Déploiement de DNS VNF avec le réseau SRIOV sur Openstack CVIM - Exemple de configuration pour Prime Network Registrar (DNS)

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés

Informations générales

Configuration

- 1. Configuration matérielle requise
- 2.Identification des cartes réseau Intel
  - Étape 1. Utilisation de la commande lspci
  - Étape 2. Vérification de XL710
  - Étape 3. Vérification de E810CQDA2
  - Étape 4. Confirmation de la prise en charge des pilotes
- 3. Configuration du BIOS/UEFI
- 4. Configuration d'OpenStack
- 5. Image VNF Cisco Prime Network Registrar (CPNR)
- 6. Accès administratif

Présentation de l'architecture

- Diagramme de connectivité d'interface réseau VNF
- Organigramme
- Exemples de configurations
- Points clés

Déploiement de CPNR VNF avec ports SR-IOV et interface de liaison de sauvegarde active sur OpenStack

Principales caractéristiques du déploiement

Pourquoi le mode Cross-NUMA est requis

- 1. Réseau compatible NUMA dans OpenStack
- 2. Pourquoi le mode Cross-NUMA est nécessaire

Limitation de la taille des connecteurs pour les ports OVS

- Qu'est-ce que Conntrack ?
- Impact de Conntrack sur les ports OVS
- Comment réduire les limites de connexion

Comment SR-IOV résout les problèmes de suivi

- 1. Élimine la dépendance Conntrack
- 2. Évolutivité supérieure

- 3. Latence réduite

Pourquoi le mode de sauvegarde active est choisi pour les ports SR-IOV sur la machine virtuelle CPNR

- 1. Redondance sans complexité
- 2. Aucun LAG (Link Aggregation Group) requis
- 3. Basculement transparent
- 4. Indépendance matérielle
- 5. Optimisé pour SR-IOV

Qu'est-ce qu'une interface Linux Bond ?

Comment fonctionne le mode de sauvegarde active ?

- Principales caractéristiques du mode de sauvegarde active

Flux du trafic en mode de sauvegarde active

- Fonctionnement normal
- Scénario de basculement
- Scénario de restauration

Scénario: Liaison de secours active avec ports SR-IOV

Étape 1. Mise en réseau OpenStack

- Étape 1.1. Création De Réseaux Openvswitch
- Étape 1.2. Création de sous-réseaux pour les réseaux Openvswitch
- Étape 1.3. Création de réseaux SR-IOV

Étape 2. Saveurs OpenStack

- Étape 2.1. Création d'une saveur Cross-NUMA
- Étape 2.2. Configuration des propriétés NUMA

Étape 3 : configuration de la liaison en mode de sauvegarde active

- Étape 3.1. Configuration de l'interface de liaison
- Étape 3.2. Configuration des interfaces esclaves
- Étape 3.3. Application de la configuration

Vérifier

- 1. Vérifier l'état VNF
- 2. Vérification de la connectivité réseau
- 3. Vérifier le positionnement NUMA

Meilleures pratiques

Dépannage

- 1. Vérification de la configuration SR-IOV
- 2. Vérifier le positionnement NUMA
- 3. Problèmes d'interface de liaison
- 4. Problèmes de connectivité réseau

Conclusion

Introduction

Ce document décrit le déploiement pas à pas de CPNR sur OpenStack Cisco Virtualized

Infrastructure Manager (CVIM) à l'aide de SR-IOV et de la liaison Active-Backup.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connaissance des concepts OpenStack et SR-IOV (Single Root Input/Output Virtualization)
- Connaissances pratiques sur Cisco Virtual Interface Manager (VIM), Cisco Elastic Services Controller et les commandes et réseaux Linux

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes

Informations générales

Dans l'environnement réseau actuel, les fonctions de réseau virtuel (VNF) jouent un rôle essentiel dans la mise en place de services réseau agiles, évolutifs et efficaces. Pour les VNF nécessitant une connectivité réseau hautes performances, SR-IOV est une technologie couramment utilisée. SR-IOV permet aux VNF de contourner le commutateur virtuel de l'hyperviseur et d'accéder directement aux ressources physiques du contrôleur d'interface réseau (NIC), réduisant ainsi la latence et augmentant le débit.

Configuration

Avant de procéder au déploiement, assurez-vous que ces conditions préalables sont remplies.

1. Configuration matérielle requise

- Cartes réseau compatibles SR-IOV :

    - Au moins deux cartes réseau physiques compatibles SR-IOV avec SR-IOV activé dans le BIOS/UEFI (Unified Extensible Firmware Interface).
    - Exemple : sriov0 mappé au noeud 0 NUMA (Non-Uniform Memory Access) et sriov1 mappé au noeud 1 NUMA.

- Hôtes compatibles NUMA :
    - Les noeuds de calcul doivent prendre en charge l'architecture NUMA.
    - La prise en charge NUMA doit être activée dans le BIOS/UEFI des hôtes.

2. Identification des cartes réseau Intel

Les cartes réseau Intel XL710 et E810CQDA2 sont couramment utilisées pour les réseaux SR-IOV hautes performances. Afin de vérifier le modèle de carte réseau sur l'hôte, référez-vous à ces étapes :

Étape 1. Utilisation de la commande lspci

Exécutez cette commande afin de répertorier les périphériques PCI (Peripheral Component Interconnect) associés aux contrôleurs réseau :

```
lspci | grep -i ethernet
```

Exemple de sortie :

```
81:00.0 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 02)
82:00.0 Ethernet controller: Intel Corporation Ethernet Controller E810-C for QSFP (rev 03)
```

Étape 2. Vérification de XL710

Si la carte réseau est Intel XL710, vous pouvez voir le contrôleur Ethernet XL710 dans la sortie.

Étape 3. Vérification de E810CQDA2

Si la carte réseau est Intel E810CQDA2, vous pouvez voir le contrôleur Ethernet E810-C dans la sortie.

Étape 4. Confirmation de la prise en charge des pilotes

Afin de vérifier le pilote de la carte réseau en cours d'utilisation, exécutez :

```
ethtool -i <NIC_INTERFACE_NAME>
```

Exemple de sortie pour XL710 :

```
driver: i40e
version: 2.13.10
```

Exemple de sortie pour E810CQDA2 :

```
driver: ice
version: 1.7.12
```

Assurez-vous que la version du pilote correspond à la matrice de compatibilité de votre distribution OpenStack et Linux.

3. Configuration du BIOS/UEFI

- Activer SR-IOV :

Assurez-vous que SR-IOV est activé dans le BIOS/UEFI des serveurs.

- Activer la technologie de virtualisation pour les E/S dirigées (VT-d)/AMD-Vi :

Intel VT-d ou AMD-Vi doivent être activés pour la fonctionnalité de transfert PCI et SR-IOV.

4. Configuration d'OpenStack

- Services OpenStack de base :

Assurez-vous que les services OpenStack tels que Nova, Neutron, Glance et Keystone sont

installés et configurés.

- Configuration des neutrons :

Neutron doit prendre en charge Openswitch (OVS) pour les réseaux d'orchestration/gestion et SR-IOV pour les réseaux d'applications/services.

- Configuration SR-IOV :

Les noeuds de calcul doivent être configurés pour prendre en charge SR-IOV, avec des fonctions virtuelles (VF) créées sur les cartes réseau.

5. Image VNF Cisco Prime Network Registrar (CPNR)

- Compatibilité des images VNF :

L'image VNF CPNR doit prendre en charge les interfaces SR-IOV et inclure les pilotes nécessaires.

- Télécharger dans Aperçu :

Assurez-vous que l'image CPNR VNF est disponible dans OpenStack Glance.

6. Accès administratif

- CLI OpenStack :

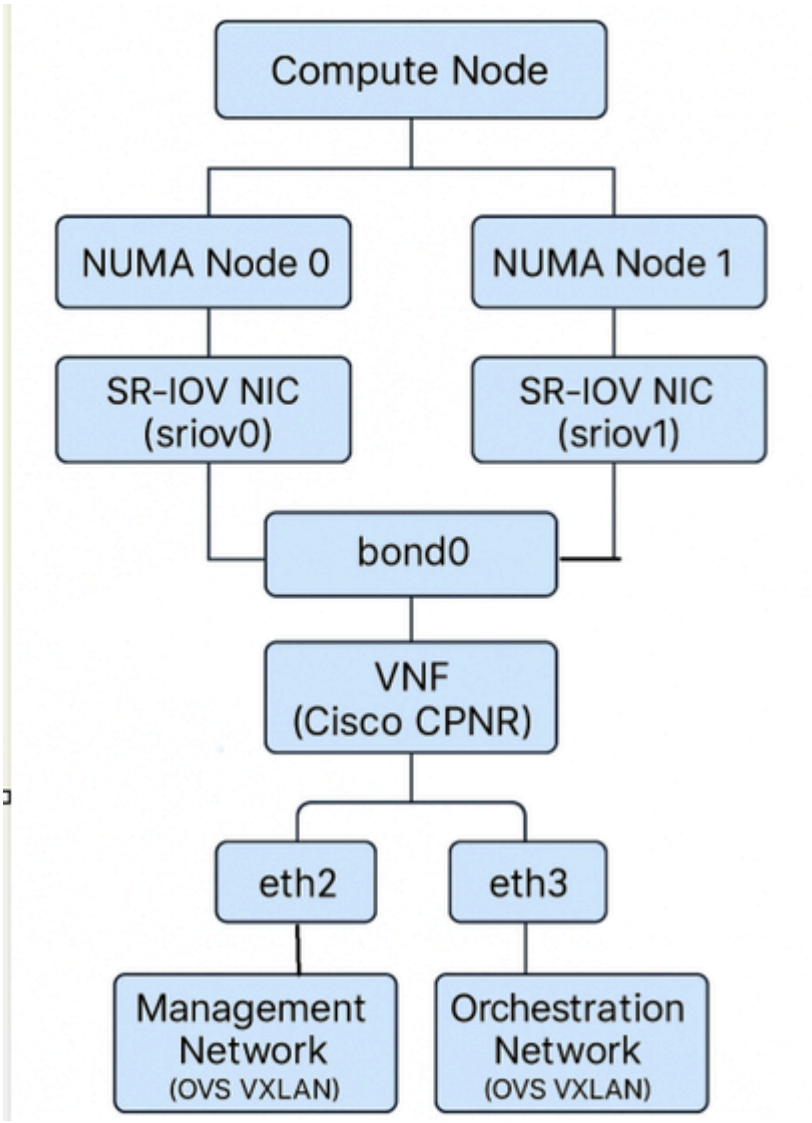
Assurez l'accès à l'interface de ligne de commande OpenStack pour créer des réseaux, des saveurs et lancer le VNF.

- Privilèges racine ou admin :

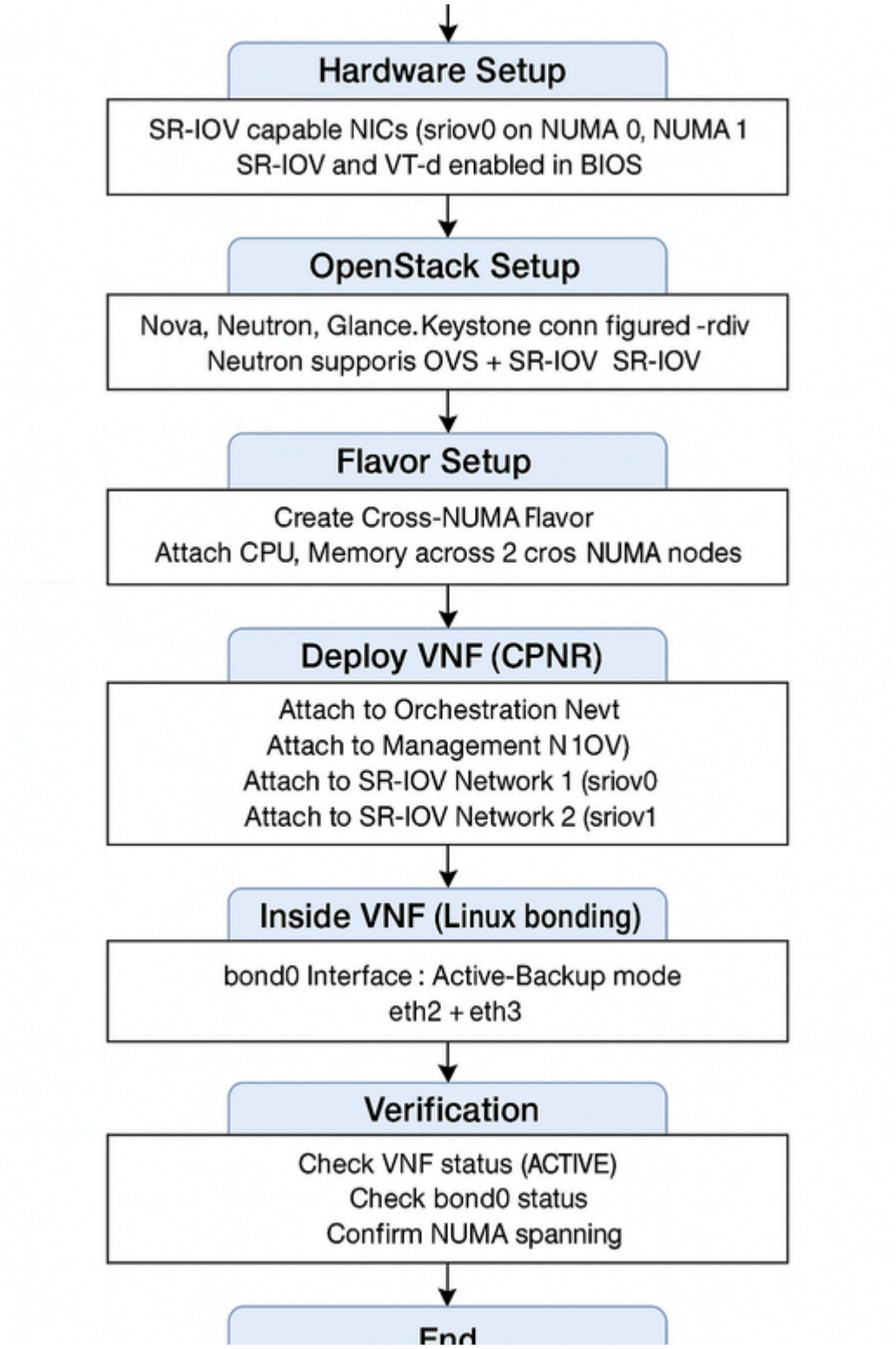
Accès racine ou administratif pour configurer la mise en réseau sur l'hôte Linux et au sein du VNF.

Présentation de l'architecture

Diagramme de connectivité d'interface réseau VNF



Organigramme



SRIOV

Exemples de configurations

Voici un exemple de code XML de Cisco Elastic Services Controller (ESC), démontrant le déploiement d'une machine virtuelle (VM) à l'aide de la mise en réseau SR-IOV (avec type>direct</type>) pour un locataire nommé test-tenant et une VM nommée sriov-vm-1. Cela inclut plusieurs interfaces, y compris les interfaces SR-IOV (directes) :

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
    <tenants>
        <tenant>
            <name>test-tenant</name>
            <managed_resource>true</managed_resource>
            <vim_mapping>false</vim_mapping>
            <deployments>
                <deployment>
                    <name>sriov-vm-deployment</name>
                    <vm_group>
                        <name>sriov-vm-1-group</name>
                        <locator>
                            <vim_id>vim1</vim_id>
                            <vim_project>default</vim_project>
                        </locator>
                        <image>sriov-image</image>
                        <flavor>custom-flavor</flavor>
                        <bootup_time>300</bootup_time>
                        <recovery_wait_time>30</recovery_wait_time>
                        <recovery_policy>
                            <action_on_recovery>REBOOT_ONLY</action_on_recovery>
                        </recovery_policy>
                        <interfaces>
                            <!-- Management Interface -->
                            <interface>
                                <nicid>0</nicid>
                                <network>mgmt-net</network>
                                <ip_address>192.x.x.x</ip_address>
                            </interface>

                            <!-- SR-IOV Interface 1 -->
                            <interface>
                                <nicid>1</nicid>
                                <type>direct</type>
                                <network>sriov-net-1</network>
                                <addresses>
                                    <address>
                                        <address_id>0</address_id>
                                        <subnet>sriov-subnet-1</subnet>
                                        <ip_address>10.x.x.x</ip_address>
                                    </address>
                                </addresses>
                            </interface>

                            <!-- SR-IOV Interface 2 -->
                            <interface>
                                <nicid>2</nicid>
                                <type>direct</type>
                                <network>sriov-net-2</network>
                                <addresses>
                                    <address>
                                        <address_id>0</address_id>
                                        <subnet>sriov-subnet-2</subnet>
                                        <ip_address>10.y.y.y</ip_address>
                                    </address>
                                </addresses>
                            </interface>
                        </interfaces>
```

```
                        <scaling>
                            <min_active>1</min_active>
                            <max_active>1</max_active>
                            <elastic>false</elastic>
                        </scaling>
                        <config_data>
                            <configuration>
                                <dst>--user-data</dst>
                                <file>file://tmp/init/sriov-vm-1.cfg</file>
                            </configuration>
                        </config_data>
                    </vm_group>
                </deployment>
            </deployments>
        </tenant>
    </tenants>
</esc_datamodel>
```

Points clés

- type>direct</type> sous <interface> active SR-IOV (PCI passthrough) pour cette carte réseau.
- Chaque interface SR-IOV possède son propre réseau et sous-réseau.
- Vous pouvez associer IPv4/IPv6 dans <adresses> selon les besoins.

Exemple de fichier Day0 à transmettre sur le fichier XML Cisco ESC :

```
Content-Type: multipart/mixed; boundary="===============2678395050260980330=="
MIME-Version: 1.0`

--===============2678395050260980330==
MIME-Version: 1.0
Content-Type: text/cloud-boothook; charset="us-ascii"

#cloud-boothook
#!/bin/bash
if [ ! -f /etc/cloud/cloud.cfg.orig ]; then
cp /etc/cloud/cloud.cfg /etc/cloud/cloud.cfg.orig
cp /etc/cloud/cloud.cfg.norootpasswd /etc/cloud/cloud.cfg
fi

--===============2678395050260980330==
MIME-Version: 1.0
Content-Type: text/cloud-config; charset="us-ascii"

#cloud-config
hostname: cpnr
ssh_authorized_keys:
 - ssh-rsa TEST1KEYAAAAB3NzaC1yc2EAAAADAQABAAACAQC7pf8gvOWH/Zv8iAlTv6LWEiPGA3B6t96G6LwTHF6iXOqQxyIUkg8I
 - ssh-rsa TEST2KEYAAAAB3NzaC1yc2EAAAADAQABAAACAQDAmkQGCZUrYqkZOC0J9t7mF9La9zYOqfzzFkk1wWtPga+aANOaFgjq
runcmd:
 - /usr/sbin/useradd Test1 -d /home/Test1 -s /bin/bash -g users; (/bin/echo changeme; /bin/echo changem
```

```
 - nmcli con add type ethernet con-name eth0 ifname eth0 ip4 10.xx.xx.xx/24
 - nmcli con add type ethernet con-name eth1 ifname eth1 ip4 172.xx.xx.xx/23
 - nmcli connection add type bond con-name bond0 ifname bond0 bond.options "mode=active-backup,miimon=1
 - nmcli connection add type ethernet ifname ens6 master bond0
 - nmcli connection add type ethernet ifname ens7 master bond0
 - nmcli con up eth0
 - nmcli con up eth1
 - nmcli con up bond-slave-ens6
 - nmcli con up bond-slave-ens7
 - nmcli con down bond0
 - nmcli con up bond0
 - nmcli connection reload
 - hostnamectl set-hostname CPNRDNSTEST

--===============2678395050260980330==
```

Déploiement de CPNR VNF avec ports SR-IOV et interface de liaison de sauvegarde active sur OpenStack

CPNR est une fonction de réseau virtuel (VNF) vitale qui fournit des services de gestion des adresses IP (IPAM), DHCP et DNS (Domain Name Server) pour les réseaux d'entreprise et de fournisseurs de services. Le déploiement de CPNR en tant que VNF dans OpenStack nécessite une planification minutieuse, en particulier lors de l'utilisation de ports SR-IOV, de configurations inter-NUMA, et d'une interface de liaison Active-Backup pour la redondance et les performances.

Cet article explique le processus détaillé de déploiement du VNF CPNR sur OpenStack. Il comprend :

- Configuration du mode Cross-NUMA, qui est essentiel pour accéder aux cartes réseau SR-IOV à partir de plusieurs noeuds NUMA.
- Configuration de la liaison Active-Backup, garantissant une haute disponibilité sans nécessiter de configurations côté commutateur.
- Configuration des réseaux OpenStack, des saveurs et de la présentation.
- Planification des adresses IP, configuration de la mise en réseau Linux à l'aide de fichiers ifcfg et déploiement du VNF à l'aide de Cisco ESC.

Principales caractéristiques du déploiement

1. Sensibilisation à l'AMNUM :
    - Le VNF CPNR couvre les noeuds NUMA pour accéder aux cartes réseau SR-IOV (sriov0 sur NUMA 0 et sriov1 sur NUMA 1).
    - Le mode Cross-NUMA est requis car, en mode Single-NUMA, OpenStack permet uniquement à un VNF de se connecter à des cartes réseau physiquement situées sur

le même noeud NUMA où le VNF est lancé. En activant le mode NUMA croisé, le VNF peut utiliser les cartes réseau et les ressources des deux noeuds NUMA.

2. Liaison Active-Backup :

   - Une interface bond0 est créée à l'aide de cartes réseau SR-IOV (eth2 de sriov0 et eth3 de sriov1).
   - Le mode de sauvegarde active assure la redondance et la tolérance aux pannes sans nécessiter de configuration côté commutateur.

3. Réseau OpenStack :

   - Orchestration et réseaux de gestion : Basé sur Openvswitch pour le trafic de contrôle et d'administration.
   - Réseaux d'applications/services : SR-IOV pour le trafic hautes performances.

Pourquoi le mode Cross-NUMA est requis

1. Réseau compatible NUMA dans OpenStack

NUMA est une architecture de mémoire dans laquelle chaque processeur (et sa mémoire locale et ses périphériques) est regroupé en un noeud NUMA. Dans OpenStack, le positionnement compatible NUMA garantit que les VNF sont affectés de manière optimale aux ressources sur le même noeud NUMA afin de réduire la latence et d'optimiser les performances.

- Les cartes réseau SR-IOV sont NUMA-Local :
  - Chaque carte réseau physique est liée à un noeud NUMA spécifique. Exemple :
    - sriov0 est connecté au noeud NUMA 0.
    - sriov1 est connecté au noeud NUMA 1.

- Limitation du mode NUMA unique :
  - Lorsqu'un VNF est lancé en mode NUMA unique, OpenStack permet uniquement au VNF de se connecter aux cartes réseau locales au noeud NUMA où le VNF est lancé. Cela signifie :
    - Si le VNF est lancé sur NUMA 0, il peut seulement se connecter aux NIC sur sriov0.
    - Si le VNF est lancé sur NUMA 1, il ne peut se connecter qu'aux cartes réseau sur sriov1.

2. Pourquoi le mode Cross-NUMA est nécessaire

Le VNF CPNR nécessite l'accès à :

- Réseau d'orchestration (Openvswitch, agnostique NUMA)
- Réseau de gestion (Openvswitch, indépendant de NUMA)
- Réseau 1 SR-IOV : Connecté à sriov0 (noeud NUMA 0)
- Réseau SR-IOV 2 : Connecté à sriov1 (noeud NUMA 1).

Dans ce déploiement, le VNF CPNR nécessite l'accès aux cartes réseau SR-IOV à partir de NUMA 0 (sriov0) et de NUMA 1 (sriov1) pour assurer la redondance et la haute disponibilité. Pour ce faire :

- Le VNF doit être lancé en mode Cross-NUMA, ce qui permet à OpenStack d'allouer le CPU, la mémoire et les cartes réseau à partir de plusieurs noeuds NUMA.
- Cela garantit que le VNF peut se connecter aux cartes réseau sur sriov0 et sriov1, permettant l'utilisation des deux ports SR-IOV dans une configuration de liaison Active-Backup.

Limitation de la taille des connecteurs pour les ports OVS

Qu'est-ce que Conntrack ?

Conntrack est une fonctionnalité du noyau Linux utilisée pour suivre les connexions réseau, en particulier pour la traduction d'adresses de réseau (NAT) et les règles de pare-feu. Pour les ports basés sur OVS dans OpenStack, conntrack est utilisé pour gérer l'état de connexion et appliquer les règles de groupe de sécurité.

Impact de Conntrack sur les ports OVS

1. Table de suivi :
    - Chaque connexion active consomme une entrée dans la table contrack.
    - La taille de la table conntrack est limitée par le paramètre nf_conntrack_max.

2. Limite par défaut :
    - Par défaut, la taille de la table conntrack est de 65536 entrées. Pour les charges de travail avec des débits de connexion élevés (par exemple, les VNF avec de nombreux

flux simultanés), cette limite peut être rapidement épuisée, ce qui entraîne l'abandon de paquets.

3. Impact sur les ports OVS :

   - Si la table conntrack est pleine, les nouvelles connexions sont abandonnées, ce qui peut avoir un impact important sur les performances VNF.
   - Cela est particulièrement pertinent pour les réseaux Orchestration et Management, qui utilisent des ports OVS.

Comment réduire les limites de connexion

1. Augmenter la taille de la table Conntrack :

   - Afficher la limite actuelle :

     ```
     sysctl net.netfilter.nf_conntrack_max
     ```

   - Augmentez la limite :

     ```
     sysctl -w net.netfilter.nf_conntrack_max=262144
     ```

   - Rendre la modification persistante :

     ```
     echo "net.netfilter.nf_conntrack_max=262144" >> /etc/sysctl.conf
     ```

2. Surveiller l'utilisation du suivi :

   Vérifiez les statistiques de suivi :

   ```
   cat /proc/sys/net/netfilter/nf_conntrack_count
   ```

3. Optimiser les règles de groupe de sécurité :

   Réduisez le nombre de règles appliquées aux ports OVS afin de minimiser la surcharge de suivi de connexion.

Comment SR-IOV résout les problèmes de suivi

1. Élimine la dépendance Conntrack

Les ports SR-IOV contournent le chemin de données OVS et les fonctionnalités du noyau Linux telles que conntrack. Cela supprime entièrement la surcharge de suivi de connexion.

2. Évolutivité supérieure

Contrairement aux ports OVS, qui sont limités par la taille de la table conntrack (nf_contrack_max), les ports SR-IOV peuvent traiter un nombre pratiquement illimité de connexions.

3. Latence réduite

En déchargeant le traitement des paquets sur le matériel de la carte réseau, les ports SR-IOV éliminent la latence introduite par le traitement logiciel contrack.

Pourquoi le mode de sauvegarde active est choisi pour les ports SR-IOV sur la machine virtuelle CPNR

Le mode de liaison Active-Backup est particulièrement bien adapté à ce déploiement en raison de sa simplicité, de sa tolérance aux pannes et de sa compatibilité avec les interfaces SR-IOV. Voici pourquoi :

1. Redondance sans complexité

- Mode de sauvegarde active : Une seule interface (l'interface active) transmet et reçoit du trafic à un moment donné. Les autres interfaces restent en mode veille.
- Si l'interface active échoue (par exemple, en raison d'une défaillance de liaison ou d'un problème matériel), la liaison bascule automatiquement vers une interface de secours. Cela garantit une connectivité réseau continue sans intervention manuelle.

2. Aucun LAG (Link Aggregation Group) requis

- Contrairement aux autres modes de liaison (par exemple, 802.3ad ou balance-alb), le mode

de sauvegarde active ne nécessite pas de configuration LACP (Link Aggregation Control Protocol) ou côté commutateur.
- Ceci est particulièrement important pour les ports SR-IOV, car les VF SR-IOV ne prennent généralement pas en charge les configurations LACP ou LAG.

3. Basculement transparent

- Le basculement est quasi instantané, avec un minimum d'interruption du trafic.
- Lorsque l'interface active échoue, la liaison promeut immédiatement une interface de secours à l'état actif.

4. Indépendance matérielle

Le mode de sauvegarde active fonctionne indépendamment des commutateurs physiques ou du matériel sous-jacents. La logique de basculement réside entièrement dans le noyau Linux, ce qui le rend très portable et polyvalent.

5. Optimisé pour SR-IOV

Les VF SR-IOV sont liées à des cartes réseau physiques et à des noeuds NUMA spécifiques. En utilisant le mode Active-Backup, vous pouvez combiner des VF de différents noeuds NUMA dans une interface de liaison logique unique (bond0). Cela garantit une haute disponibilité tout en utilisant efficacement les ressources NUMA.

Le mode Active-Backup est l'un des modes les plus simples et les plus répandus dans le domaine de l'agrégation Linux. Il est conçu pour fournir une haute disponibilité en garantissant que le trafic continue à circuler de manière transparente même si l'une des interfaces liées échoue. Il s'agit d'une explication approfondie du fonctionnement du mode de sauvegarde active, de ses caractéristiques clés et de ses avantages.

Qu'est-ce qu'une interface Linux Bond ?

Une interface de liaison dans Linux combine deux interfaces réseau ou plus dans une interface logique unique. Cette interface logique, appelée liaison (par exemple, liaison0), est utilisée pour fournir :

- Redundancy: Garantir une haute disponibilité de la connectivité réseau.
- Amélioration des performances : Dans d'autres modes (par exemple, balance-rr ou 802.3ad), il peut également agréger la bande passante.

Comment fonctionne le mode de sauvegarde active ?

En mode de sauvegarde active, une seule interface (appelée interface active) est utilisée à un moment donné pour transmettre et recevoir du trafic. La ou les autres interfaces restent en mode veille. Si l'interface active échoue, l'une des interfaces de secours est promue à l'état active, et le trafic est automatiquement réacheminé vers la nouvelle interface active.

Principales caractéristiques du mode de sauvegarde active

1. Interface active unique :
    - À un moment donné, une seule interface physique dans la liaison est active pour transmettre et recevoir le trafic.
    - Les interfaces de secours sont totalement passives, sauf en cas de basculement.

2. Basculement automatique :
    - Si l'interface active échoue (par exemple, en raison d'un problème matériel, d'une déconnexion de câble ou d'une défaillance de liaison), la liaison bascule automatiquement vers une interface de secours.
    - Le basculement est transparent et ne nécessite aucune intervention manuelle.

3. Prise en charge de re-basculement :

    Une fois l'interface défaillante restaurée, elle peut redevenir active automatiquement (si elle est configurée pour le faire) ou rester en mode veille, selon la configuration de liaison.

4. Aucune configuration requise côté commutateur :
    - Contrairement aux autres modes de liaison (par exemple, 802.3ad ou balance-rr), le mode Active-Backup ne nécessite aucune configuration spéciale sur les commutateurs physiques (par exemple, LAG ou LACP).
    - Cela le rend idéal pour les scénarios où la configuration côté commutateur n'est pas possible ou lors de la liaison des fonctions virtuelles SR-IOV, qui ne prennent pas en charge LAG.

5. Surveillance :
    - La liaison surveille en permanence l'intégrité de toutes les interfaces membres à l'aide du paramètre miimon(Media Independent Interface Monitor).
    - Si une défaillance de liaison est détectée, la liaison passe immédiatement à une

interface de secours saine.

Flux du trafic en mode de sauvegarde active

Fonctionnement normal

1. Interface active :

    - Le trafic passe exclusivement par l'interface active (par exemple, eth2 dans une liaison de eth2 et eth3).
    - L’interface de secours (eth3) reste inactive et ne transmet ni ne reçoit de trafic.

2. Surveillance :

    - La liaison surveille périodiquement l'état de toutes les interfaces membres. Pour ce faire, utilisez :
        - miimon : Vérifie l’état de la liaison de chaque interface à un intervalle configurable (par exemple, toutes les 100 ms).
        - Surveillance ARP (Address Resolution Protocol) (facultatif) : Envoie des requêtes ARP pour s'assurer que l'interface active est accessible.

Scénario de basculement

1. Échec de liaison sur l'interface active :

    Si l'interface active (eth2) tombe en panne (par exemple, débranchement du câble, défaillance matérielle de la carte réseau ou liaison désactivée), la liaison détecte immédiatement la panne à l'aide de la surveillance miimon ou ARP.

2. Basculement automatique :

    - La liaison passe à l'interface de secours (eth3), qui devient la nouvelle interface active.
    - Le trafic est réacheminé via la nouvelle interface active sans intervention manuelle.

3. Délai de basculement :

    Le processus de basculement est quasi-instantané (généralement en quelques millisecondes, selon l'intervalle miimon).

Scénario de restauration

1. Restauration de l'interface défaillante :

    - Lorsque l'interface précédemment défaillante (eth2) est restaurée, elle peut :
        - Récupérer automatiquement le rôle actif (s'il est configuré pour le faire).
        - Restez en mode veille (comportement par défaut).

2. Continuité du trafic :

    Le re-basculement est transparent et ne perturbe pas les flux de trafic en cours.

Scénario: Liaison de secours active avec ports SR-IOV

Le mode de sauvegarde active est particulièrement adapté aux interfaces SR-IOV pour les raisons suivantes :

- Les VF SR-IOV ne prennent généralement pas en charge les protocoles d'agrégation de liens comme LACP.
- La liaison en mode de sauvegarde active peut fournir une redondance sans configuration côté commutateur.

Exemple :

- eth2 est mappé à une VF SR-IOV sur sriov0 (noeud NUMA 0).
- eth3 est mappé à une VF SR-IOV sur sriov1 (noeud NUMA 1).
- La liaison (bond0) combine ces interfaces, fournissant un basculement transparent entre les VF SR-IOV.

Étape 1. Mise en réseau OpenStack

Le VNF CPNR nécessite les quatre réseaux suivants :

1. Réseau d'orchestration : Pour le trafic de contrôle et d'orchestration (basé sur Openvswitch).
2. Réseau de gestion : Pour l'accès administratif (basé sur Openvswitch).
3. Réseau 1 SR-IOV : Trafic d'application/service sur sriov0.
4. Réseau SR-IOV 2 : Trafic d'application/de service sur sriov1.

Voici la procédure de déploiement pas à pas :

Étape 1.1. Création De Réseaux Openvswitch

- Réseau d'orchestration :

```
openstack network create --provider-network-type vxlan orchestration-network
```

- Réseau de gestion :

```
openstack network create --provider-network-type vxlan management-network
```

Étape 1.2. Création de sous-réseaux pour les réseaux Openvswitch

- Sous-réseau d'orchestration :

```
openstack subnet create --network orchestration-network \
--subnet-range 192.xx.xx.0/24 orchestration-subnet
```

- Sous-réseau de gestion :

```
openstack subnet create --network management-network \
--subnet-range 10.x.x.0/24 management-subnet
```

Étape 1.3. Création de réseaux SR-IOV

- Réseau 1 SR-IOV :

```
openstack network create --provider-network-type vlan \
--provider-physical-network sriov0 --provider-segment 101 sriov-network-1
```

- Réseau SR-IOV 2 :

```
openstack network create --provider-network-type vlan \
--provider-physical-network sriov1 --provider-segment 102 sriov-network-2
```

Étape 2. Saveurs OpenStack

Étape 2.1. Création d’une saveur Cross-NUMA

Afin de s'assurer que le VNF peut accéder aux cartes réseau SR-IOV à partir des deux noeuds NUMA, créez un parfum avec la prise en charge de plusieurs NUMA :

```
openstack flavor create --ram 8192 --vcpus 4 --disk 40 cross-numa-flavor
```

Étape 2.2. Configuration des propriétés NUMA

Définissez les propriétés spécifiques à NUMA :

```
openstack flavor set cross-numa-flavor \
--property hw:numa_nodes=2 \
--property hw:cpu_policy=dedicated \
--property hw:mem_page_size=large
```

Étape 3 : configuration de la liaison en mode de sauvegarde active

Après avoir lancé le VNF, configurez l'interface de liaison pour les ports SR-IOV (eth2 et eth3) sur le VNF.

Étape 3.1. Configuration de l’interface de liaison

Créez une interface bond (bond0) en mode Active-Backup :

```
vi /etc/sysconfig/network-scripts/ifcfg-bond0
```

```
DEVICE=bond0
BOOTPROTO=static
ONBOOT=yes
BONDING_OPTS="mode=active-backup miimon=100"
IPADDR=172.xx.xx.10
NETMASK=255.xx.xx.0
GATEWAY=172.xx.xx.1
```

Étape 3.2. Configuration des interfaces esclaves

- eth2 :

  ```
  vi /etc/sysconfig/network-scripts/ifcfg-eth2
  ```

  ```
  DEVICE=eth2
  ONBOOT=yes
  MASTER=bond0
  SLAVE=yes
  ```

- eth3 :

  ```
  vi /etc/sysconfig/network-scripts/ifcfg-eth3
  ```

  ```
  DEVICE=eth3
  ONBOOT=yes
  ```

```
MASTER=bond0
SLAVE=yes
```

Étape 3.3. Application de la configuration

Redémarrez le service réseau afin d'appliquer la configuration :

```
systemctl restart network
```

Vérifier

Après avoir déployé le VNF, vérifiez son fonctionnement en procédant comme suit :

1. Vérifier l'état VNF

Vérifiez que l'instance VNF est active :

```
openstack server show cpnr-instance
```

Vérifiez que l'état est ACTIVE.

2. Vérification de la connectivité réseau

- Test Ping : Vérifiez que le VNF peut communiquer sur tous les réseaux :

```
ping <IP_ADDRESS_OF_ORCHESTRATION_NETWORK>
ping <IP_ADDRESS_OF_MANAGEMENT_NETWORK>
```

- Interface de liaison :
    - Confirmez que bond0 est actif :

```
cat /proc/net/bonding/bond0
```

Recherchez :

- Esclave actuellement actif : Indique l'interface active.
- Interface esclave : Confirme que eth2 et eth3 font partie de la liaison.

3. Vérifier le positionnement NUMA

Assurez-vous que le VNF utilise les ressources des deux noeuds NUMA :

```
nova show <INSTANCE_ID> --human | grep numa
```

Meilleures pratiques

- Surveillance et dépannage : Utilisez des outils comme tcpdump et ethtool afin de surveiller les interfaces SR-IOV.
- Sécurité: Gérez soigneusement l'accès au réseau physique et appliquez un isolement strict entre les locataires.
- Évolutivité : Planifiez la capacité physique de la carte réseau lors de l'évolution des déploiements SR-IOV, car le nombre de VF disponibles est limité par le matériel de la carte réseau.

Dépannage

Si le déploiement ne fonctionne pas comme prévu, reportez-vous aux étapes de dépannage suivantes :

1. Vérification de la configuration SR-IOV

- Vérifiez si SR-IOV est activé dans le BIOS :

```
dmesg | grep -i "SR-IOV"
```

- Vérifiez que les VF sont créés sur les cartes réseau :

```
lspci | grep Ethernet
```

2. Vérifier le positionnement NUMA

Si le VNF ne peut pas accéder aux deux cartes réseau, assurez-vous que le mode Cross-NUMA est activé :

- Vérifiez les propriétés NUMA de l'arôme :

```
openstack flavor show cross-numa-flavor
```

3. Problèmes d’interface de liaison

- Vérifiez le statut de la liaison :

```
cat /proc/net/bonding/bond0
```

- Si la liaison ne fonctionne pas :
  - Assurez-vous que les interfaces esclaves (eth2 et eth3) sont correctement configurées dans le cadre de la liaison.
  - Redémarrez le service réseau :

```
systemctl restart network
```

4. Problèmes de connectivité réseau

- Vérifiez les liaisons des ports OpenStack :

```
openstack port list --server cpnr-instance
```

- Vérifiez la configuration IP correcte dans le VNF :

```
ip addr show
```

Conclusion

Le déploiement de VNF CPNR sur OpenStack avec des ports SR-IOV nécessite le mode Cross-NUMA afin de permettre au VNF de se connecter aux cartes réseau à partir des deux noeuds NUMA. C'est essentiel parce qu'OpenStack limite les VNF en mode NUMA unique pour accéder uniquement aux ressources (cartes réseau, processeurs, mémoire) dans le noeud NUMA où le VNF est lancé. L'association du mode NUMA croisé et de la liaison Active-Backup garantit une haute disponibilité, une tolérance aux pannes et une utilisation efficace des ressources, ce qui rend ce déploiement hautement résilient et performant.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.