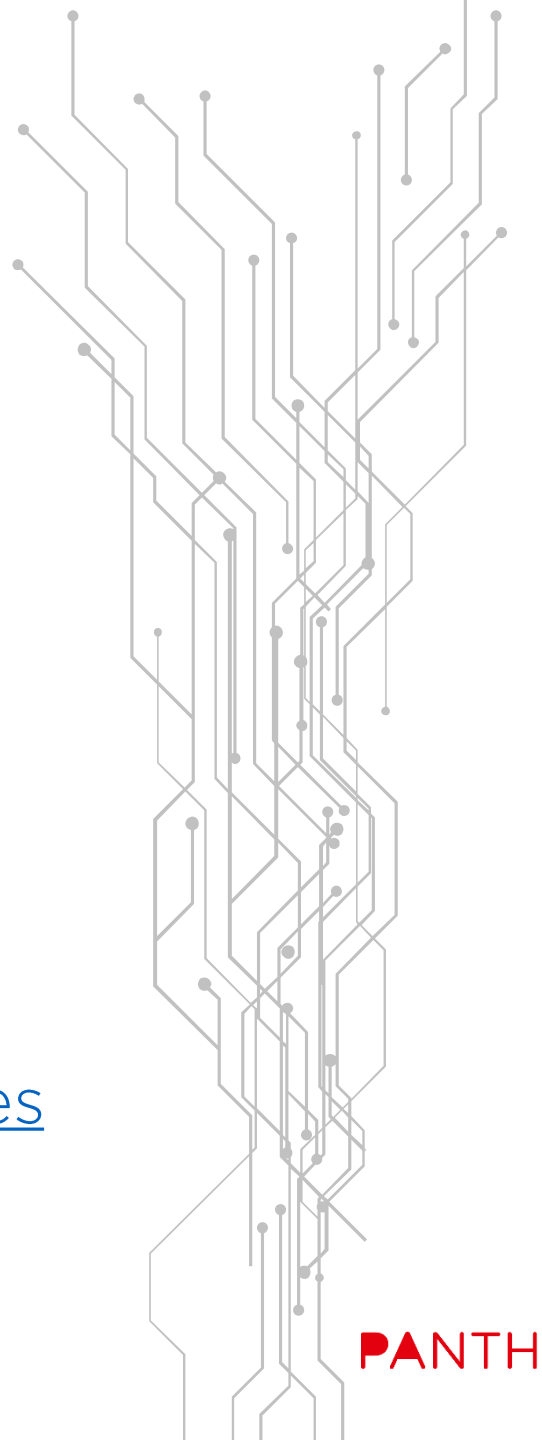
```
File  Actions  Edit  View  Help
rasto@demo:~$ kubectl get pods
```



DEMO

CNFs with FD.io VPP and NSM

https://github.com/PANTHEONtech/cnf-examples



PANTHEON.tech

Thank you!

rastislav.szabo@pantheon.tech

WEB / www.pantheon.tech
MAIL / info@pantheon.tech
LINKEDIN / PANTHEON.tech
TWITTER / @Pantheon_Tech
GITHUB / @PANTHEONtech

PANTHEON .tech