OpenShift Virtualizationのネットワーク構成を真剣に考えてみた

OpenShift Lounge+ "TALKs"
～ Virtのお供スペシャル ～

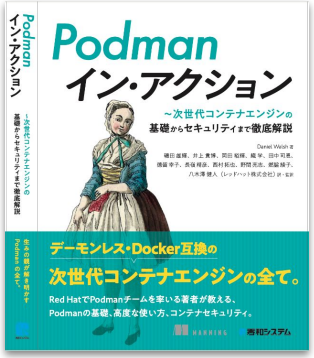
田中司恩（@tnk4on）
shtanaka@redhat.com

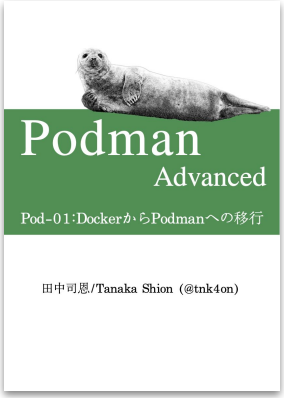
2024/12/12

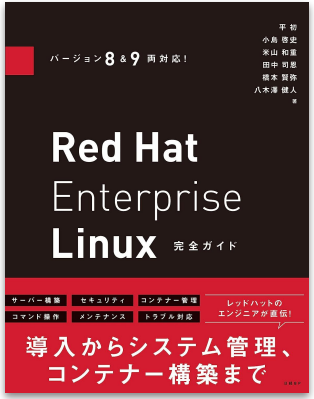
Red Hat

自己紹介

- 田中 司恩（タナカ シオン） / @tnk4on
  - https://tnk4on.github.io/
- Red Hatのソリューションアーキテクト。Red Hatへの転職と同時に福岡に移住。
- 大阪芸術大学音楽工学コース卒。在学中よりメディア・アート、サウンド・アート作品の制作を行う。
- コンテナに興味を持ち、OpenShiftやPodmanに関する情報をブログ、雑誌/書籍に多数投稿。「Podmanイン・アクション」の著者の1人。
- 2020年頃よりPodmanコミュニティに参加、PR/Issue/翻訳などを行い、Podmanの日本語情報発信を積極的に展開中。
- 最近はPodmanを推す人
- 音とテクノロジーが重なる領域が好物のデジタルクリエイター。







Red Hat

ゼロから始めるOpenShift Virtualization

ESXi上に仮想マシンとしてOpenShift Virtualizationの環境を構築する完全ガイド

- 赤帽エンジニアブログに
  - 第1回：OpenShiftのインストール
  - 第2回：OpenShiftインストール後の作業
  - 第3回：共有ストレージの作成（NFS CSIドライバーの構築）
  - 第4回：OpenShift Virtualizationのインストールと実行
  - 第5回：vSphere仮想マシンの移行

- 5分で完全理解！ゼロから始める OpenShift Virtualization

  https://speakerdeck.com/tnk4on/starting-from-zero-openshift-virtualization-at-5-min
  - 第6回 Red Hat Tech Night in RHSC 2024 のLTで登壇した資料

Red Hat

アジェンダ

- OpenShift Virtualizationのネットワークについて理解する
- プライマリネットワーク、セカンダリネットワークの概念
- ブリッジネットワークの構成（物理ネットワークへの接続）
- UDNについて

Red Hat

本資料の前提条件

- OpenShiftについては概ね理解していることとします
- OpenShiftのネットワークについてDeep Diveは行いません。可能な限り理解するのに不要な部分は抽象化し、本内容の理解の推進を優先とします。
- SR-IOVなど、仮想マシンからハードウェアを直接利用する内容は対象外とします
- 本内容には独自の見解が含まれます。内容にツッコミがある場合は個人宛てにご自由にご連絡ください。

Red Hat

本ドキュメント上での略称

- OpenShift Virtualization (OCP-V)
- OVN-Kubernetes (OVN-K)
- NodeNetworkConfigurationPolicy (NNCP)
- NetworkAttachmentDefinition (NAD)
- Open vSwitch (OVS)
- UserDefinedNetwork (UDN)

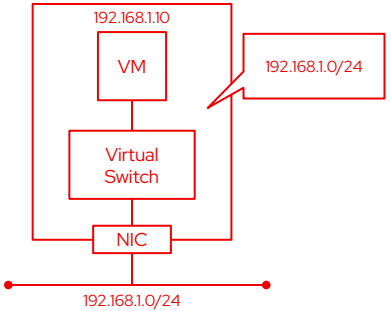
OpenShift Virtualization のネットワークについて理解する

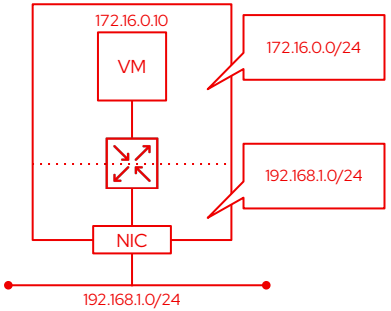
Red Hat

一般的な仮想マシンのネットワークの種類

ブリッジ型とルーティング型

- **ブリッジ型**：
  - 仮想スイッチなどを作成して物理ネットワークと接続
  - 物理ネットワークに直接接続されているように振る舞う
  - 主にハイパーバイザー型仮想環境（vSphere、Hyper-V、等）
- **ルーティング型**：
  - ホストマシンを経由して外部ネットワークに接続
  - 仮想マシンにはプライベートIPアドレスが割り当てられ、外部との通信にはNAT（Network Address Translation）を使用
  - 主にホスト型仮想環境（VMware Workstation/Fusion、UTM、等）

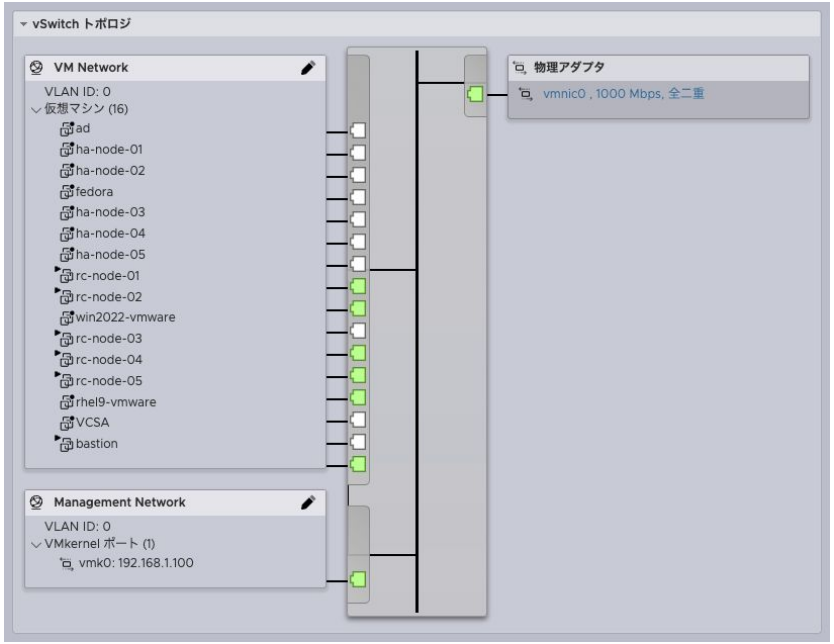


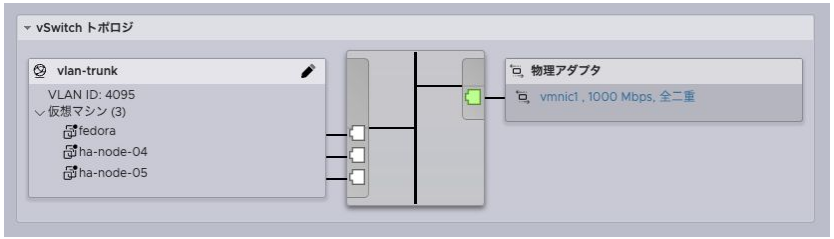


Red Hat

ESXiの標準仮想スイッチ（vSS）

ESXiの標準仮想スイッチ（vSS）はブリッジ型。VLANも使用可能。





Red Hat

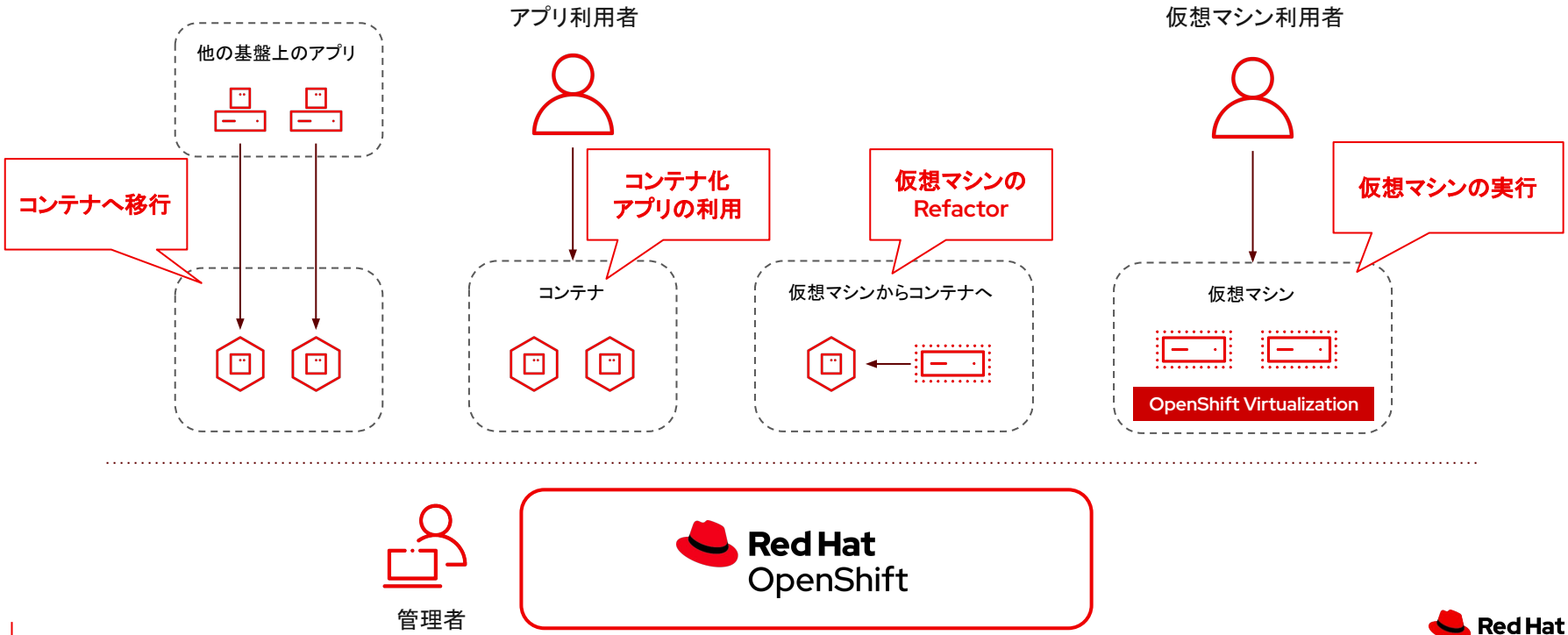
vSphereとOpenShiftのネットワーク機能比較

OpenShiftの各ノード上のOVSはOVN-Kで抽象化されている

特徴	vSS (vSphere)	vDS (vSphere)	OVN-K (OpenShift)
アーキテクチャ	ローカルホストで動作	集中管理型、複数ホスト間の一元管理	分散型、Kubernetes環境向け
スケーラビリティ	限定的	高い	高い
セキュリティ機能	基本的なVLANとトラフィック管理	高度なネットワーク制御とポリシー	高度なネットワーク制御とポリシー
クラウドネイティブ対応	低い (仮想マシン向け)	低い (仮想マシン向け)	高い (コンテナ、Kubernetes向け)
簡易性	簡単	中程度	複雑
管理方法	vSphereコンソールで個別に管理	vCenter経由で集中管理	Kubernetes API経由で管理

vSphere Standard Switch (vSS)
vSphere Distributed Switch (vDS)

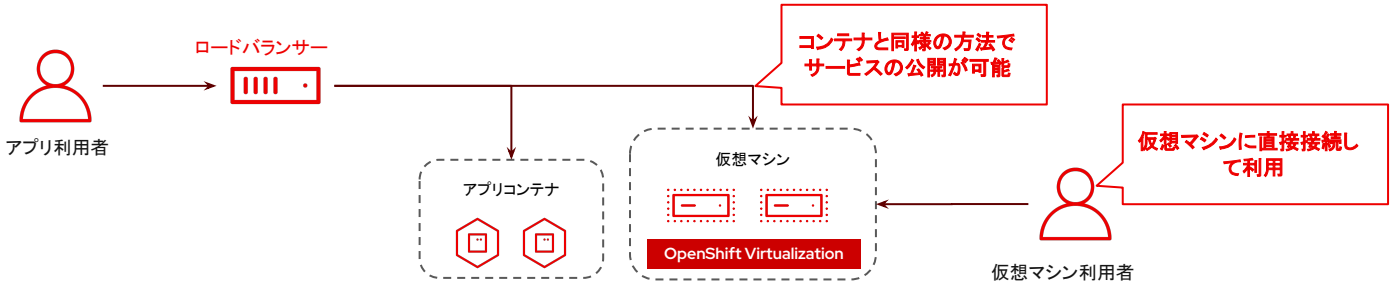
OpenShiftはコンテナと仮想マシンの統合実行環境



OpenShift Virtualizationに期待すること

- vSphereからの移行先としての仮想基盤
- OCP-Vの市場向けメッセージとしては、仮想マシンからコンテナへの移行（Refactor）を目指すことを目的とするが …

→実際にはお客様やパートナー様が最初に期待することは仮想基盤としての利用目的がほとんど



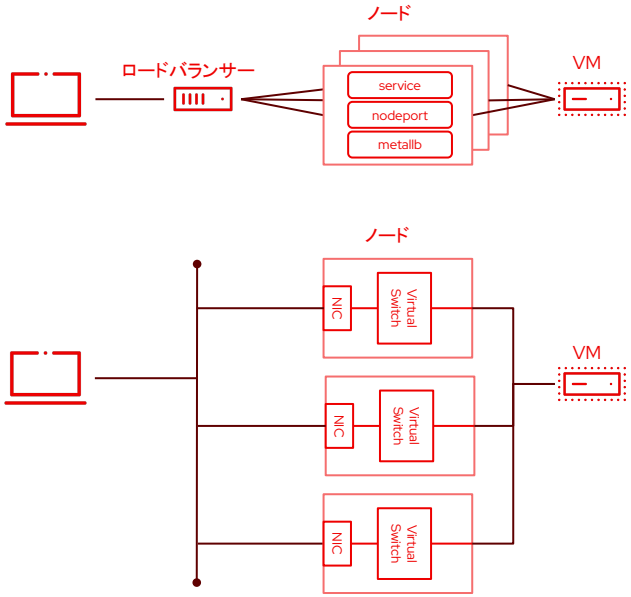
Red Hat

OpenShift上の仮想マシンへのアクセス方法

コンテナ的アプローチと仮想マシン的アプローチ

- **コンテナ的アプローチ** ：
  - RouteとService、NodePort、MetalLB等を利用
- **仮想マシン的アプローチ** ：
  - ブリッジ経由で仮想マシンと直接接続

一般的にはこういうことを期待。vSphereからの仮想基盤の代替としてはこちらの方が利用しやすい



プライマリネットワーク、セカンダリネットワークの概念

Red Hat

OpenShiftのデフォルトネットワーク

デフォルトで使用するのは Podネットワーク

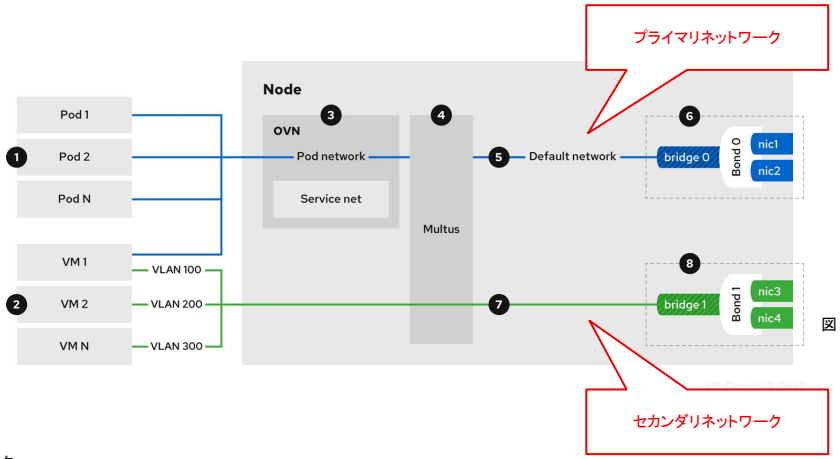
- OpenShiftのコンテナのデフォルト接続先は「 Podネットワーク」
  →OCP-Vで作成した仮想マシンも同様
- Podネットワークとは、
  - OpenShiftクラスター内のすべての Podが相互通信可能な仮想ネットワーク
  - コンテナや仮想マシンが統一されたネットワーク空間を共有し、シンプルな通信設計が可能になる
  - OVN-Kubernetesを使用（OpenShift SDNはv4.15で削除）

セカンダリネットワーク

Podネットワークがプライマリ、それ以外がセカンダリネットワーク

- OpenShiftにはセカンダリネットワークという概念がある。
- Multusを使って複数のCNIをPodにアサインできる

→複数の CNI の存在を可能にし、Pod または仮想マシンが必要なインターフェイスを使用できるようにするメタ" CNI プラグイン。



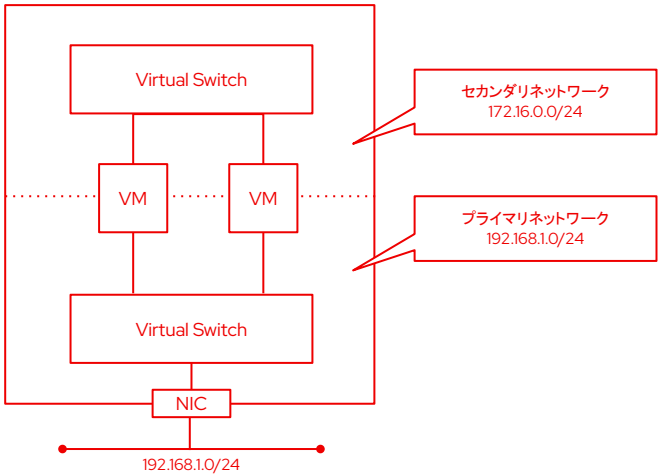
図：OpenShift Virtualization ネットワークの概要（製品ドキュメント）

製品ドキュメント：第8章 ネットワーク
https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/virtualization/networking#virt-networking

一方、一般的な仮想マシンの世界では...

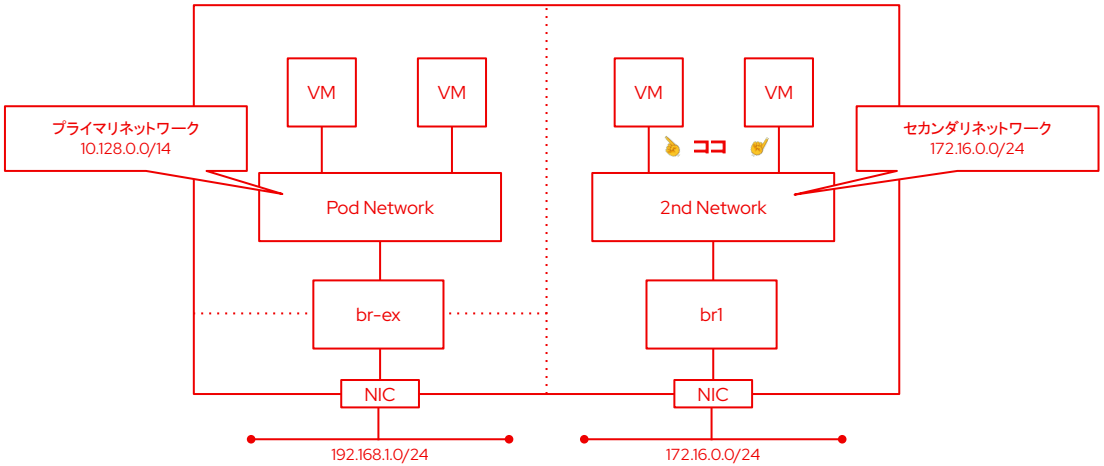
- プライマリネットワーク:
  - 仮想マシンの本IPアドレスが設定される
- セカンダリネットワーク
  - 内部通信や特定用途で利用される

プライマリ/セカンダリネットワークともに、個々の環境に合わせて自由にネットワークアドレス、IPアドレスの設定が可能



デフォルトの設定を無視すれば、普通にこういうことも可能

VMのプライマリNICにOpenShiftのセカンダリーネットワークをアサイン



ただし、下記の点に注意
※ virtctl ssh, virtctl scp を使って仮想マシンに直接接続できなくなる
※ readiness プローブと liveness プローブを使用したヘルスチェックができなくなる

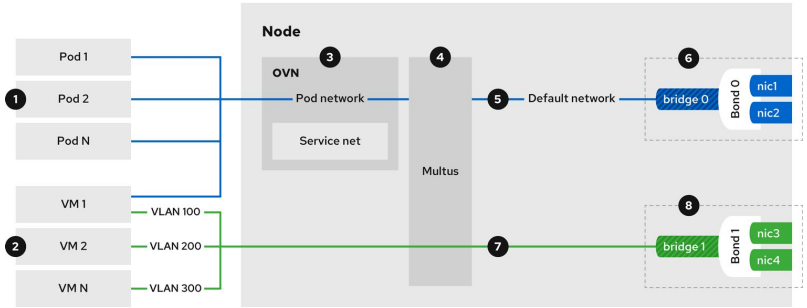
Red Hat

ブリッジネットワークの構成（物理ネットワークへの接続）

Red Hat

物理ネットワークを利用する2つの構成方法

- OpenShiftのデフォルトのネットワークは「 Podネットワーク」を使用しますが、別のネットワーク（セカンダリーネットワーク）に接続することが可能です
- OpenShift v4.17（2024/11/12時点）では、仮想マシンから物理ネットワークを利用する方法として、「 Linuxブリッジ」と「OVN-Kubernetes ローカルネット」の 2つの構成方法があります※



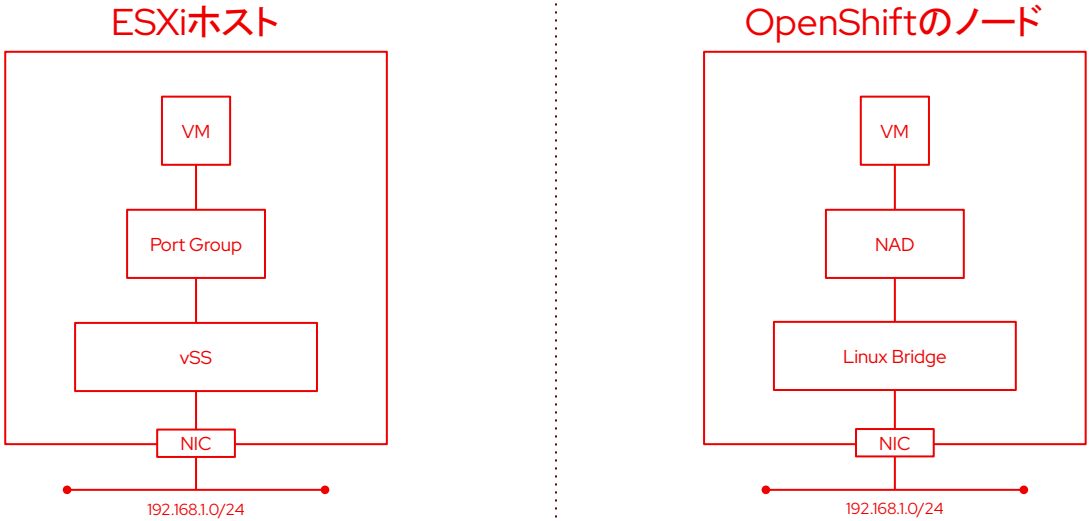
図：OpenShift Virtualization ネットワークの概要（製品ドキュメント）

※ SR-IOVは目的が異なるので本ドキュメントでは対象外

製品ドキュメント：第8章 ネットワーク
https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/virtualization/networking#virt-networking

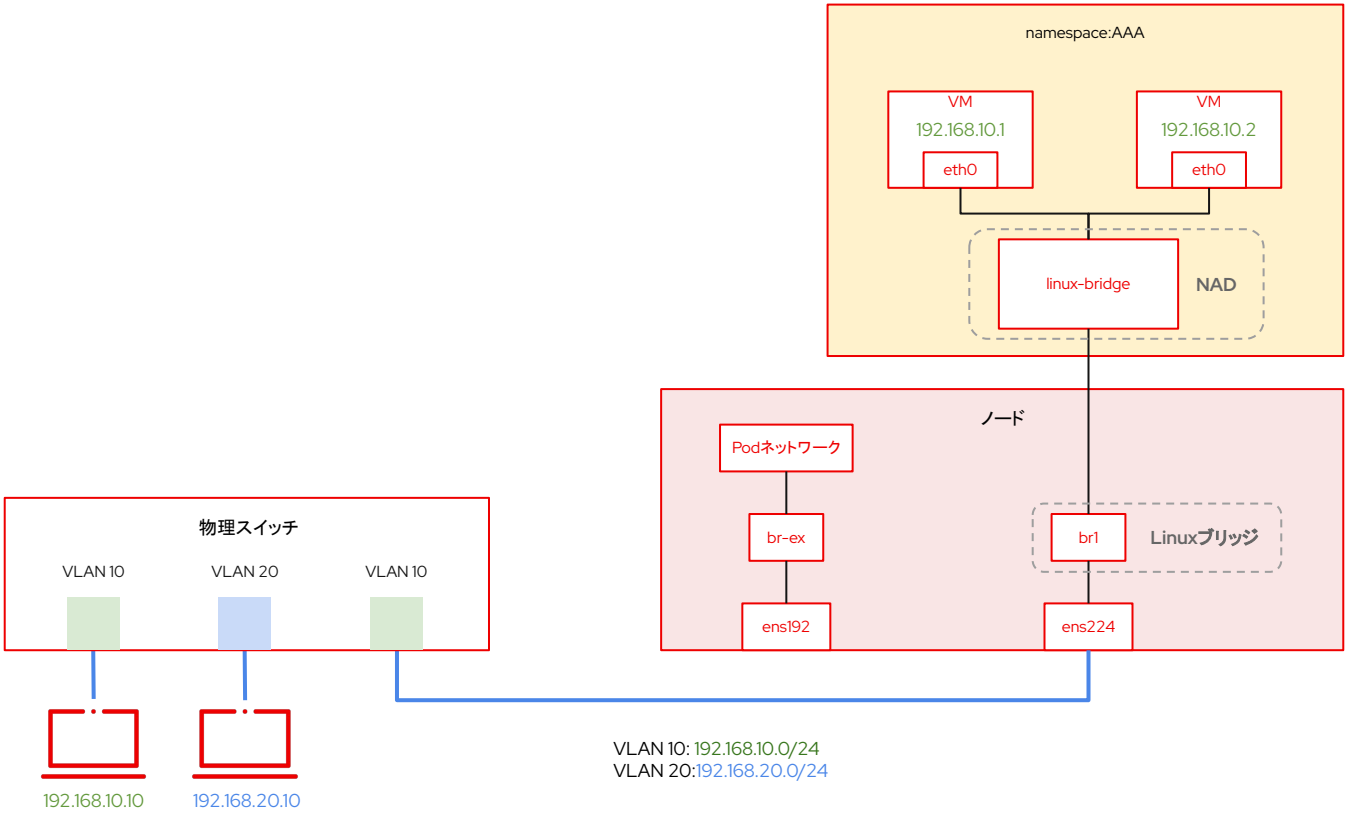
Red Hat

vSSとLinuxブリッジとの比較



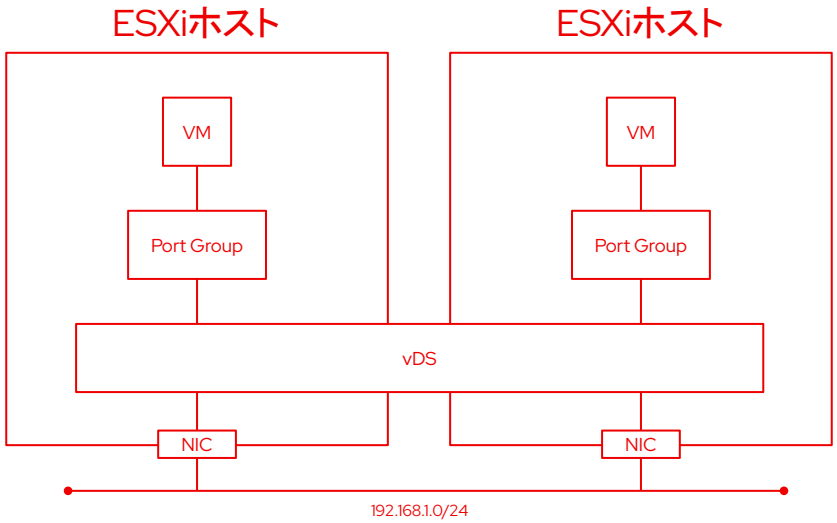
Red Hat

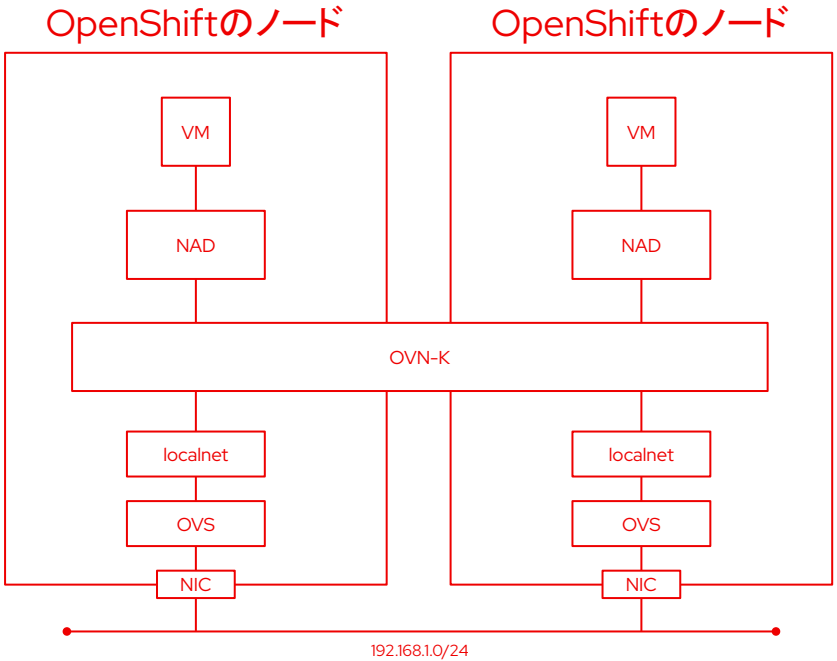
Linuxブリッジの構成概要



Red Hat

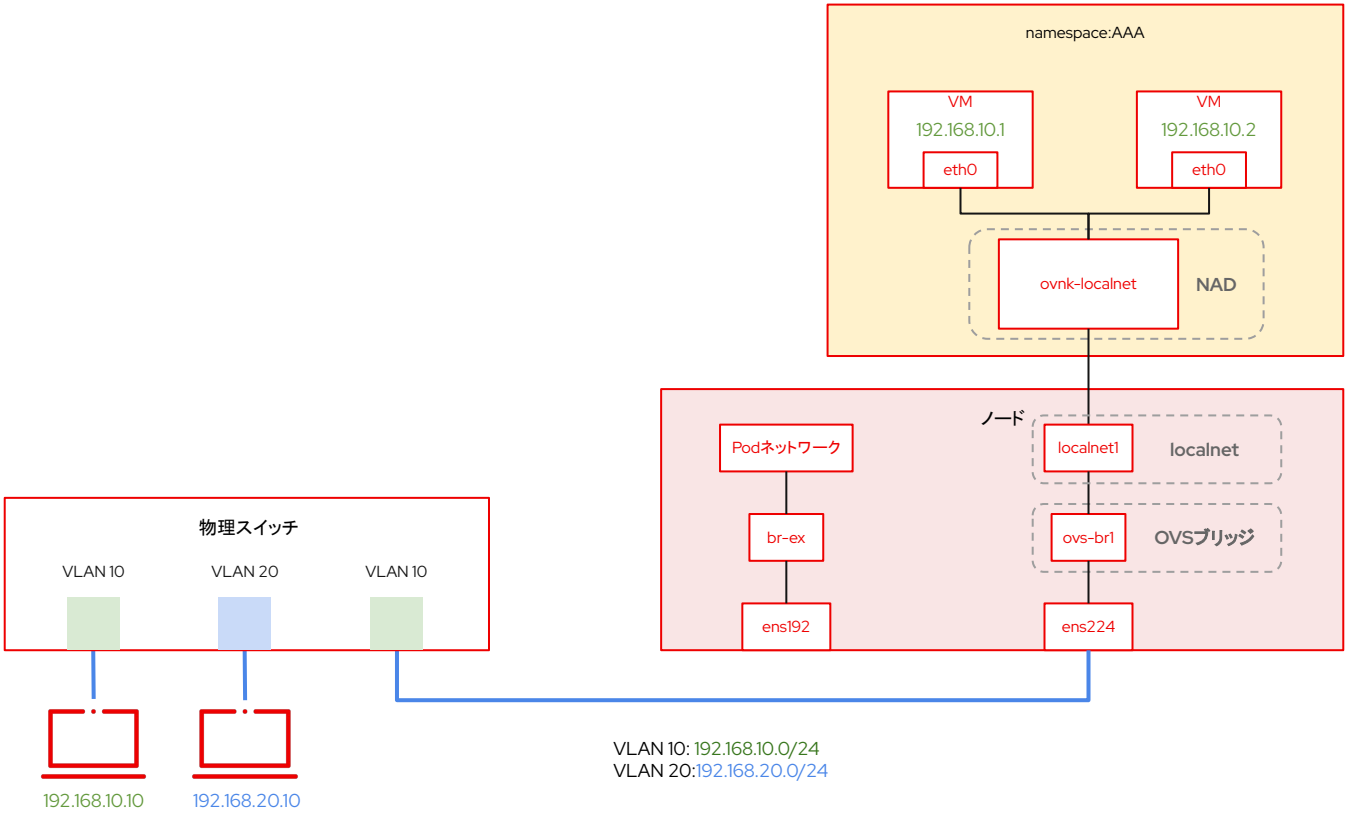
vDSとOVN-Kubernetesローカルネットとの比較



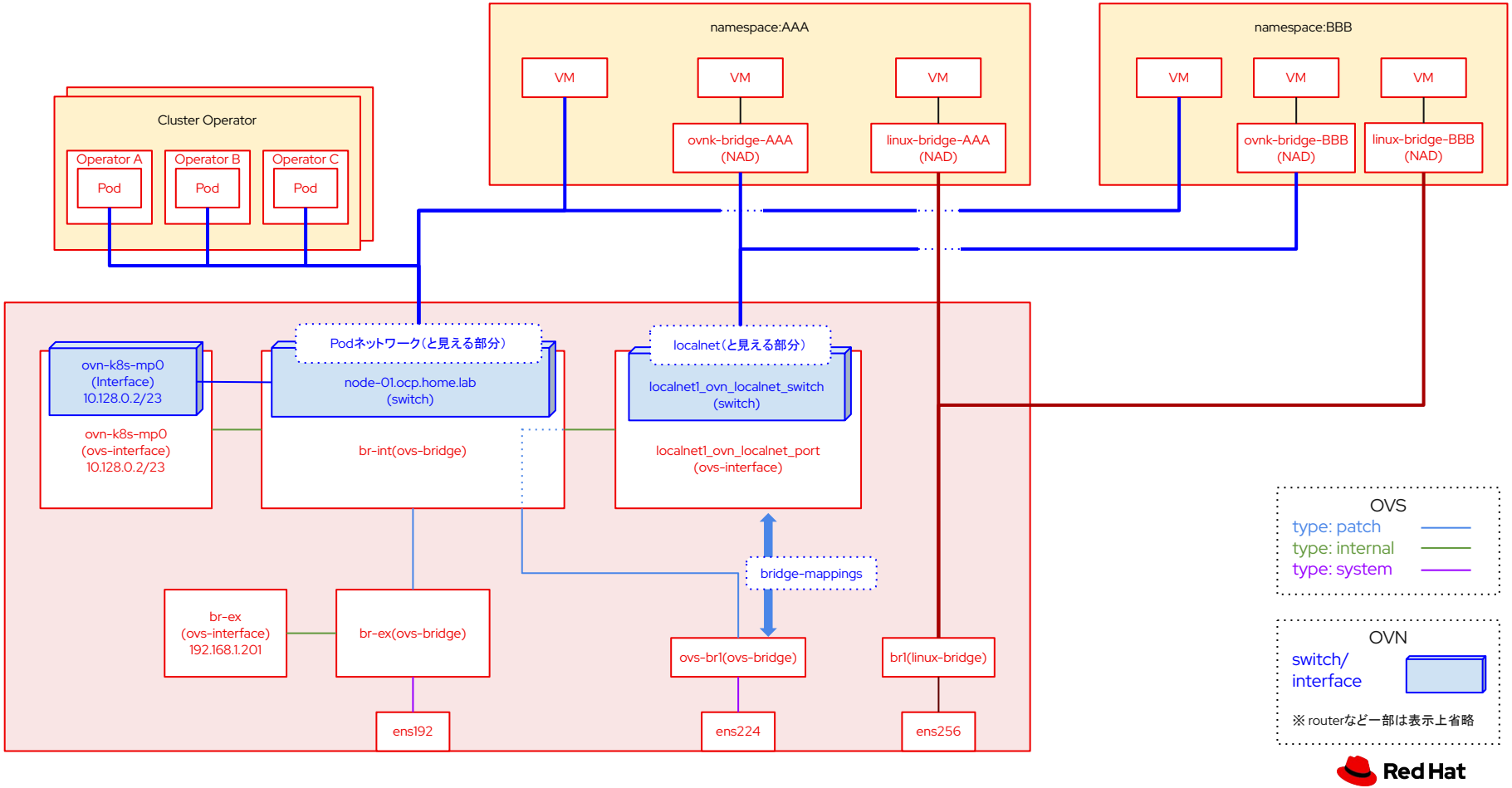


Red Hat

OVN-K localnetの構成概要



LinuxブリッジとOVN-K localnetのブリッジ構成詳細



2つの構成方法の比較

機能	Linux ブリッジ	OVN-Kubernetes ローカルネット	備考
ブリッジの種類	Linuxブリッジ	OVSブリッジとOVNローカルネットをブリッジマッピング	
異なるノード上のVMとの通信方法	物理ネットワークを経由	OVNオーバーレイネットワークを経由	
NADでのVLANの利用	はい	はい	VMはNADでuntag後のネットワークに接続
VM内でのタグVLANの利用	はい	いいえ	VM内でVLANインターフェースの作成を行うケース
ネットワークポリシー	いいえ	はい（マルチネットワークポリシー）	
管理された IP プール	いいえ	はい（※whereaboutsは利用不可）	
MAC スプーフィングフィルタリング	はい	はい	

Linux ブリッジ CNI と OVN-Kubernetes ローカルネットトポロジーの比較
https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/virtualization/networking#comparing-bridge-localnet

Red Hat

VLANを使用する構成

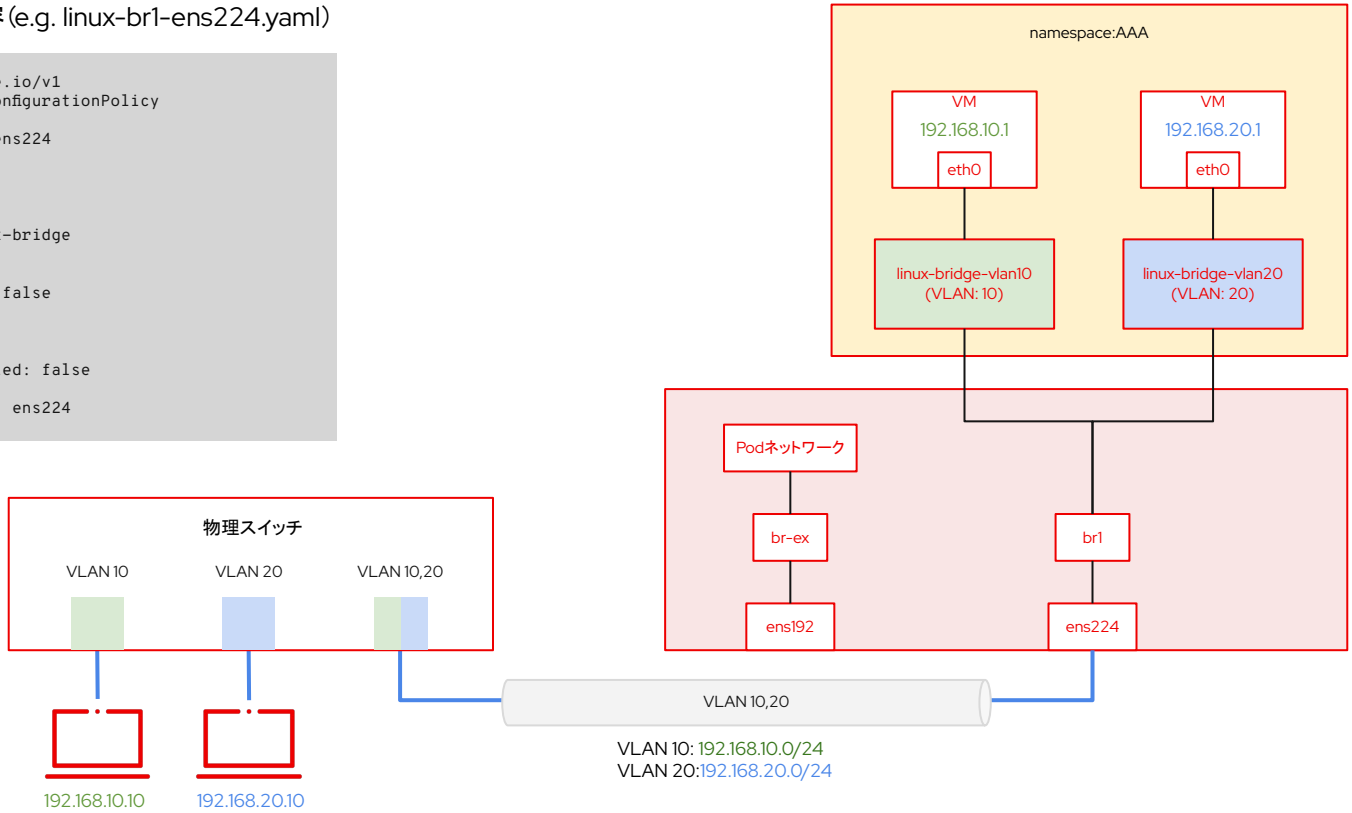
LinuxブリッジとOVN-K localnetで若干の機能差がある

- NADにVLANを設定する構成
  - Linuxブリッジ：OK
  - OVN-K localnet：OK（localnetを分ける必要あり）
- VM内までタグVLANを通す構成
  - Linuxブリッジ：OK
  - OVN-K localnet：NG

LinuxブリッジでNADにVLANを設定する構成

NADの設定内容(e.g. linux-br1-ens224.yaml)

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: linux-br1-ens224
spec:
  desiredState:
    interfaces:
      - name: br1
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        bridge:
          options:
            stp:
              enabled: false
          port:
            - name: ens224
```

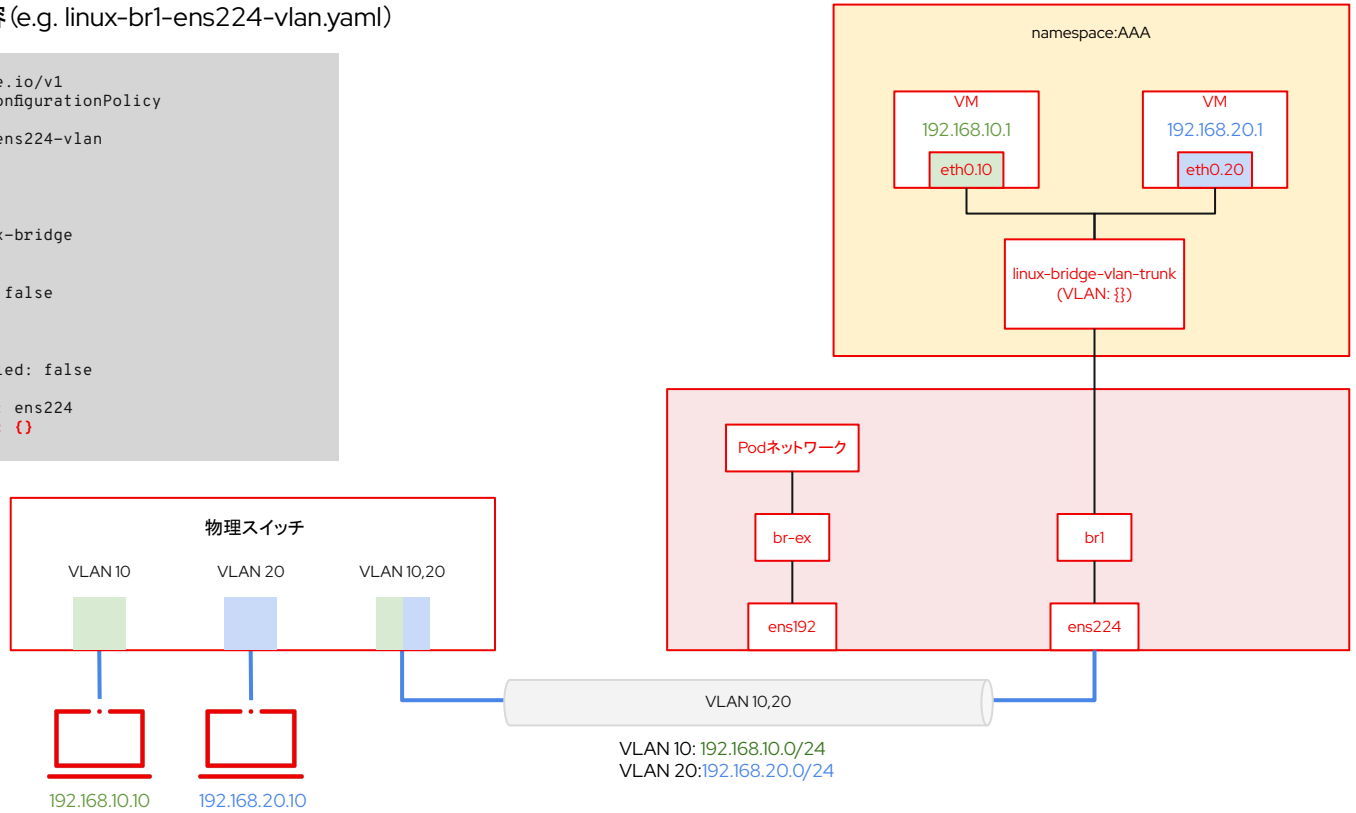


Red Hat

LinuxブリッジでVM内までタグVLANを通す構成

NADの設定内容(e.g. linux-br1-ens224-vlan.yaml)

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: linux-br1-ens224-vlan
spec:
  desiredState:
    interfaces:
      - name: br1
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        bridge:
          options:
            stp:
              enabled: false
          port:
            - name: ens224
              vlan: {}
```



Why a virtual NIC in trunk mode is not working in OpenShift Virtualization? - Red Hat Customer Portal
https://access.redhat.com/solutions/7039373

Red Hat

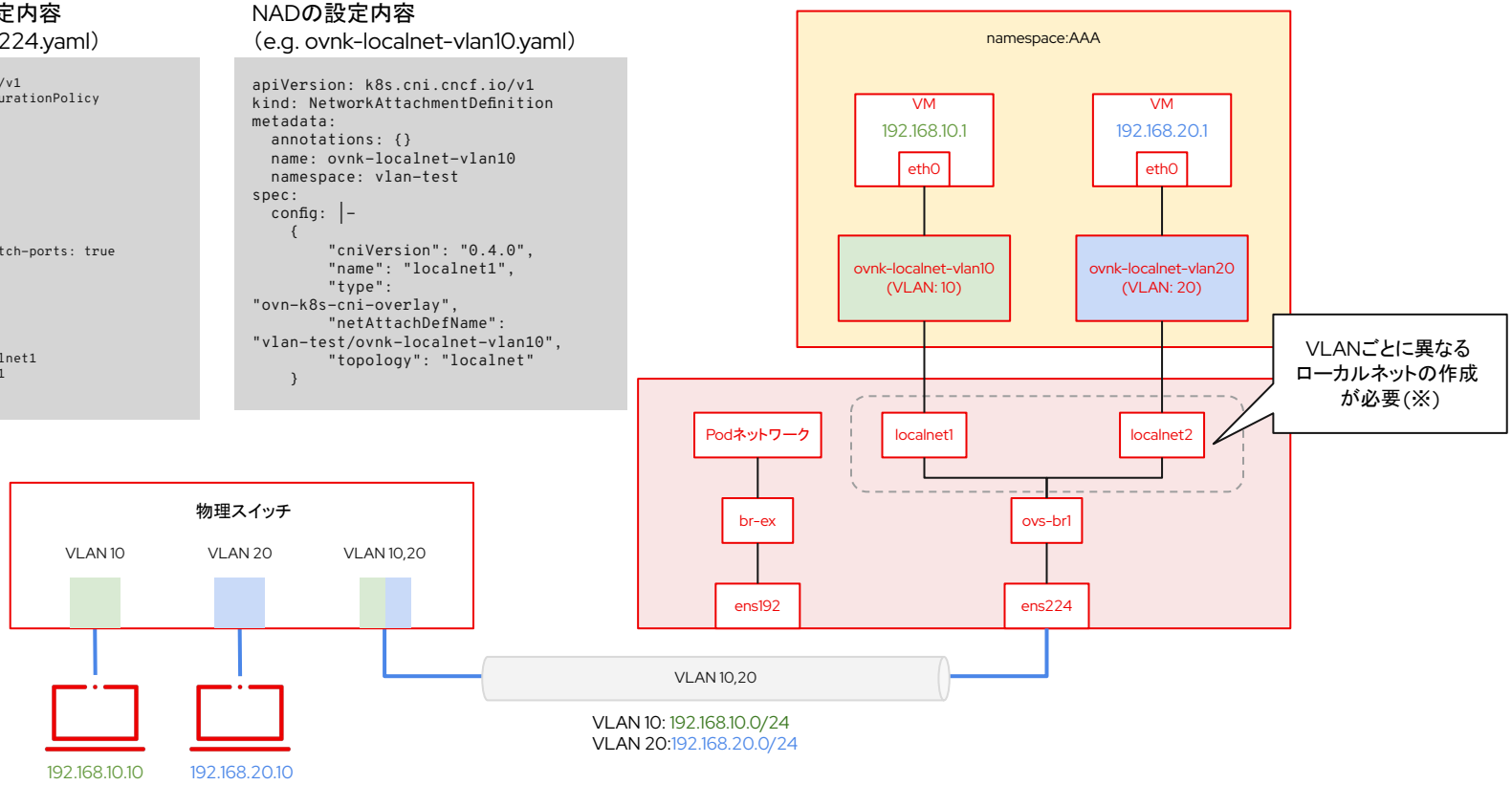
OVN-K localnetでNADにVLANを設定する構成

OVSブリッジの設定内容
（e.g. ovs-br1-ens224.yaml）

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ovs-br1-ens224
spec:
  desiredState:
    interfaces:
    - name: ovs-br1
      type: ovs-bridge
      state: up
      bridge:
        allow-extra-patch-ports: true
        options:
          stp: false
        port:
        - name: ens224
    ovn:
      bridge-mappings:
      - localnet: localnet1
        bridge: ovs-br1
        state: present
```

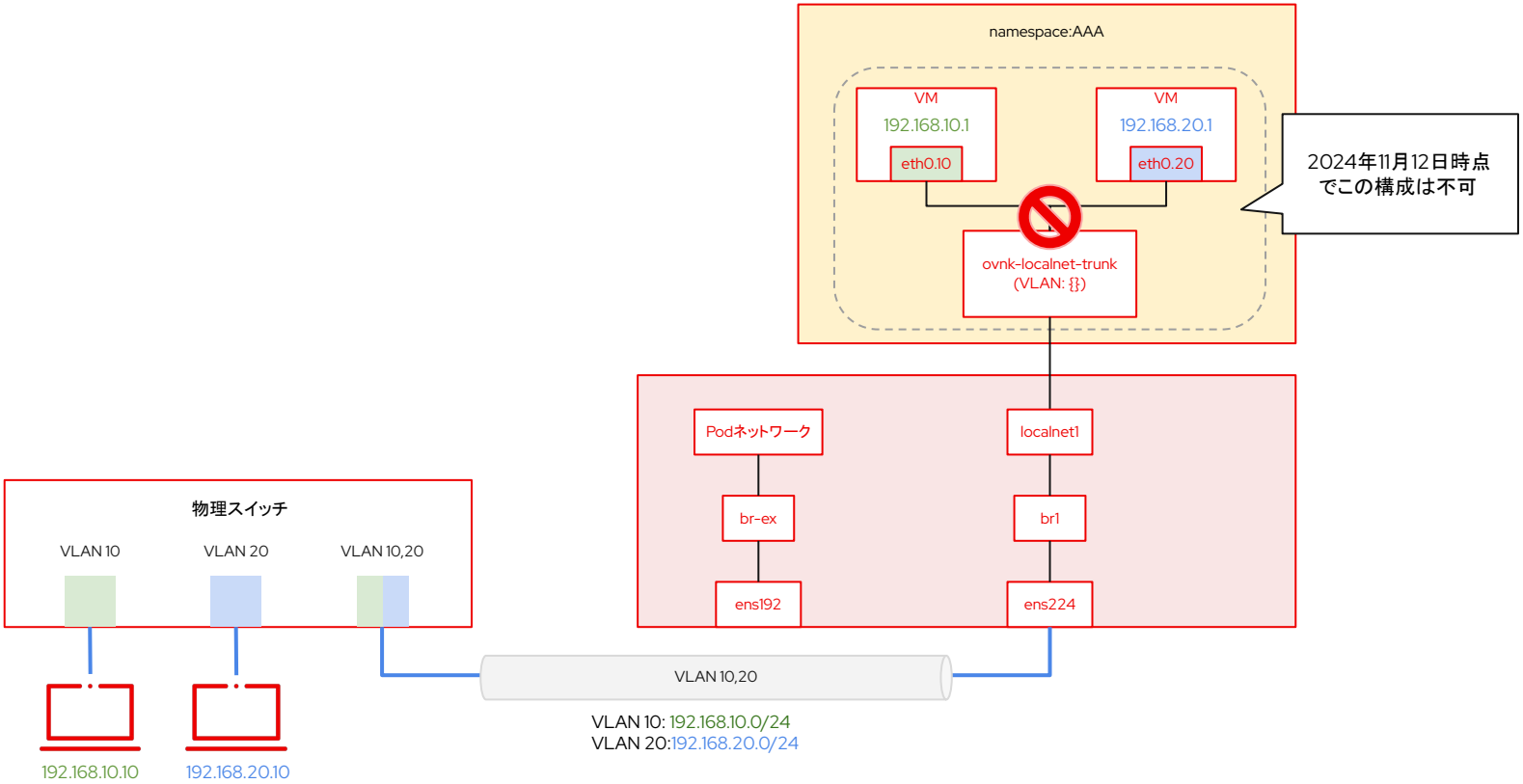
NADの設定内容
（e.g. ovnk-localnet-vlan10.yaml）

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations: {}
  name: ovnk-localnet-vlan10
  namespace: vlan-test
spec:
  config: |-
    {
        "cniVersion": "0.4.0",
        "name": "localnet1",
        "type":
"ovn-k8s-cni-overlay",
        "netAttachDefName":
"vlan-test/ovnk-localnet-vlan10",
        "topology": "localnet"
    }
```



※ Error connecting a VM to a new VLAN in an existing OVN secondary network with localnet topology - Red Hat Customer Portal
https://access.redhat.com/solutions/7078377

（NG）OVN-K localnetでVM内までタグVLANを通す構成



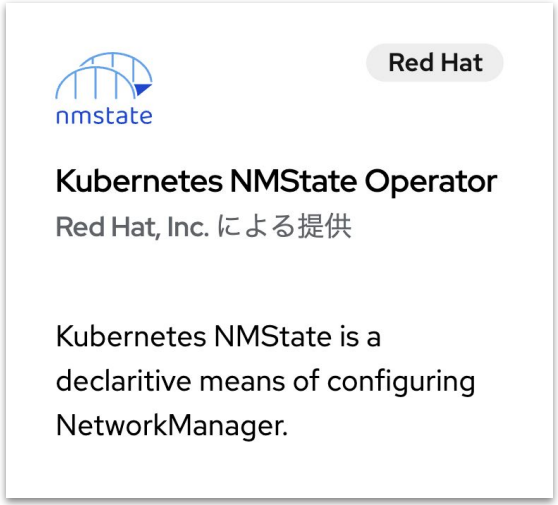
LinuxブリッジとOVN-K localnetの構成方法

- 実装方法は本資料の Appendixを参照
- 設定方法はNMState Operatorを使う。RHELのNMStateと構成方法は同じ。自動で全ノードの設定をやってくれる。元に戻すのも簡単。CLI、GUIあり。
- OVN-Kubernetes セカンダリーネットワーク のローカルネットトポロジーを利用する場合は、OpenShift Virtualization v4.15以上であること(※)

※ https://docs.redhat.com/ja/documentation/openshift_container_platform/4.15/html/virtualization/release-notes#virt-4-15-networking

NMState Operator

- **ネットワーク設定の簡略化** ：YAML形式でネットワーク設定を 記述し（NNCP）、ノードの静的IP設定やブリッジ作成などを自動適用可能
- **宣言的管理** ：現在の状態と希望状態を比較し、意図したネットワーク構成を維持。
- **容易な導入** ：Operator Hubから簡単にOperatorのインストールが可能



UDNについて

Red Hat

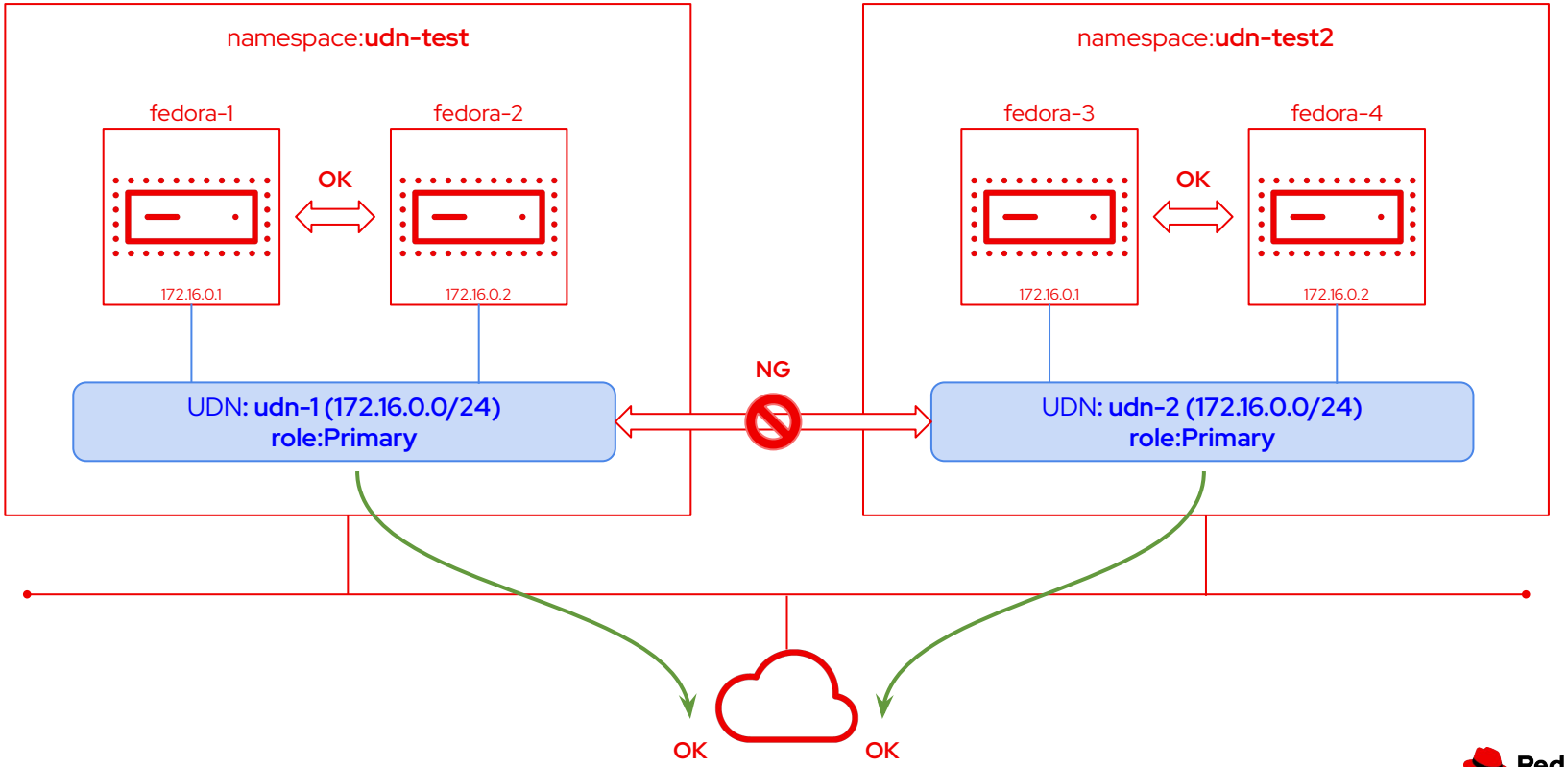
UserDefinedNetwork (UDN)とは

日本語訳ではユーザー定義ネットワーク

- OpenShift v4.17からの追加機能。現時点では TP。
- 論理的に分離されたネットワークを構成できる。
  - マルチテナントの作成が可能
  - 同一クラスター上で、重複したネットワークアドレス、IPアドレスの使用が可能
- 作成したUDN間は独立しており、デフォルトで通信ができない
  - UDNを使わない場合はNetworkPolicyを設定する必要がある
- L2 UDNとL3 UDNがある
- L3 UDNではノードごとにレイヤー 2 セグメントが作成され、それぞれに異なるサブネットが割り当てらる
- Linuxブリッジ、OVN-K localnetと同様に、セカンダリネットワークとしてアサイン可能

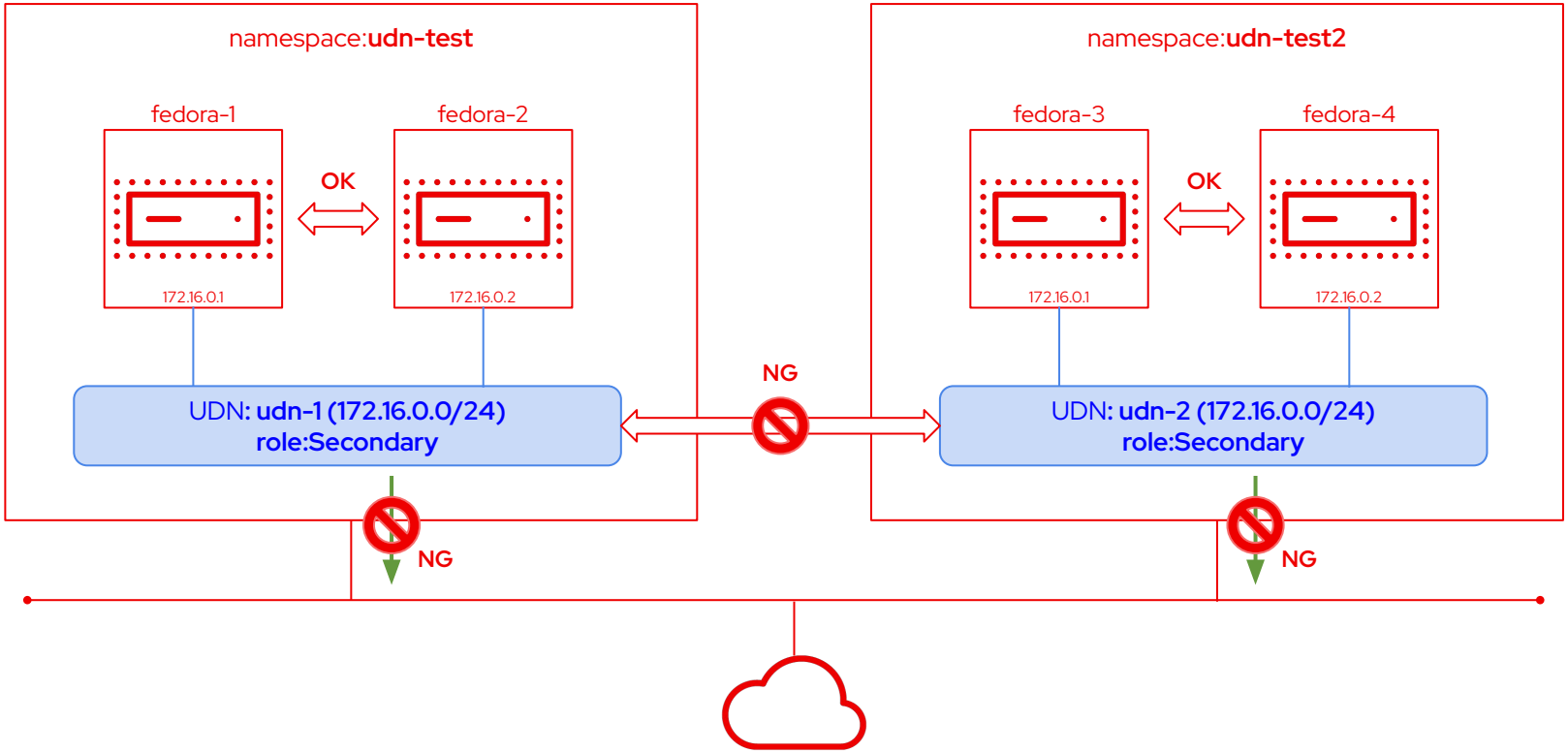
https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/networking/understanding-user-defined-networks

同じネットワークサブネット範囲を持つ L2 UDNを作成した構成例（role:Primary）



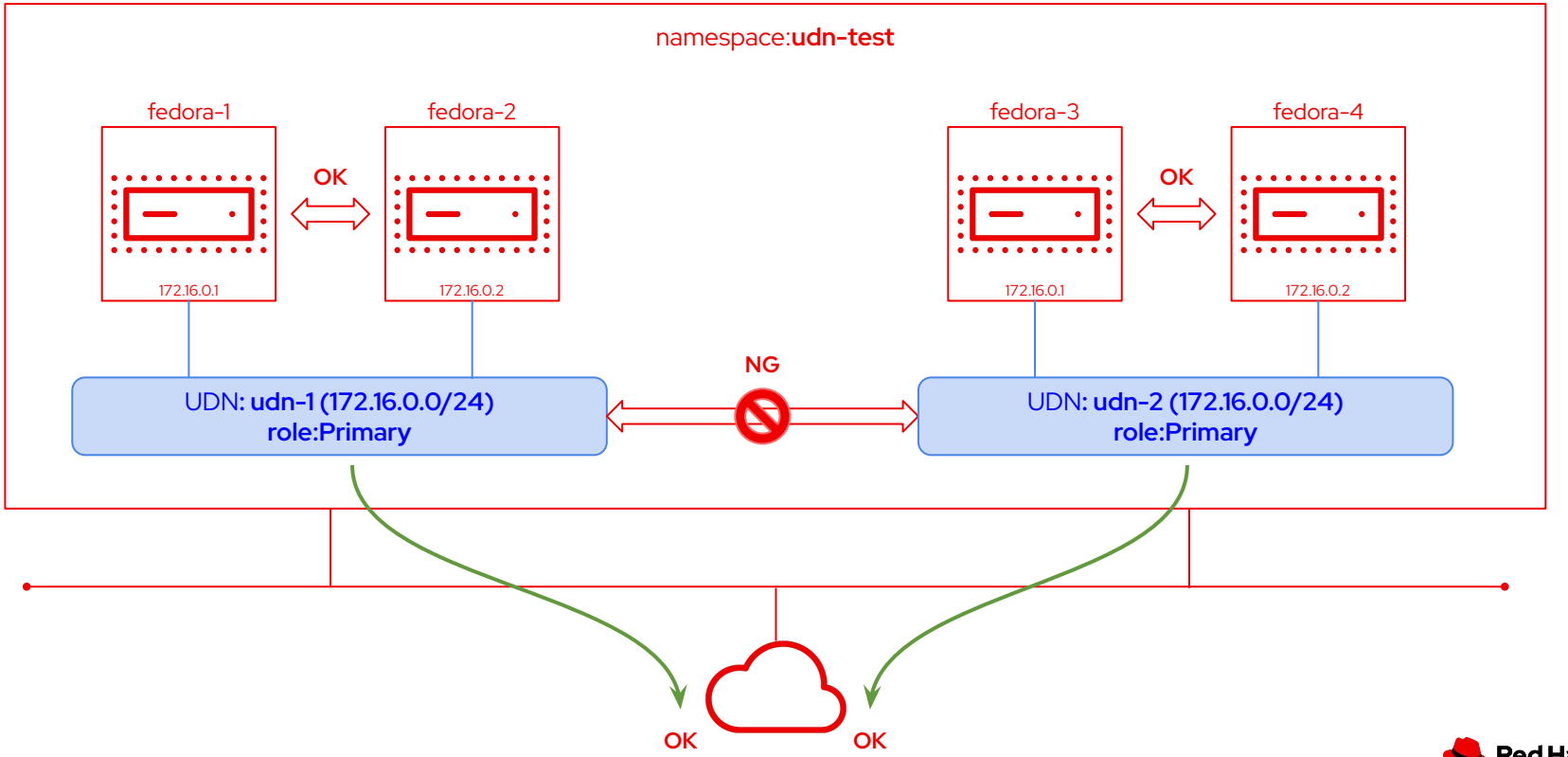
Red Hat

同じネットワークサブネット範囲を持つ L2 UDNを作成した構成例（role:Secondary）



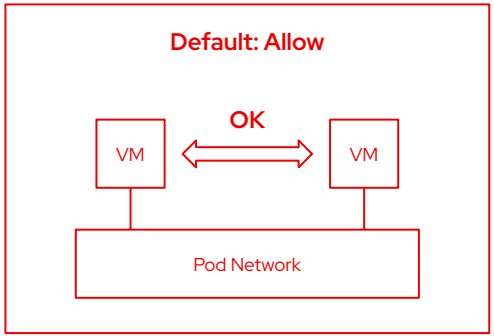
Red Hat

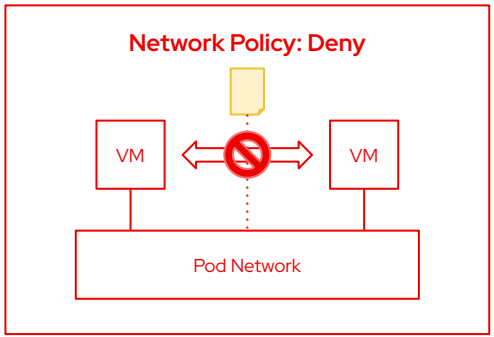
同一ネームスペース内に、異なるネットワークサブネット範囲を持つ　L2 UDNを作成した構成例



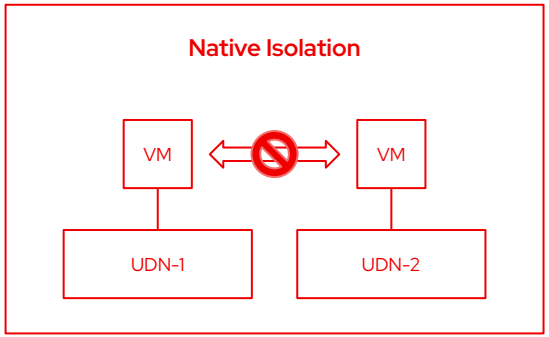
UDNとPodネットワークの違い

Podネットワーク





UDN



Red Hat

OpenShift VirtualizationでUDNを有効化する方法

OpenShift v4.18.0-rc.1、2024/12/9時点の内容

OCP-VでUDNを利用するには下記の作業が必要です

- Feature Gateを使用したTP機能の有効化
- OCP-Vのインストール
- UDNの作成

（OpenShift v4.17.4およびv4.18.0-rc.1で確認済み）

TP機能を有効化するとクラスターのアップグレードができなくなる

必ず壊れても良い検証用クラスターで実施すること

製品ドキュメント:
8.7. フィーチャーゲートを使用した機能の有効化
https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/nodes/nodes-cluster-enabling

Red Hat

TP機能を有効化すると元に戻せなくなるので注意

必ず壊れても良い検証用クラスターで実施すること

警告

クラスターで `TechPreviewNoUpgrade` 機能セットを有効にすると、元に戻すことができず、マイナーバージョンの更新が妨げられます。本番クラスターでは、この機能セットを有効にしないでください。

```
$ oc patch featuregate cluster --type=merge -p '{"spec":{"featureSet":""}}'
The FeatureGate "cluster" is invalid: spec.featureSet: Invalid value: "string": TechPreviewNoUpgrade may not be changed
```

製品ドキュメント:
8.7. フィーチャーゲートを使用した機能の有効化
https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/nodes/nodes-cluster-enabling

Red Hat

既知の問題（2024/12/9時点）

- VM作成時、`cloudInitNoCloud` で静的IPアドレスを指定しても自動割り当てされる（該当製品ドキュメント ※1）
- L2 UDN作成時、`subnets` の指定を外すとエラーになる（ GitHubのソース ※2）
- `ipamLifecycle: Persistent` を指定したL2 UDNを作成してもIPアドレスが保持されない（ =VM再起動でIPアドレスが変わる）。（該当製品ドキュメント ※3）
- OpenShift v4.17ではL2 UDN(Primary)で外部ネットワークに通信ができない（ OpenShift v4.18では問題解消）
- OpenShift v4.17ではロールにSecondaryが指定できない（OpenShift v4.18では問題解消）
- GUIでVMに作成済みのUDNを指定できない（passtが指定できない）

※1 https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/virtualization/virt-configuring-viewing-ips-for-vms#virt-configuring-ip-vm-cli_virt-configuring-viewing-ips-for-vms
※2 https://github.com/openshift/ovn-kubernetes/blob/release-4.17/go-controller/pkg/crd/userdefinednetwork/v1/types.go#L141
※3 https://docs.redhat.com/ja/documentation/openshift_container_platform/4.17/html/networking/understanding-user-defined-networks#limitations-for-udn_understanding-user-defined-networks

UDNの今後と期待

- UDNは絶賛開発中。OpenShiftのアップデートとともに機能も進化していくはず。
- 今後はOCP-VによるIaaSサービスの提供が容易になる
- マルチテナント以外の便利なユースケースについてはまだまだ検証が必要
- UDNはルーティング型。仮想マシンを物理ネットワークに直接接続したい場合は、 Linuxブリッジまたは OVN-Kローカルネットとの使い分けは今後も必要。
- 実力は未知数だが、可能性は無限大

Thank you


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

Appendix

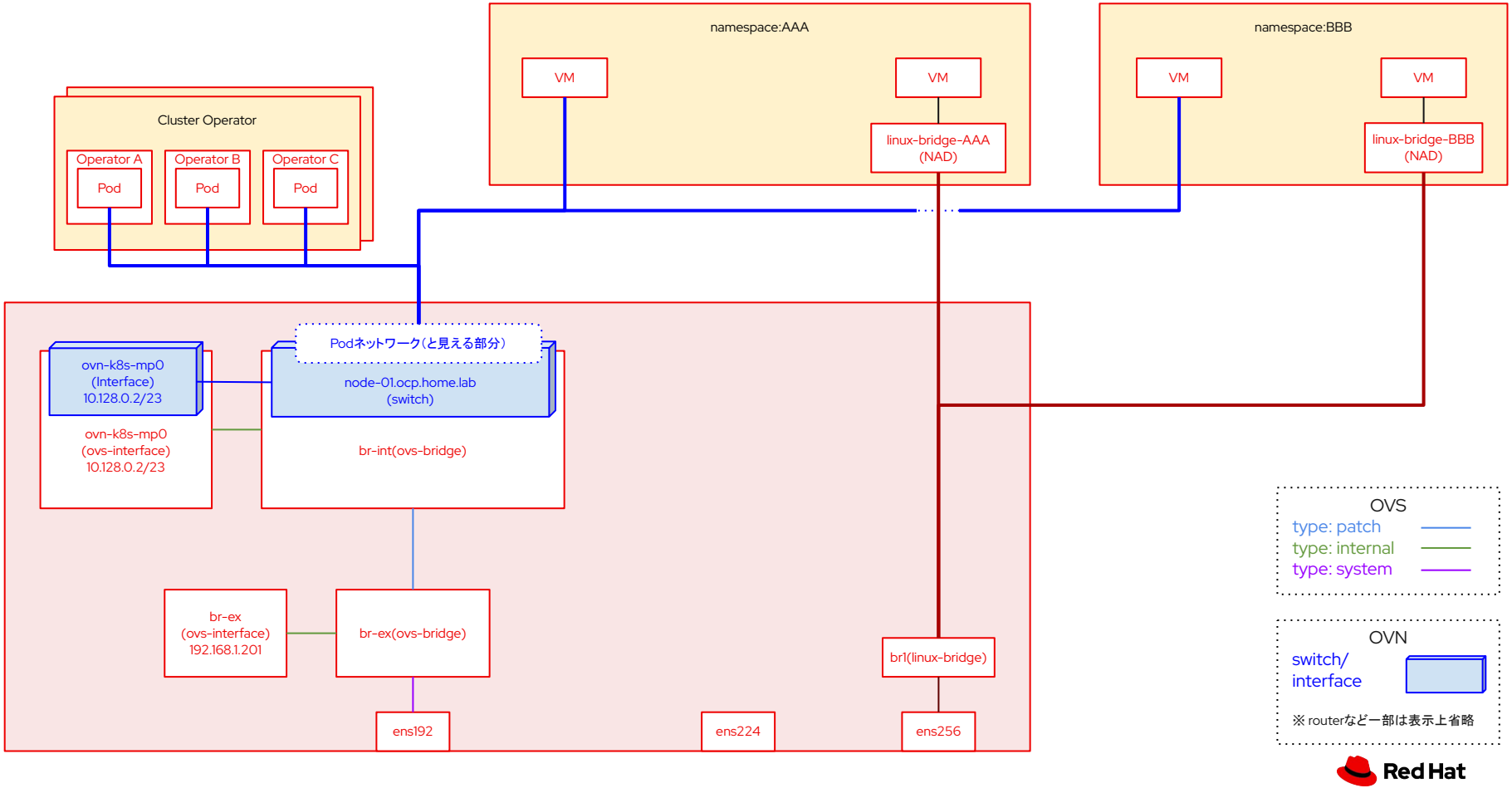
アジェンダ

- Linuxブリッジの構成
- OVN-Kubernetes ローカルネットの構成

Red Hat

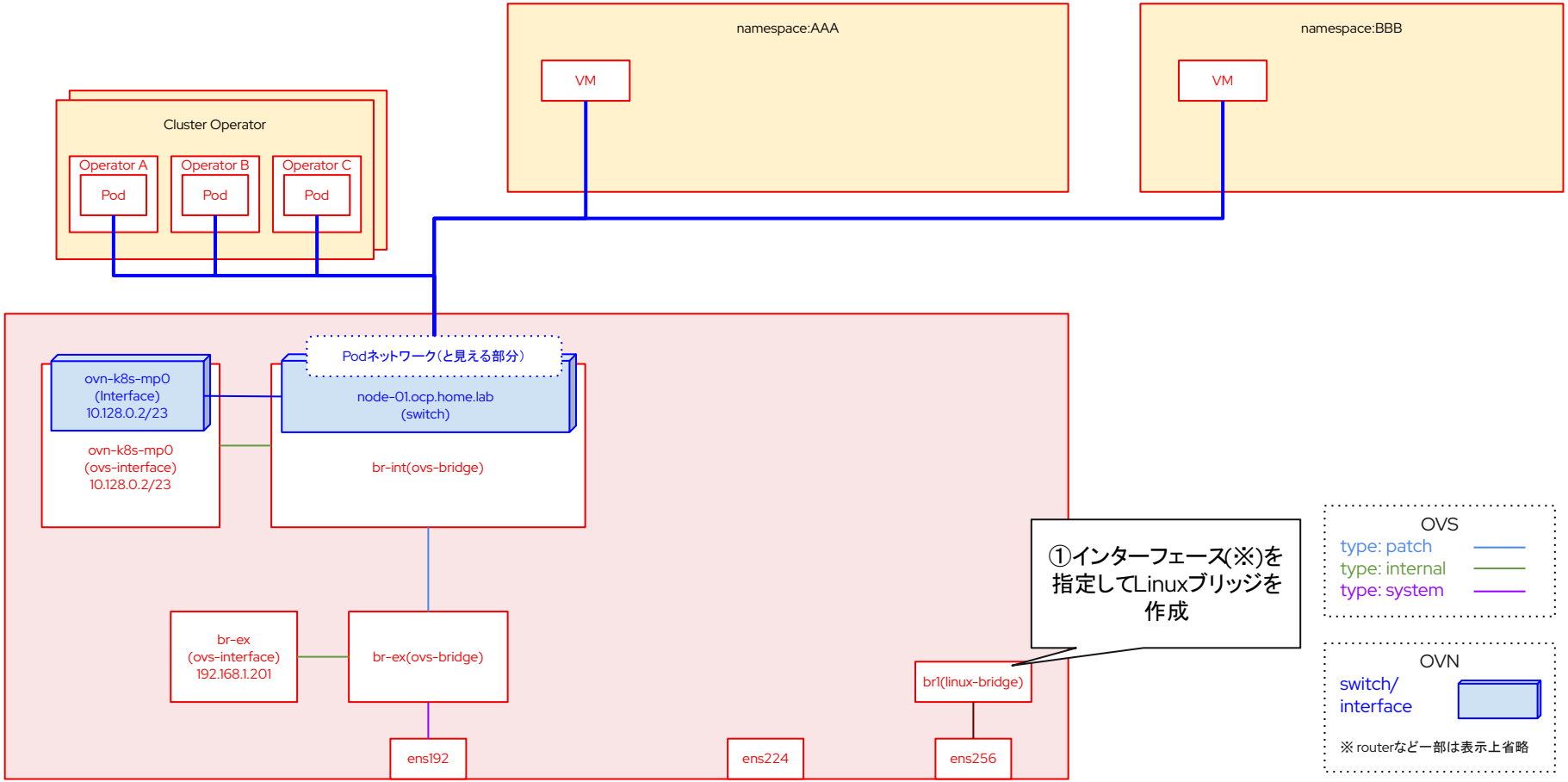
Linuxブリッジの構成

Linuxブリッジの構成完成図



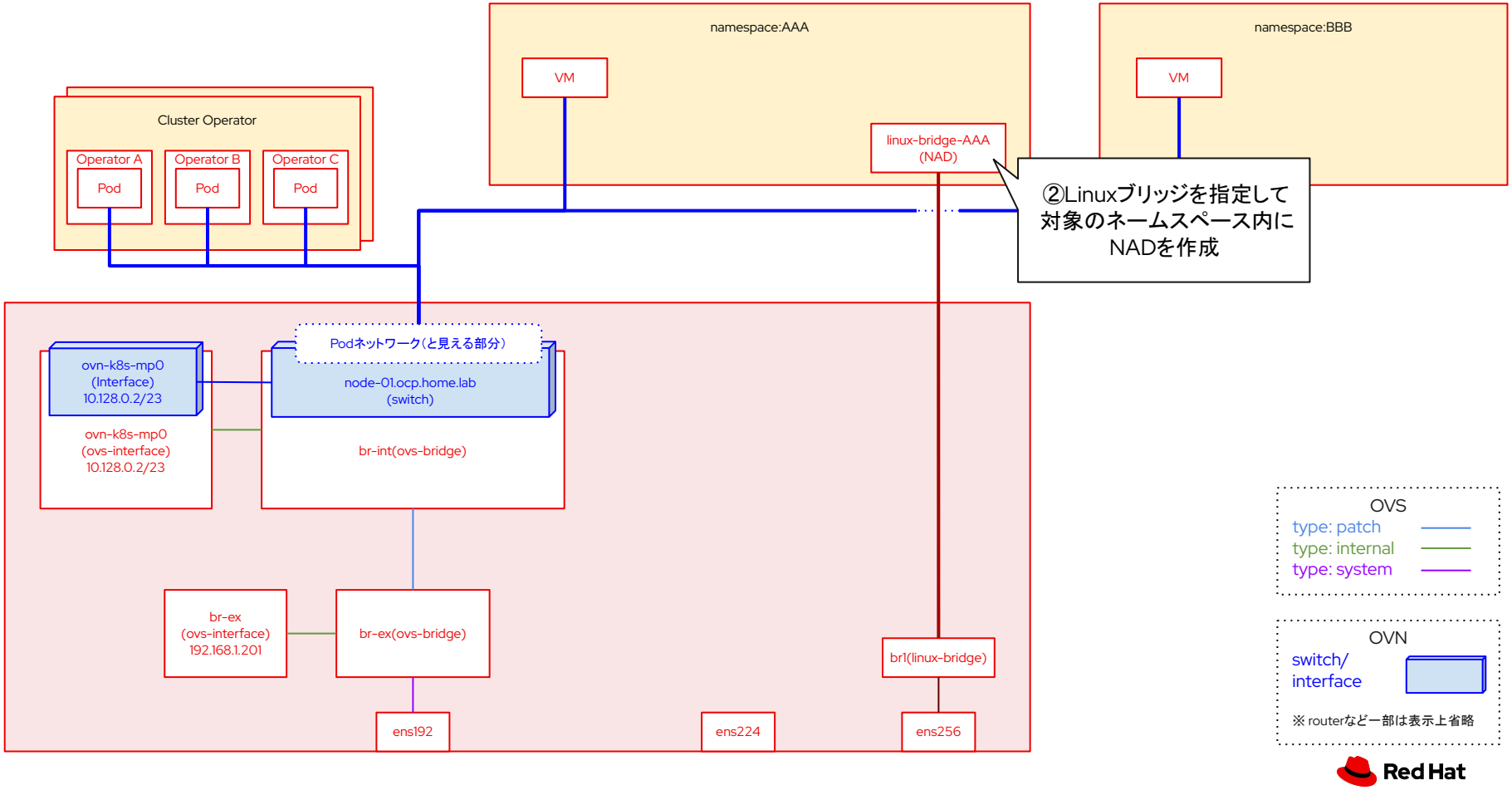
Red Hat

①Linuxブリッジを作成する



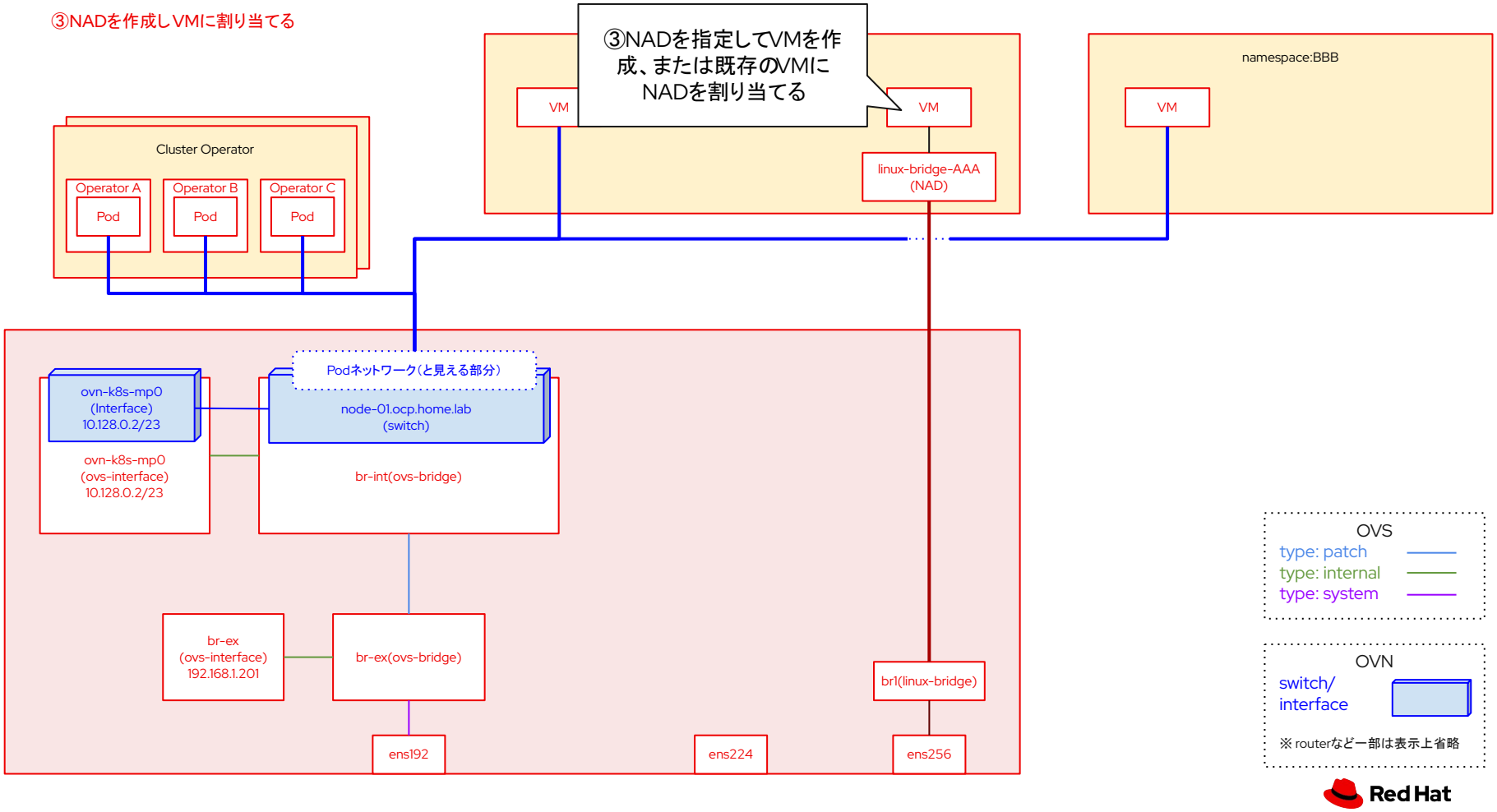
※物理ネットワークインターフェース、ボンディング、VLANデバイス

②Linuxブリッジを指定して NADを作成



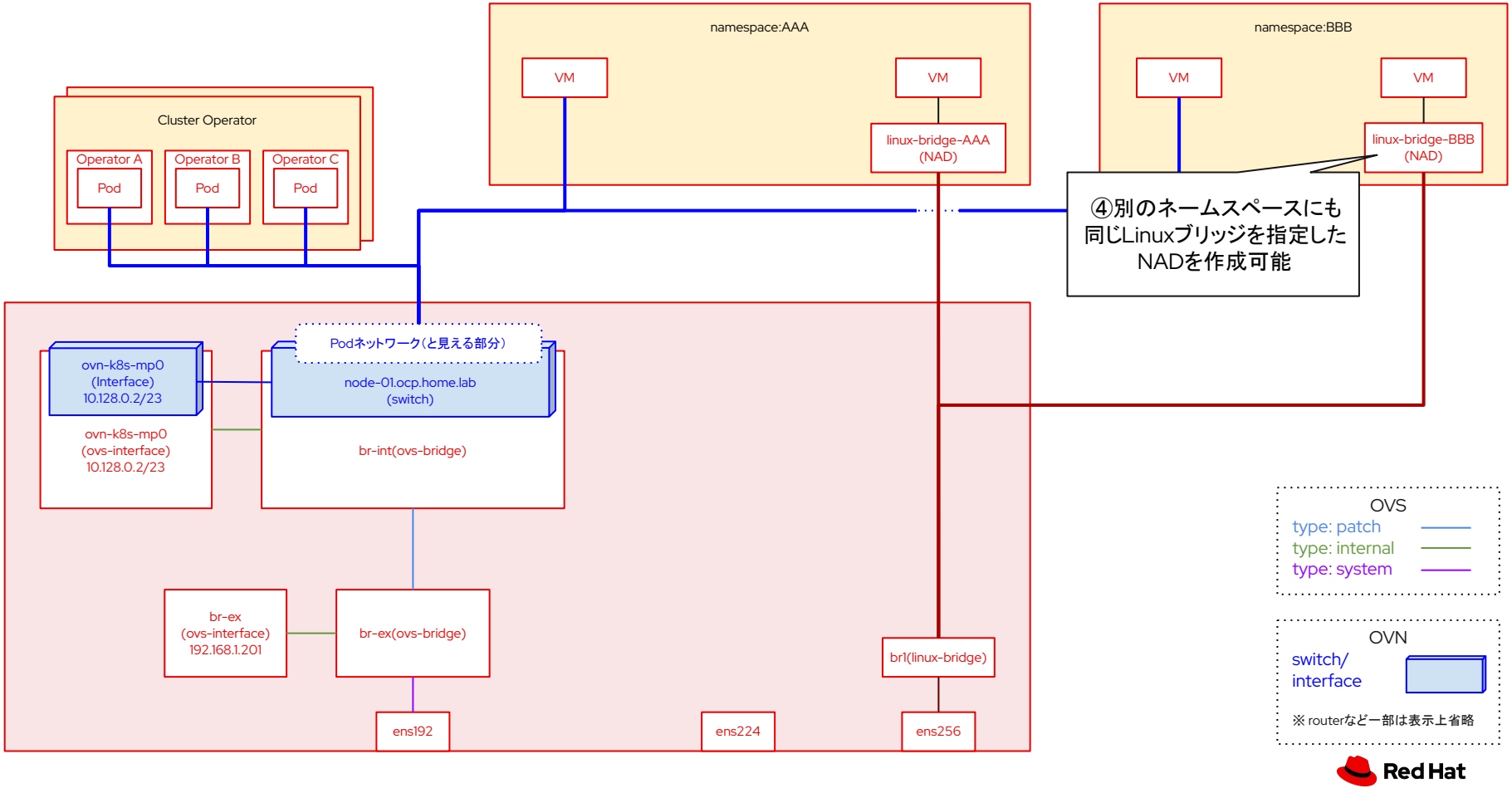
Red Hat

③NADを作成しVMに割り当てる



Red Hat

④別のネームスペースにも同様に NADを作成可能



Red Hat

①Linuxブリッジを作成する

```
$ cat > linux-br1-ens256.yaml <<EOF
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: linux-br1-ens256
spec:
  desiredState:
    interfaces:
      - name: br1
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        bridge:
          options:
            stp:
              enabled: false
          port:
            - name: ens256
EOF
$ oc apply -f linux-br1-ens256.yaml
nodenetworkconfigurationpolicy.nmstate.io/linux-br1-ens256 created
$ oc get nncp linux-br1-ens256
NAME                STATUS      REASON
linux-br1-ens256    Available   SuccessfullyConfigured
```

LinuxブリッジNNCPの内容（e.g. linux-br1-ens256.yaml）

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: linux-br1-ens256
spec:
  desiredState:
    interfaces:
      - name: br1
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        bridge:
          options:
            stp:
              enabled: false
          port:
            - name: ens256
```

- NNCP名：linux-br1-ens256
- ブリッジ名：br1
- ブリッジ先インターフェース：ens256
- IPv4アドレス：無効
- STP：無効

※特定のノードのみに適用したい場合は**spec.nodeSelector**を使用する

Red Hat

②Linuxブリッジを指定してNADを作成

```
$ cat > nad-linux-bridge-br1.yaml <<EOF
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations: {}
  name: nad-linux-bridge-br1
  namespace: vlan-test
spec:
  config: |-
    {
        "cniVersion": "0.3.1",
        "name": "nad-linux-bridge-br1",
        "type": "bridge",
        "bridge": "br1",
        "ipam": {},
        "macspoofchk": true,
        "preserveDefaultVlan": false
    }
EOF
$ oc apply -f nad-linux-bridge-br1.yaml
networkattachmentdefinition.k8s.cni.cncf.io/nad-linux-bridge-br1 created
$ oc get net-attach-def nad-linux-bridge-br1
NAME                   AGE
nad-linux-bridge-br1   2m44s
```

NADの内容（e.g. nad-linux-bridge-br1.yaml）

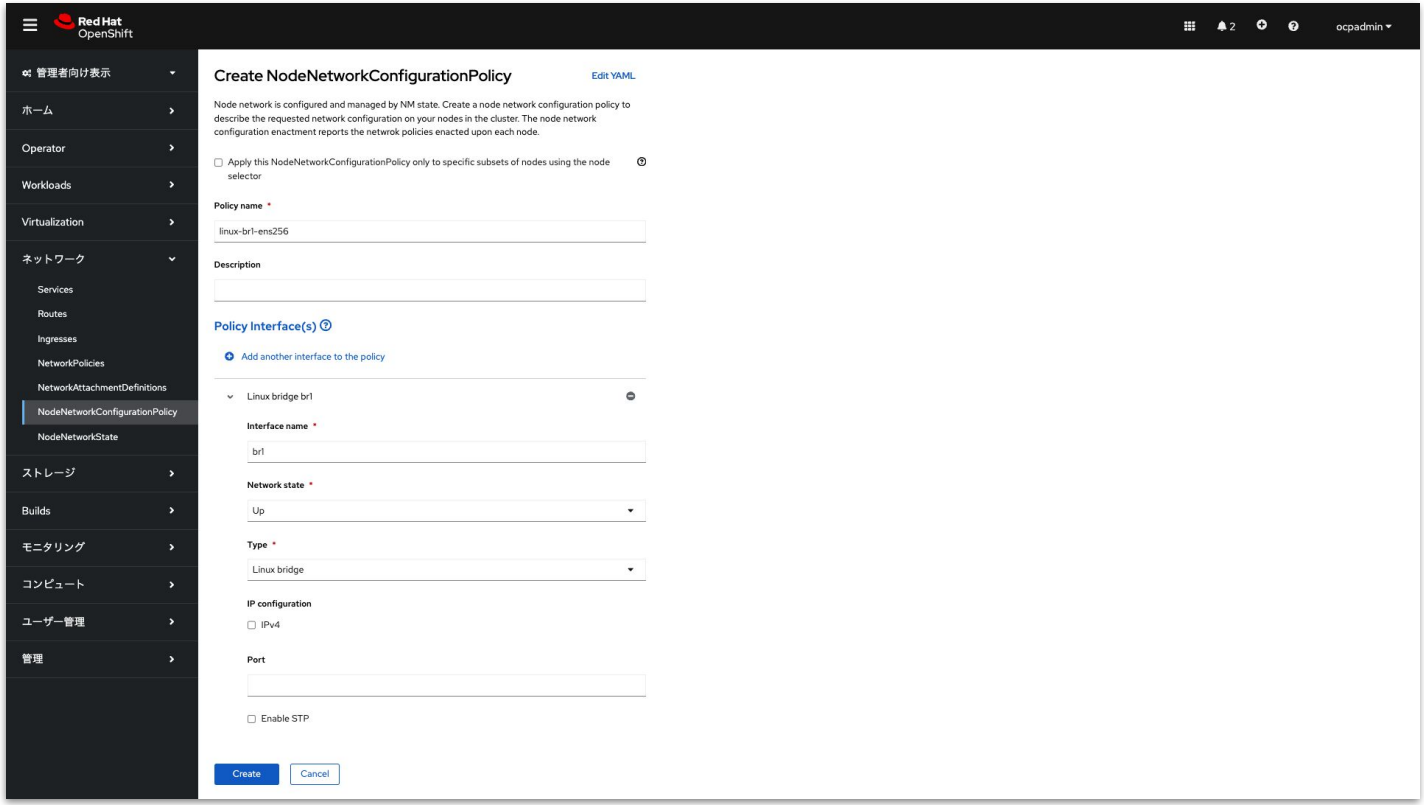
```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations: {}
  name: nad-linux-bridge-br1
  namespace: vlan-test
spec:
  config: |-
    {
        "cniVersion": "0.3.1",
        "name": "nad-linux-bridge-br1",
        "type": "bridge",
        "bridge": "br1",
        "ipam": {},
        "macspoofchk": true,
        "preserveDefaultVlan": false
    }
```

- NAD名：nad-linux-bridge-br1
- ネームスペース：vlan-test
- タイプ：bridge
- ブリッジ先インターフェース：br1
- MAC スプーフィングチェック：true（デフォルト値）

Red Hat

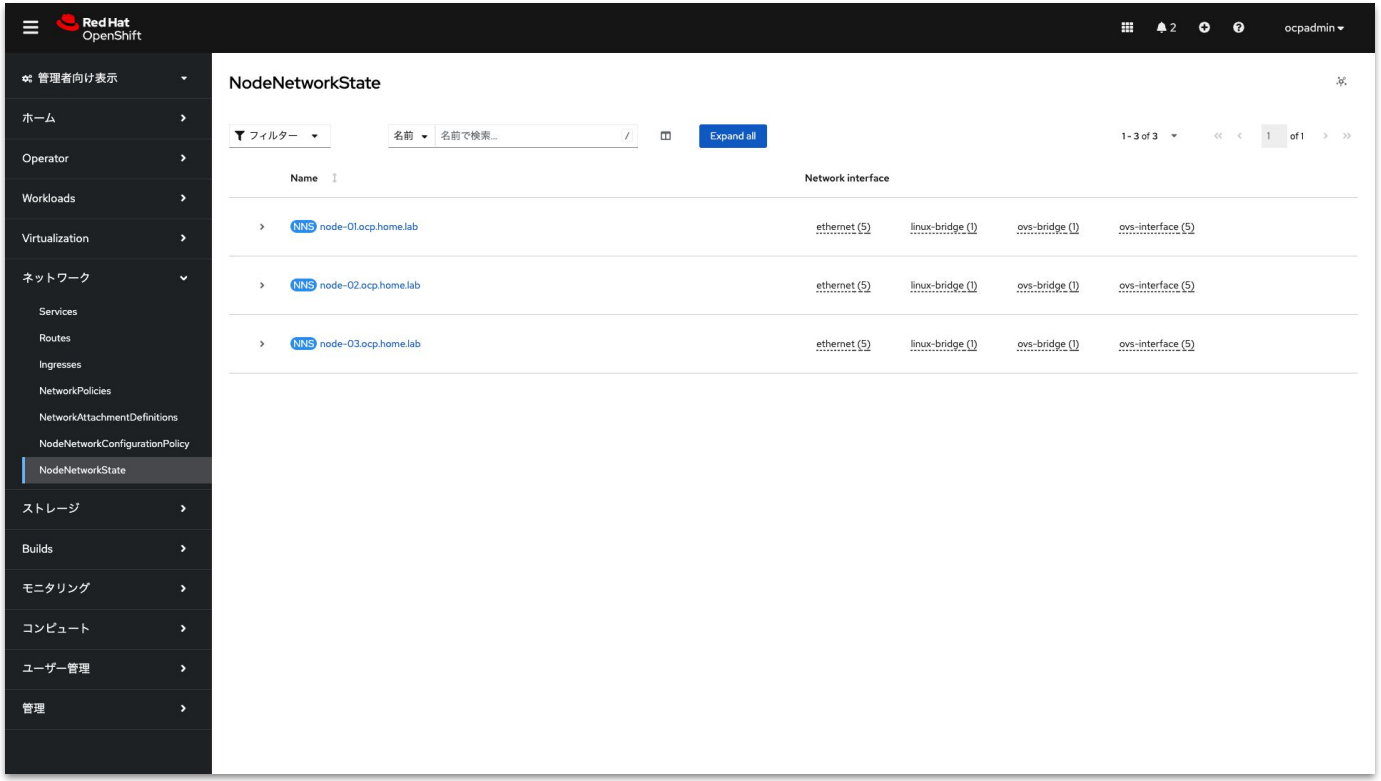
GUIでLinuxブリッジを構成する

①Linuxブリッジを作成する

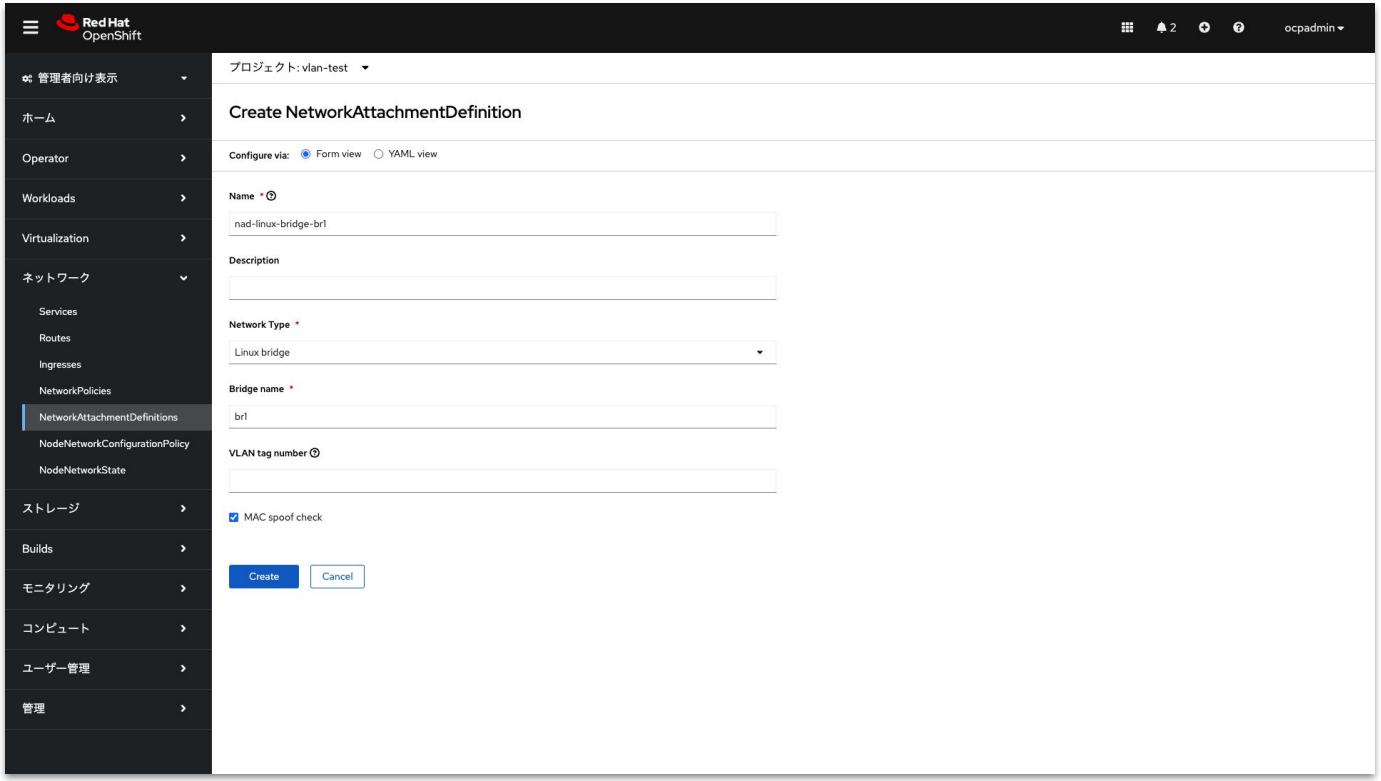


Red Hat

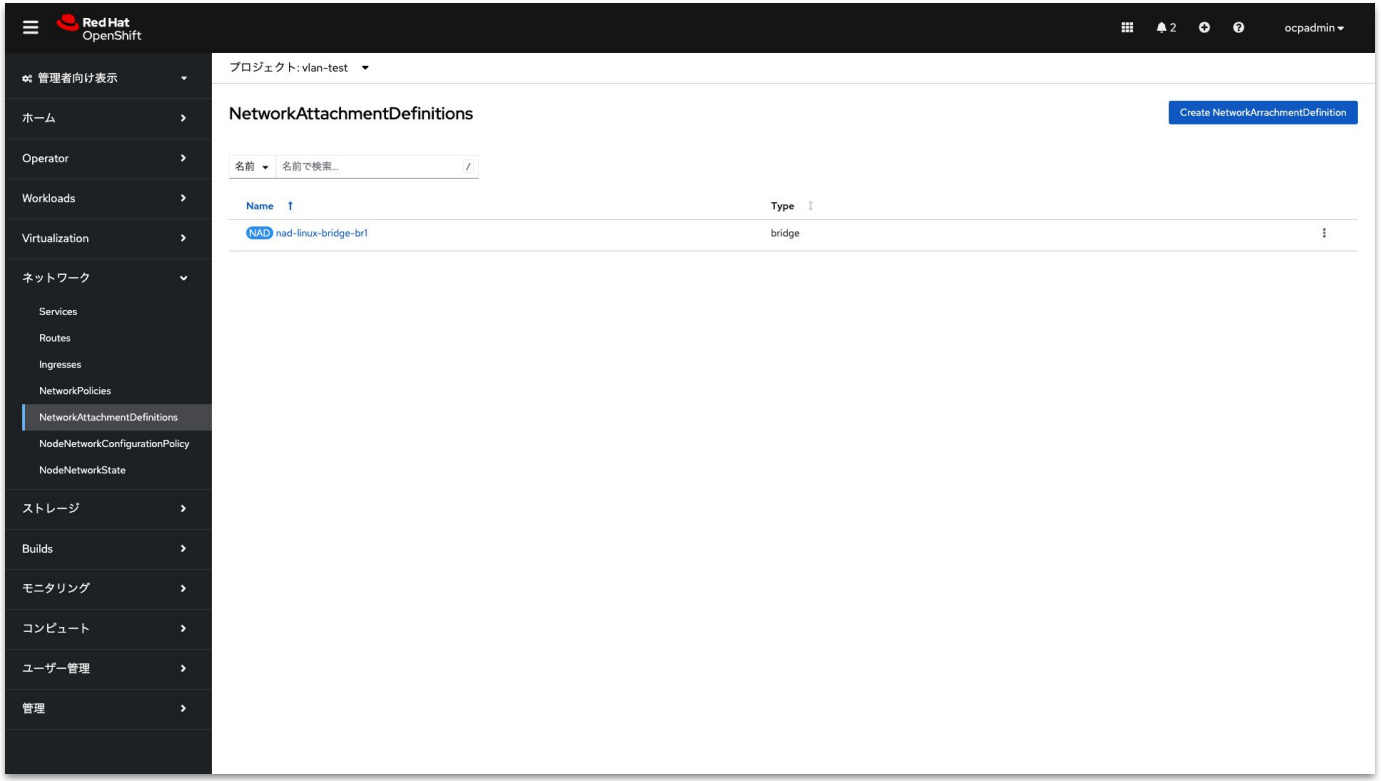
①Linuxブリッジを作成する（作成後の確認）



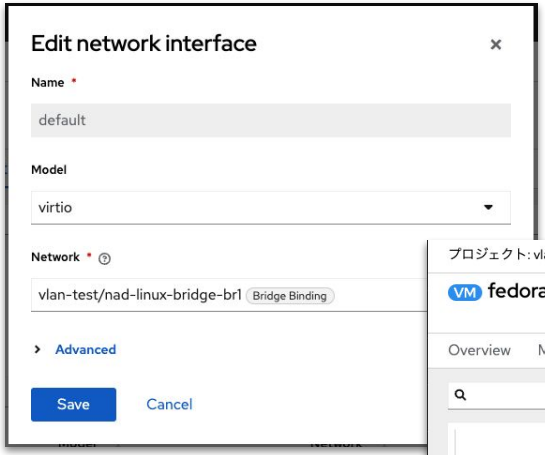
②Linuxブリッジを指定してNADを作成

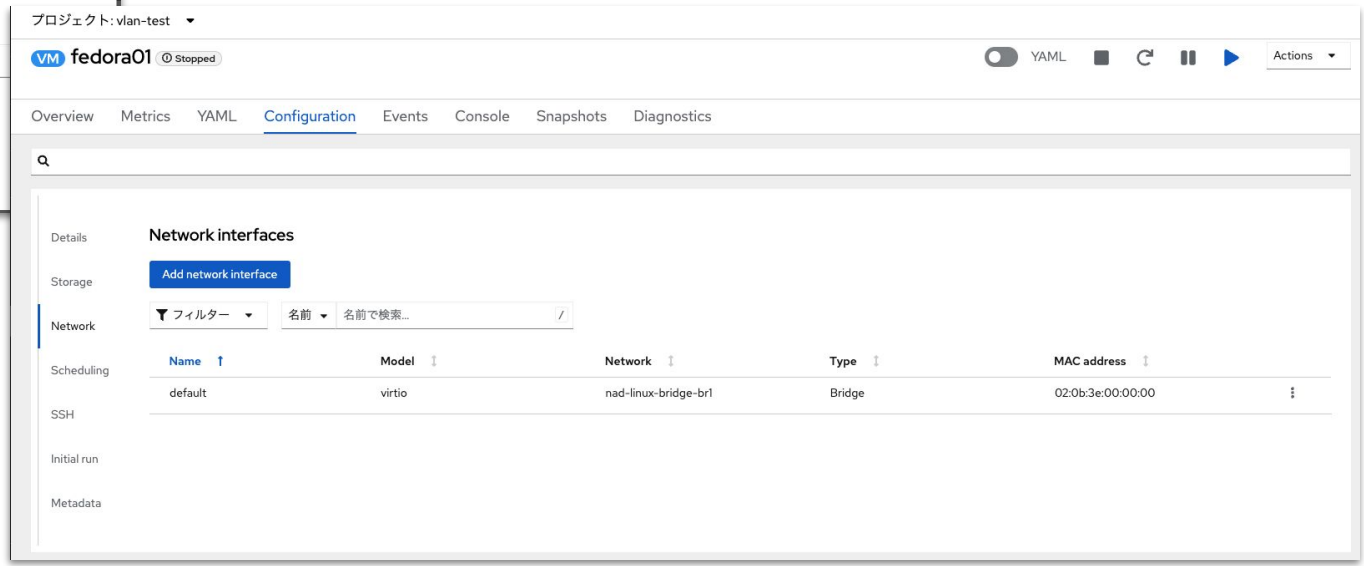


②Linuxブリッジを指定してNADを作成（作成後の確認）



③NADを指定してVMを作成、または既存のVMにNADを割り当てる



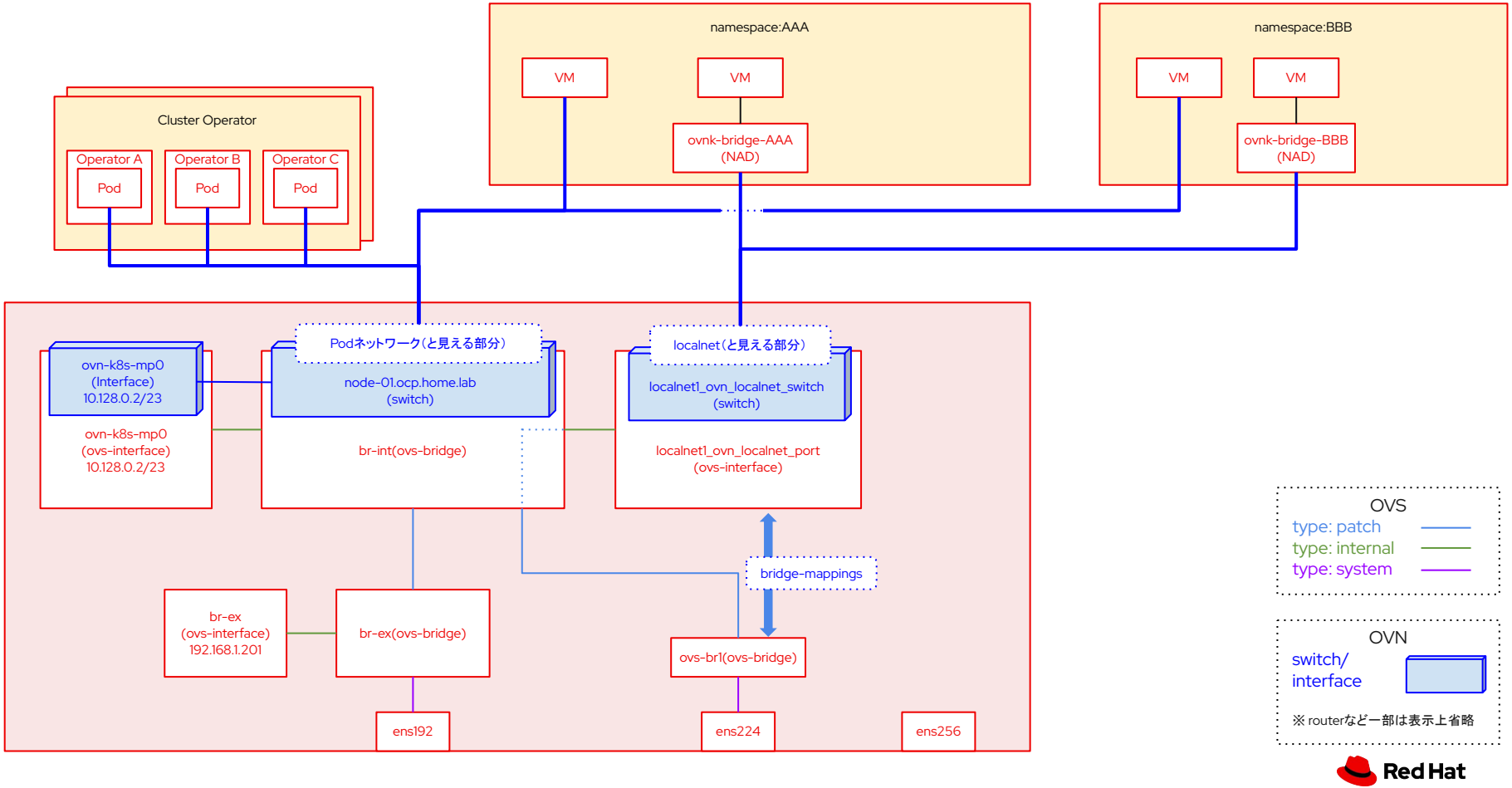


Red Hat

OVN-Kubernetes ローカルネットの構成

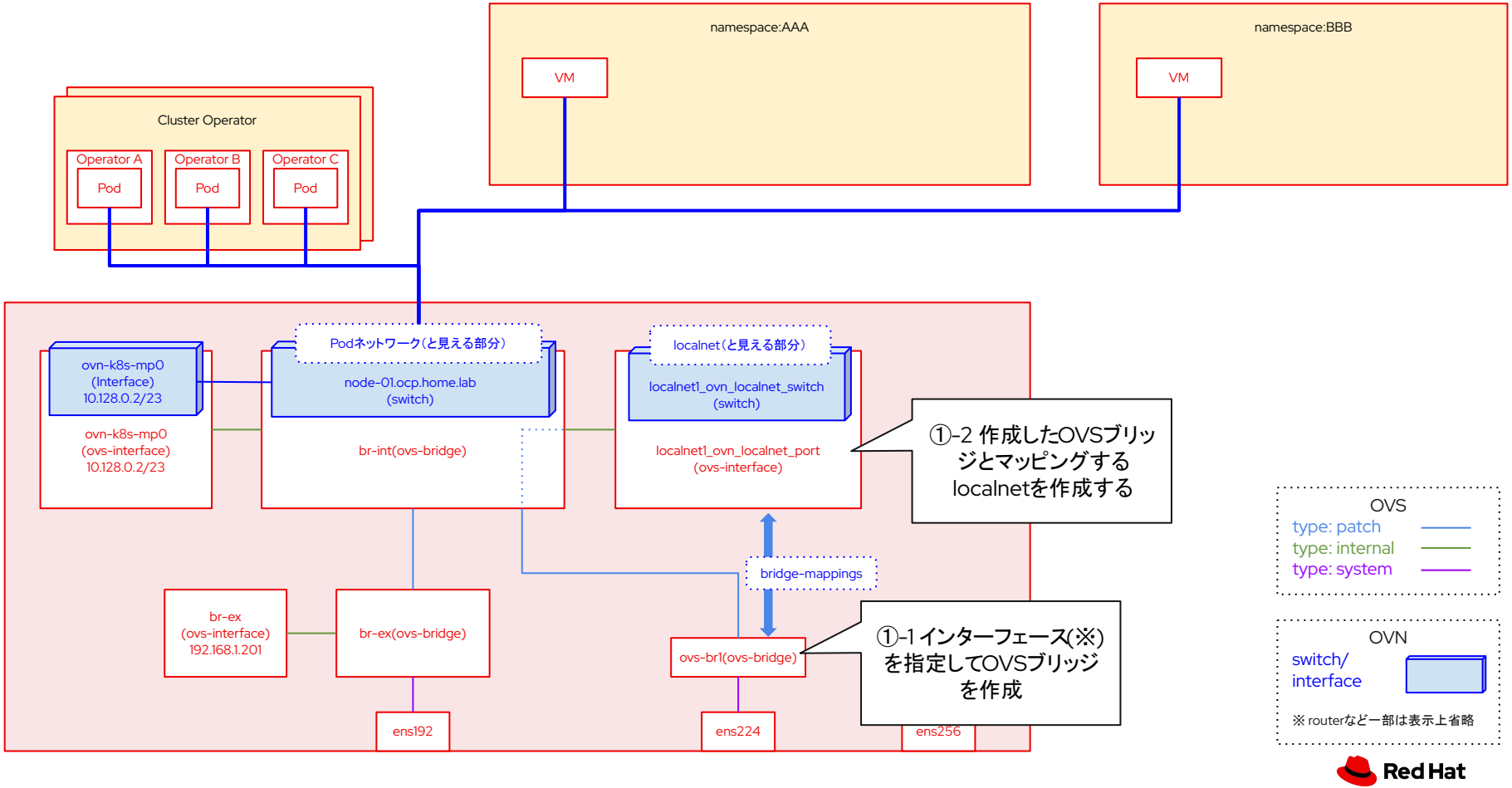
Red Hat

OVN-K localnetの構成完成図



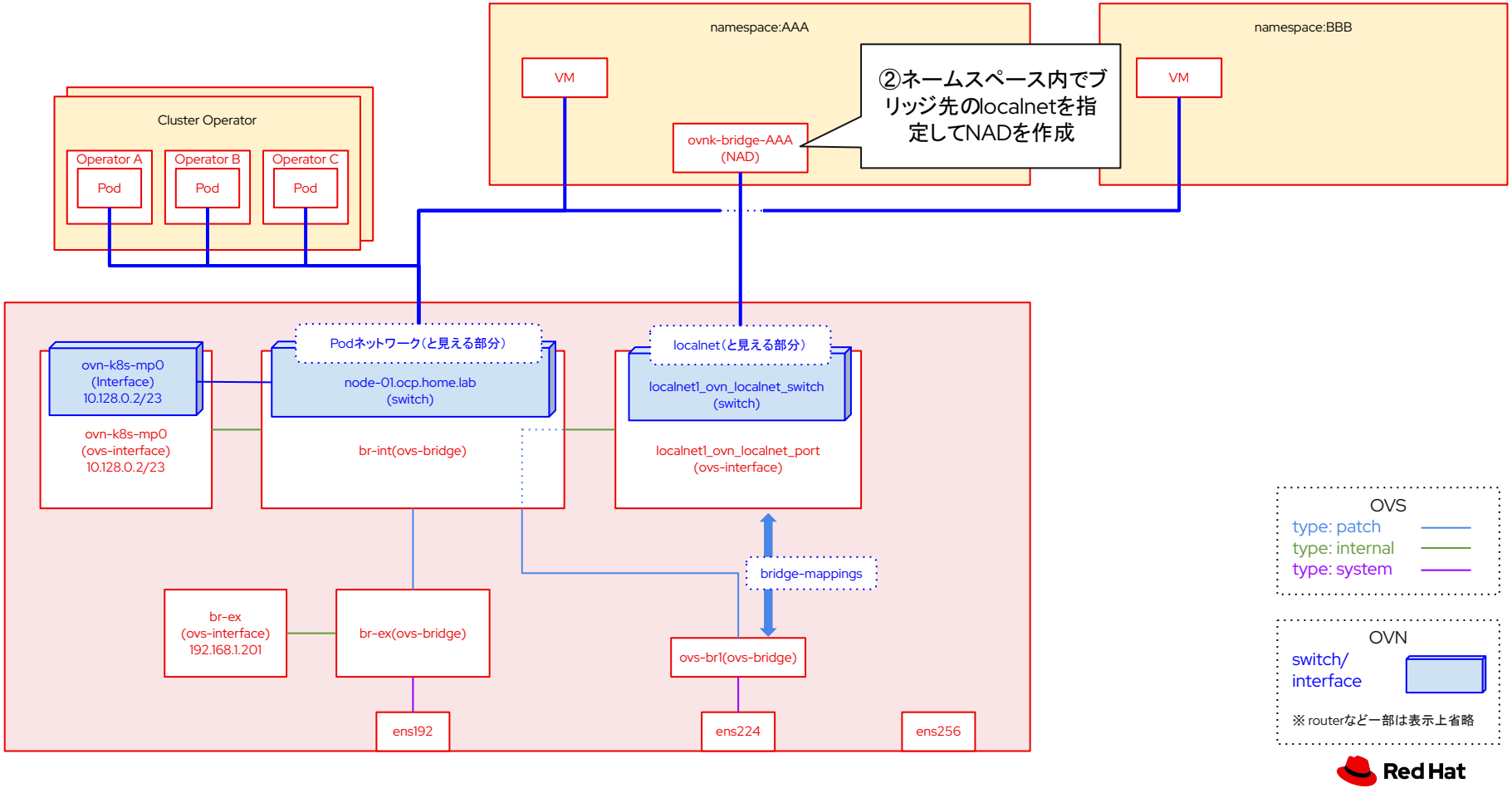
Red Hat

①OVSブリッジとlocalnetを作成し、マッピングする

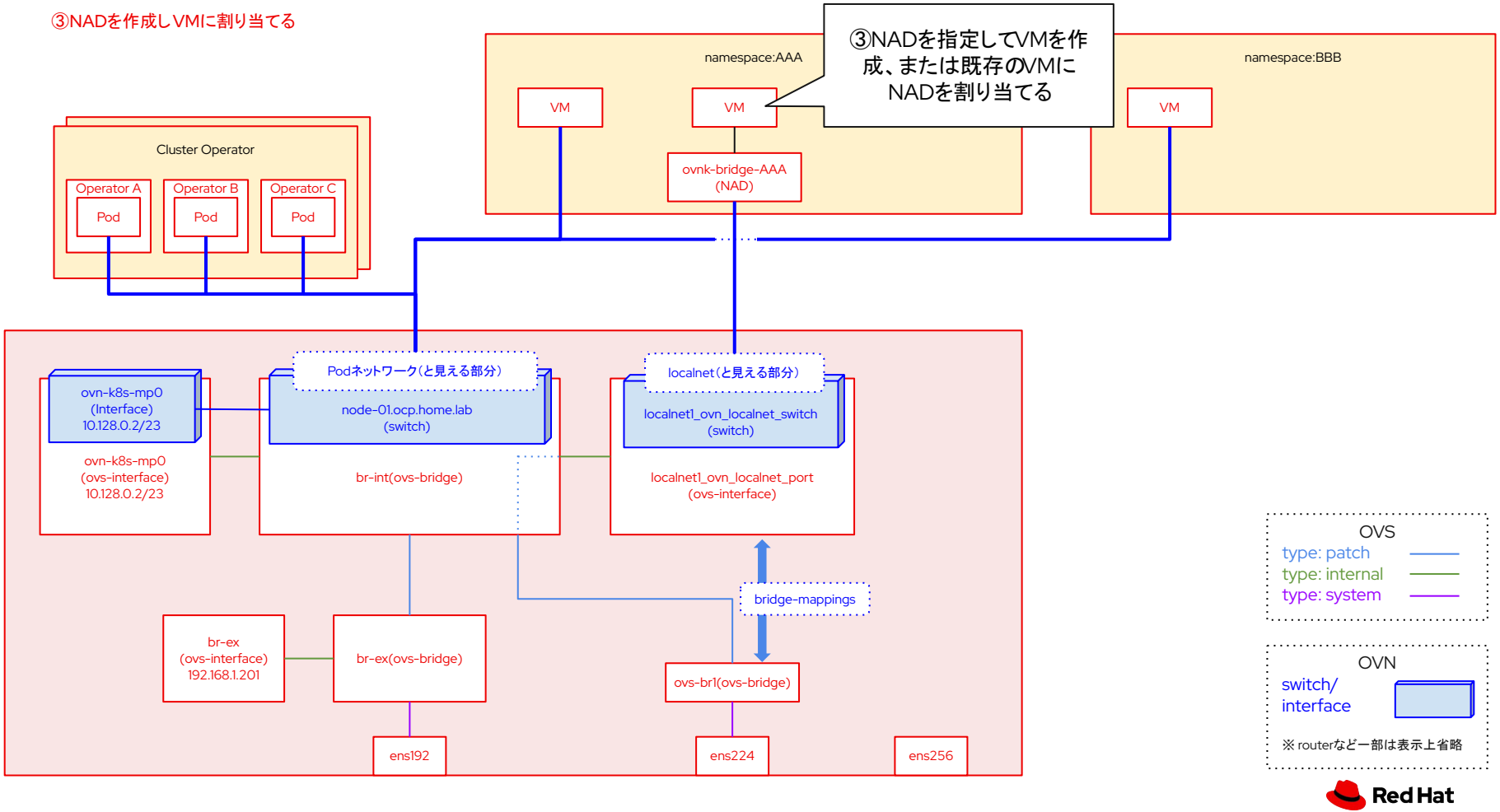


Red Hat

②localnetを指定してNADを作成する

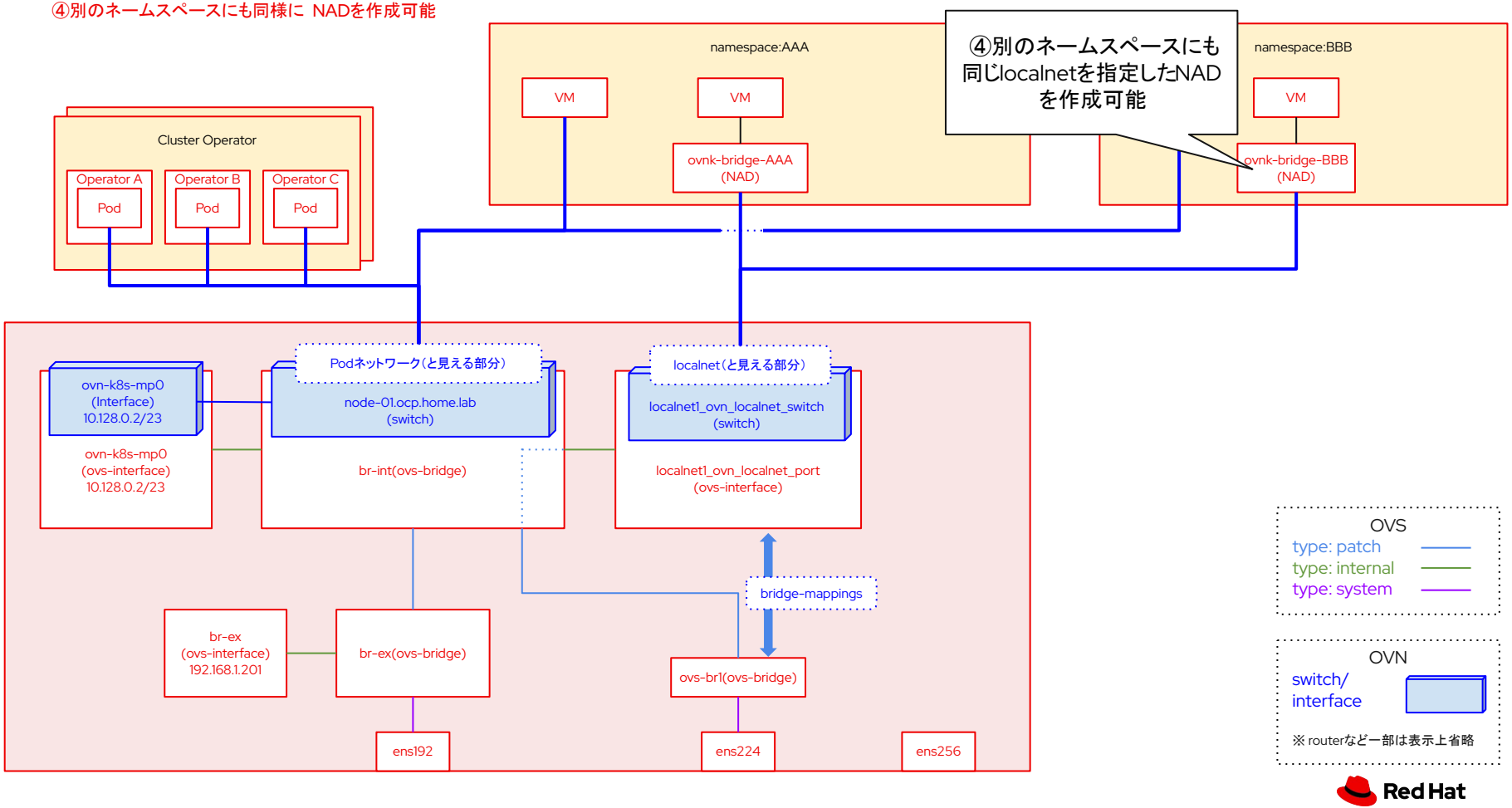


③NADを作成しVMに割り当てる



Red Hat

④別のネームスペースにも同様に NADを作成可能



①OVSブリッジとlocalnetを作成し、マッピングする

```
$ cat > ovs-br1-ens224.yaml <<EOF
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ovs-br1-ens224
spec:
  desiredState:
    interfaces:
    - name: ovs-br1
      type: ovs-bridge
      state: up
      bridge:
        allow-extra-patch-ports: true
        options:
          stp: false
        port:
        - name: ens224
    ovn:
      bridge-mappings:
      - localnet: localnet1
        bridge: ovs-br1
        state: present
EOF
$ oc apply -f ovs-br1-ens224.yaml
nodenetworkconfigurationpolicy.nmstate.io/ovs-br1-ens224 created
$ oc get nncp ovs-br1-ens224
NAME             STATUS      REASON
ovs-br1-ens224   Available   SuccessfullyConfigured
```

LinuxブリッジNNCPの内容（e.g. ovs-br1-ens224.yaml）

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ovs-br1-ens224
spec:
  desiredState:
    interfaces:
    - name: ovs-br1
      type: ovs-bridge
      state: up
      bridge:
        allow-extra-patch-ports: true
        options:
          stp: false
        port:
        - name: ens224
    ovn:
      bridge-mappings:
      - localnet: localnet1
        bridge: ovs-br1
        state: present
```

- NNCP名：ovs-br1-ens224
- ブリッジ名：ovs-br1
- ブリッジ先インターフェース：ens224
- IPv4アドレス：無効
- STP：無効
- ブリッジマッピング：
  - ローカルネット名：localnet1
  - ブリッジ名：ovs-br1

※特定のノードのみに適用したい場合はspec.nodeSelectorを使用する

Red Hat

②localnetを指定してNADを作成する

```
$ cat > nad-ovnk-bridge-ovs-br1.yaml <<EOF
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations: {}
  name: nad-ovnk-bridge-ovs-br1
  namespace: vlan-test
spec:
  config: |-
    {
        "cniVersion": "0.4.0",
        "name": "localnet1",
        "type": "ovn-k8s-cni-overlay",
        "netAttachDefName": "vlan-test/nad-ovnk-bridge-ovs-br1",
        "topology": "localnet"
    }
EOF
$ oc apply -f nad-ovnk-bridge-ovs-br1.yaml
networkattachmentdefinition.k8s.cni.cncf.io/nad-ovnk-bridge-ovs-br1 created
$ oc get net-attach-def nad-ovnk-bridge-ovs-br1
NAME                      AGE
nad-ovnk-bridge-ovs-br1   17s
```

NADの内容（e.g. nad-ovnk-bridge-ovs-br1.yaml）

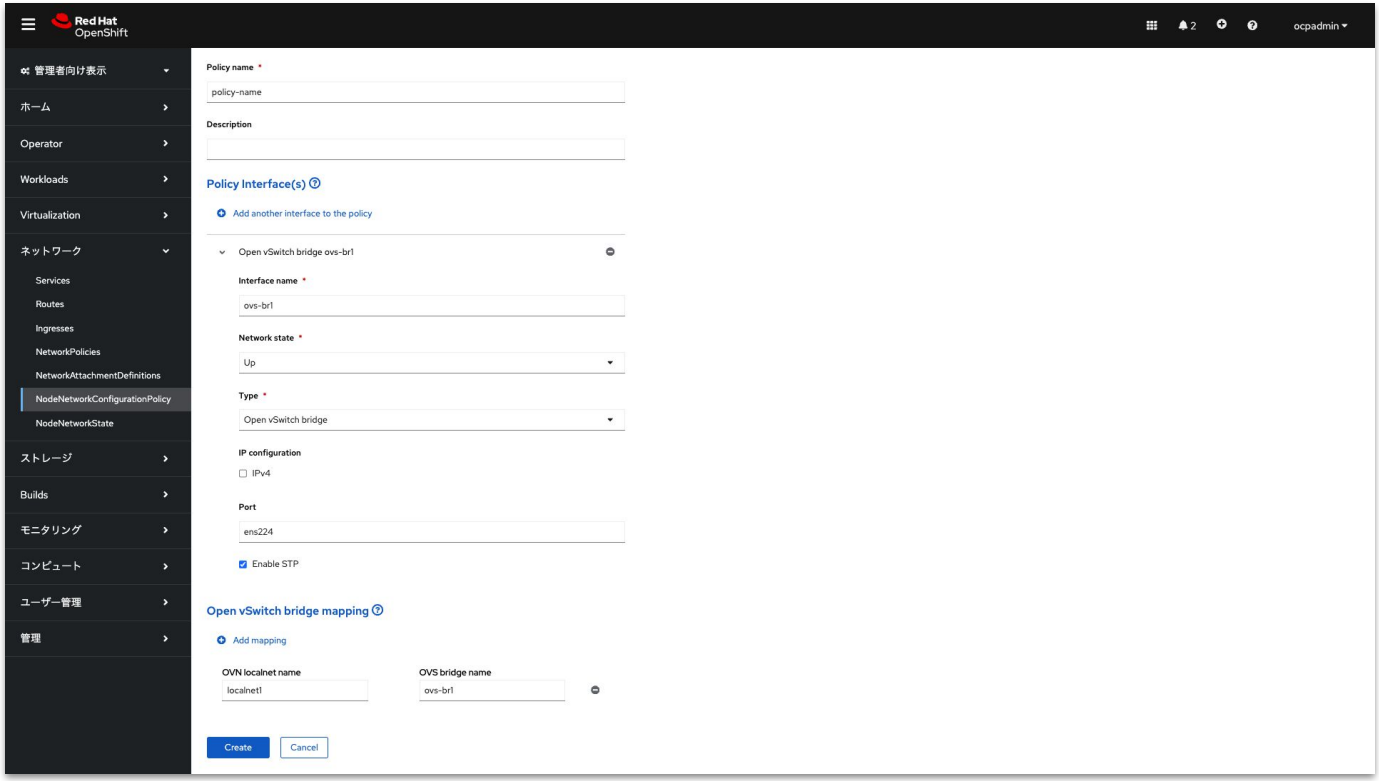
```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations: {}
  name: nad-ovnk-bridge-ovs-br1
  namespace: vlan-test
spec:
  config: |-
    {
        "cniVersion": "0.4.0",
        "name": "localnet1",
        "type": "ovn-k8s-cni-overlay",
        "netAttachDefName": "vlan-test/nad-ovnk-bridge-ovs-br1",
        "topology": "localnet"
    }
```

- NAD名：nad-ovnk-bridge-ovs-br1
- ネームスペース：vlan-test
- ネットワーク：localnet1
- タイプ：ovn-k8s-cni-overlay
- トポロジー：localnet

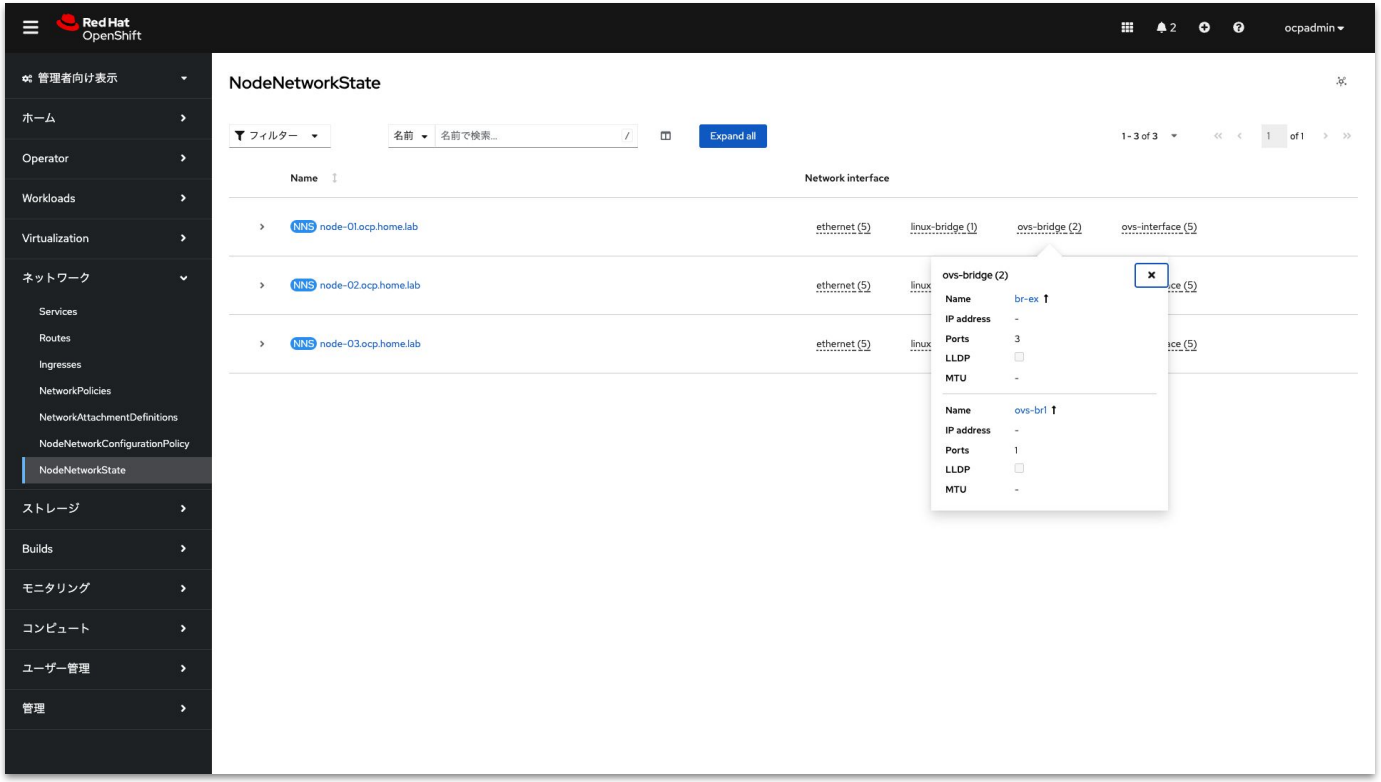
Red Hat

GUIでOVN-K localnetを構成する

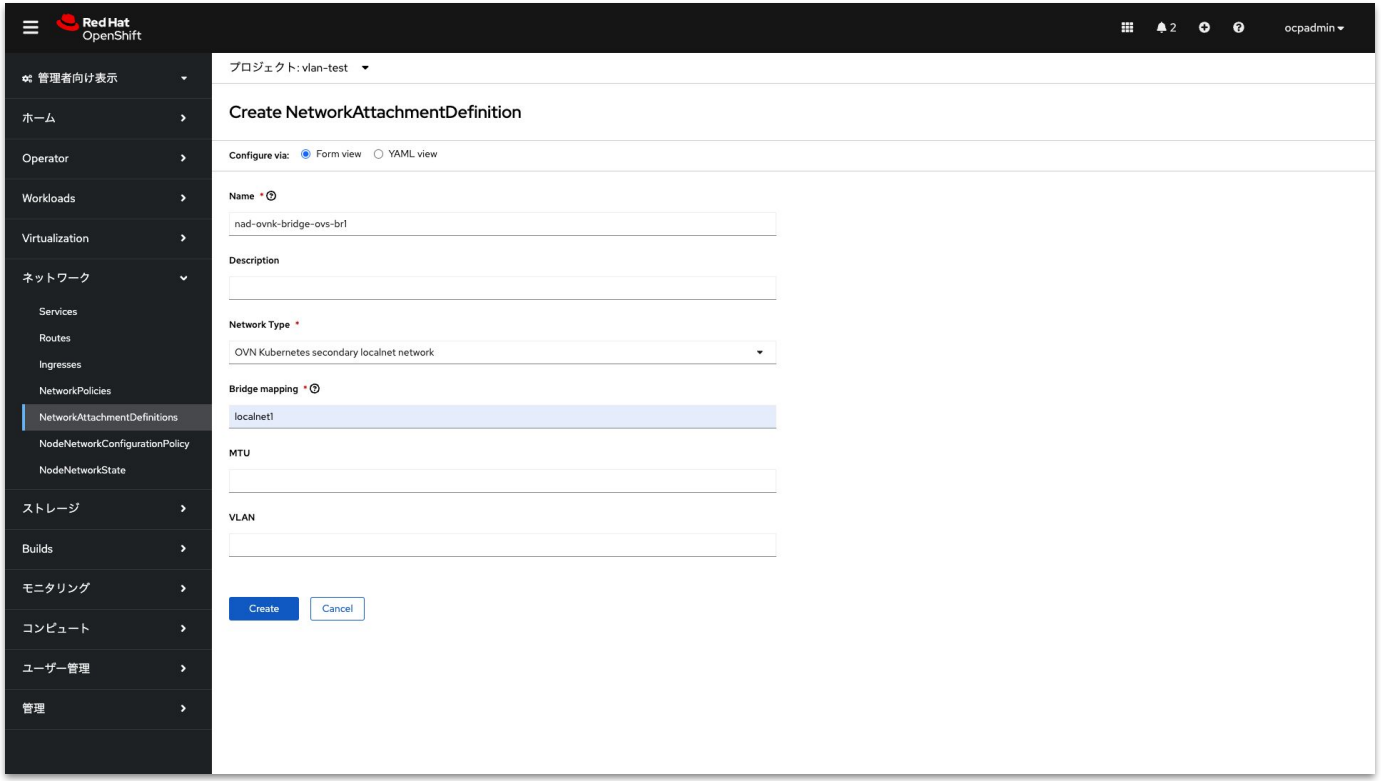
①OVSブリッジとlocalnetを作成し、マッピングする



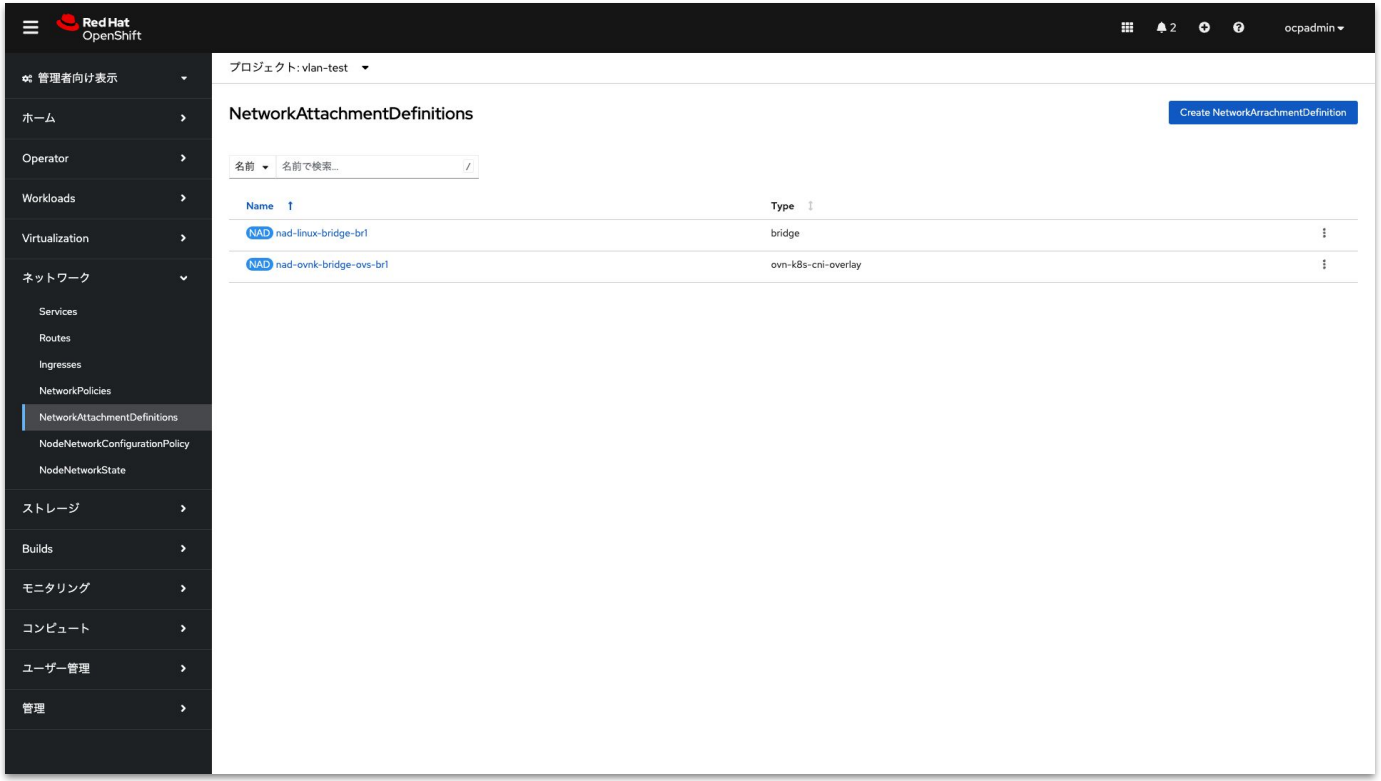
①OVSブリッジとlocalnetを作成し、マッピングする（作成後の確認）



②localnetを指定してNADを作成する



②localnetを指定してNADを作成する（作成後の確認）



③NADを指定してVMを作成、または既存のVMにNADを割り当てる