Implementeer DNS VNF met SRIOV Network op Openstack CVIM - Configuratievoorbeeld voor Prime Network Registrar (DNS)

Inhoud

Inleiding

Dit document beschrijft de stapsgewijze implementatie van CPNR op OpenStack Cisco Virtualized Infrastructure Manager (CVIM) met behulp van SR-IOV en Active-Backup bonding.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Bekendheid met OpenStack en Single Root Input/Output Virtualization (SR-IOV) concepten
- Werkende kennis over Cisco Virtual Interface Manager (VIM) en Cisco Elastic Services Controller en Linux-opdrachten en netwerken

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

In het huidige netwerklandschap spelen Virtual Network Functions (VNF's) een cruciale rol bij het mogelijk maken van flexibele, schaalbare en efficiënte netwerkservices. Voor VNF's die een krachtige netwerkconnectiviteit vereisen, is SR-IOV een veelgebruikte technologie. Met SR-IOV kunnen VNF's de virtuele switch van de hypervisor omzeilen en rechtstreeks toegang krijgen tot de bronnen van de Physical Network Interface Controller (NIC), waardoor de latentie wordt verminderd en de doorvoer wordt verhoogd.

Configuratie

Voordat u doorgaat met de implementatie, moet u ervoor zorgen dat aan deze vereisten wordt voldaan.

1. Hardwarevereisten

- Voor SR-IOV geschikte NIC's:
  - Ten minste twee fysieke NIC's met SR-IOV-functionaliteit die geschikt zijn voor SR-IOV in de BIOS/Unified Extensible Firmware Interface (UEFI).
  - Voorbeeld: sriov0toegewezen aan niet-uniforme geheugentoegang (NUMA) node 0 en sriov1toegewezen aan NUMA node 1.

- NUMA-bewuste hosts:
  - Compute nodes moeten de NUMA-architectuur ondersteunen.
  - NUMA-ondersteuning moet zijn ingeschakeld in het BIOS/UEFI van de host.

2. Intel NIC-kaarten identificeren

Intel XL710 en E810CQDA2 NIC-kaarten worden vaak gebruikt voor hoogwaardige SR-IOV-

netwerken. Voer de volgende stappen uit om het NIC-kaartmodel op de host te controleren:

Stap 1. De opdracht lspci gebruiken

Voer deze opdracht uit om een lijst te maken van de apparaten voor randapparaatverbindingen (PCI's) die zijn gekoppeld aan netwerkcontrollers:

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

Voorbeeld uitvoer:

```
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)
```

Stap 2. XL710 verifiëren

Als de NIC Intel XL710 is, ziet u Ethernet-controller XL710 in de uitvoer.

Stap 3. E810CQDA2 verifiëren

Als de NIC Intel E810CQDA2 is, ziet u Ethernet-controller E810-Cin de uitvoer.

Stap 4. Stuurprogrammaondersteuning bevestigen

Om het gebruikte NIC-stuurprogramma te controleren, voert u het volgende uit:

```
ethtool -i
```

Voorbeeld van uitvoer voor XL710:

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

Voorbeeld van uitvoer voor E810CQDA2:

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

Zorg ervoor dat de driverversie overeenkomt met de compatibiliteitsmatrix voor uw OpenStack- en Linux-distributie.

3. BIOS-/UEFI-configuratie

- SR-IOV inschakelen:

Zorg ervoor dat SR-IOV is ingeschakeld in het BIOS/UEFI van de server.

- Virtualisatietechnologie inschakelen voor gerichte I/O (VT-d)/AMD-Vi:

Intel VT-d of AMD-Vi moet zijn ingeschakeld voor PCI-doorvoer en SR-IOV-functionaliteit.

4. OpenStack Setup

- Core OpenStack Services:

Zorg ervoor dat OpenStack-services zoals Nova, Neutron, Glance en Keystone zijn geïnstalleerd en geconfigureerd.

- Neutron configuratie:

Neutron moet zowel OpenSwitch (OVS) voor orkestratie-/beheernetwerken als SR-IOV voor applicatie-/servicenetwerken ondersteunen.

- SR-IOV-configuratie:

De compute nodes moeten worden geconfigureerd om SR-IOV te ondersteunen, waarbij virtuele functies (VF's) op de NIC's zijn gemaakt.

5. VNF-image van Cisco Prime Network Registrar (CPNR)

- Compatibiliteit met VNF-images:

Het CPNR VNF-image moet SR-IOV-interfaces ondersteunen en de benodigde stuurprogramma's bevatten.

- Uploaden naar Glance:

Zorg ervoor dat het CPNR VNF-image beschikbaar is in OpenStack Glance.

6. Administratieve toegang

- OpenStack CLI:

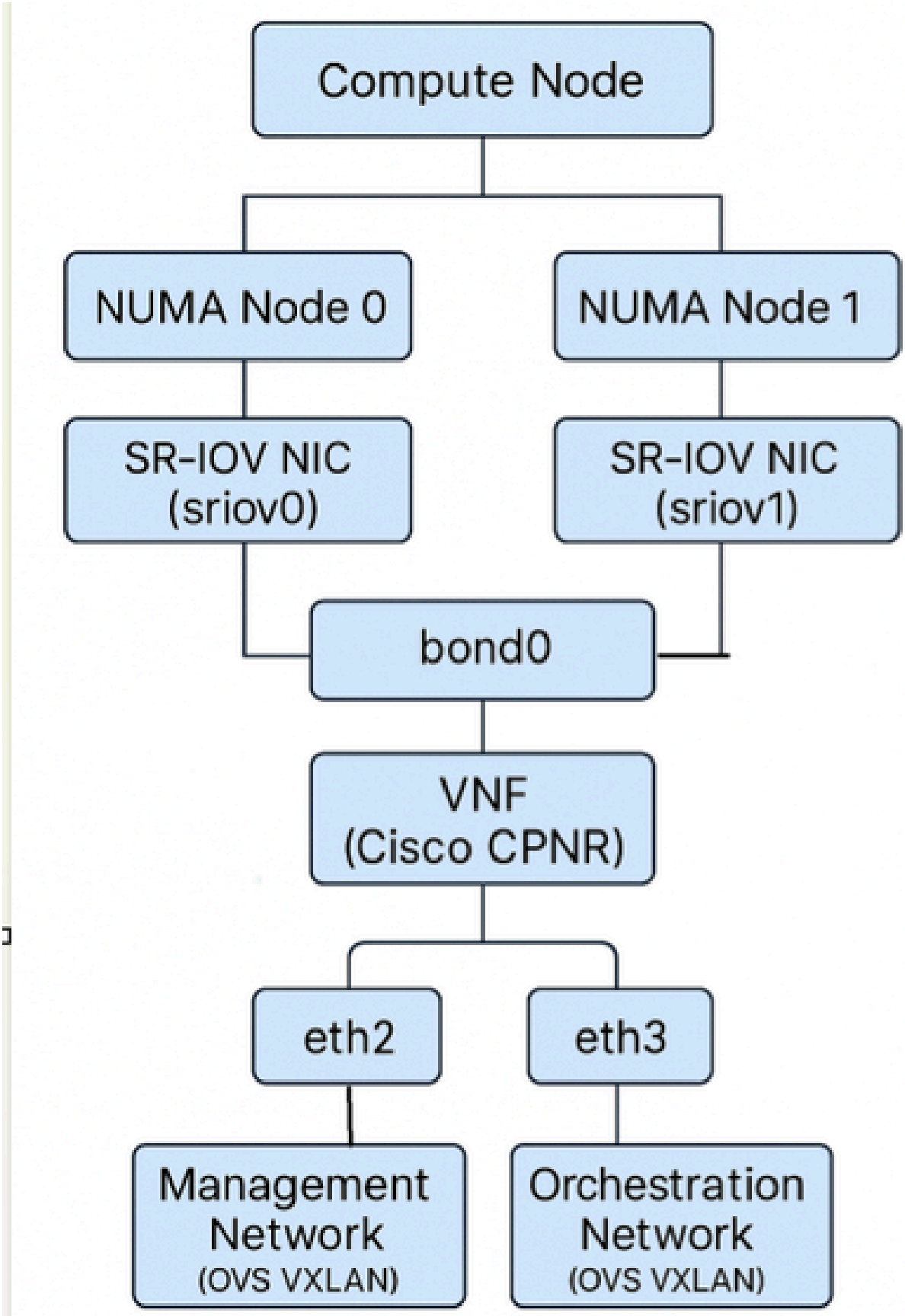
Zorg voor toegang tot de OpenStack CLI voor het maken van netwerken, smaken en het starten van de VNF.

- Root- of beheerdersrechten:

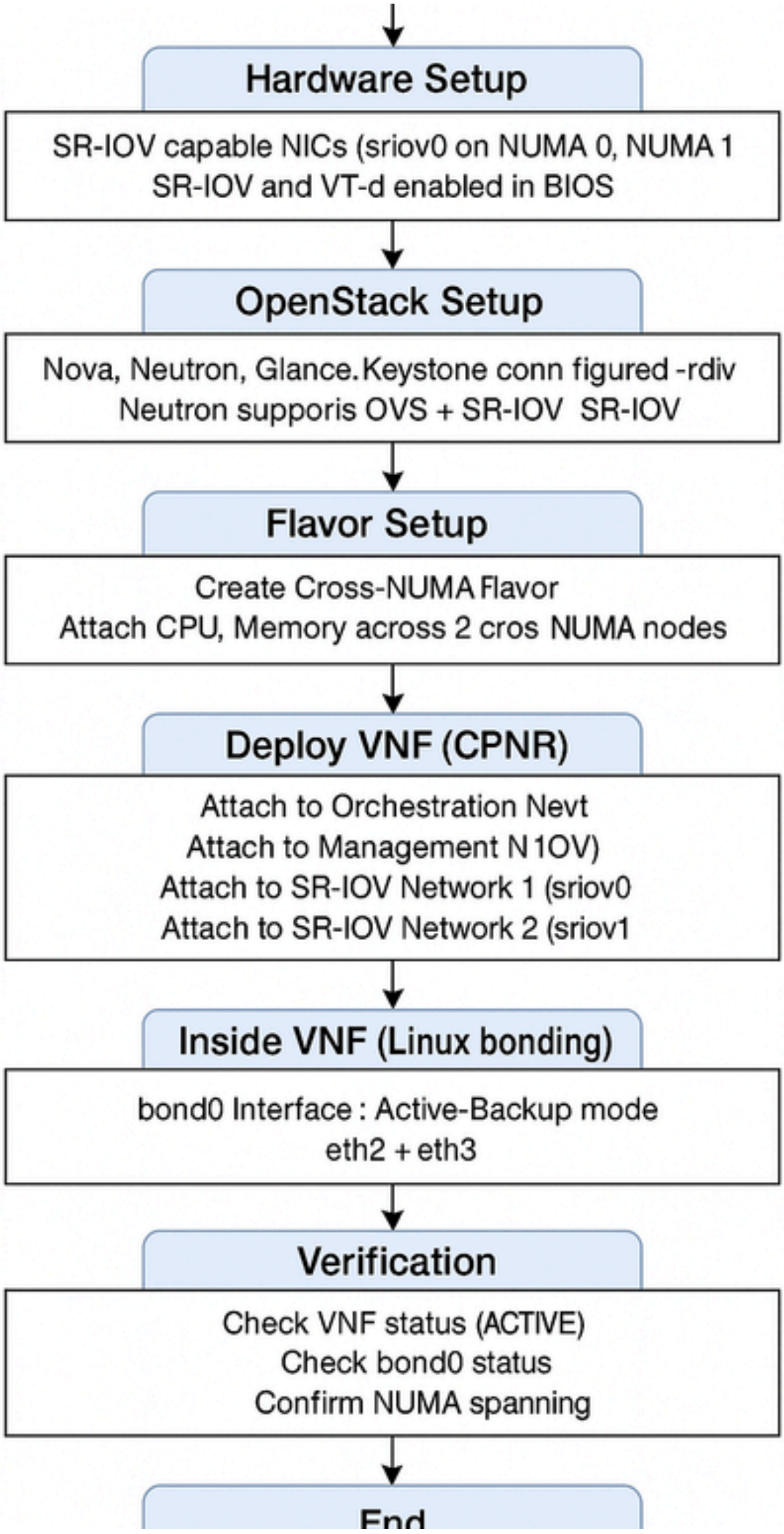
Root- of beheerderstoegang voor het configureren van netwerken op de Linux-host en binnen de VNF.

Architectuuroverzicht

VNF-netwerkinterfaceverbindingsdiagram



stroomschema


Hardware Setup
SR-IOV capable NICs (sriov0 on NUMA 0, NUMA 1
SR-IOV and VT-d enabled in BIOS
OpenStack Setup
Nova, Neutron, Glance.Keystone conn figured -rdiv
Neutron supporis OVS + SR-IOV SR-IOV
Flavor Setup
Create Cross-NUMA Flavor
Attach CPU, Memory across 2 cros NUMA nodes
Deploy VNF (CPNR)
Attach to Orchestration Nevt
Attach to Management N1OV)
Attach to SR-IOV Network 1 (sriov0
Attach to SR-IOV Network 2 (sriov1
Inside VNF (Linux bonding)
bond0 Interface : Active-Backup mode
eth2 + eth3
Verification
Check VNF status (ACTIVE)
Check bond0 status
Confirm NUMA spanning
End

test-tenant

true

false

sriov-vm-deployment

sriov-vm-1-group

vim1

default

sriov-image

custom-flavor

300

30

REBOOT_ONLY

0

mgmt-net

192.168.10.101

1

direct

sriov-net-1

0

sriov-subnet-1

10.10.10.10

2

direct

sriov-net-2

0

sriov-subnet-2

10.10.20.10

1

1

false

```
--user-data



file://tmp/init/sriov-vm-1.cfg
```

Belangrijkste punten

- type>direct</type> onder <interface> maakt SR-IOV (PCI-passthrough) voor die NIC mogelijk.
- Elke SR-IOV-interface heeft zijn eigen netwerk en subnet.
- U kunt IPv4/IPv6 koppelen in <adressen> als dat nodig is.

Voorbeeld van Day0-bestand om de Cisco ESC XML door te geven:

```
Content-Type: multipart/mixed; boundary="==========267839505026098033O=="
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 AAAAB3NzaC1yc2EAAAADAQABAAACAQC7pf8gvOWH/Zv8iAlTv6LWEiPGA3B6t96G6LwTHF6iXOqQxyIUkg8IkqZ6wNwx
 - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDAmkQGCZUrYqkZOC0J9t7mF9La9zYOqfzzFkk1wWtPga+aANOaFgjqbjj+VlBd
runcmd:
 - /usr/sbin/useradd FMLVL1 -d /home/FMLVL1 -s /bin/bash -g users; (/bin/echo changeme; /bin/echo chang
 - 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 CPNRDNS01CO

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

CPNR VNF implementeren met SR-IOV-poorten en Active-Backup Bond-interface op OpenStack

CPNR is een vitale Virtual Network Function (VNF) die IP Address Management (IPAM), DHCP en Domain Name Server (DNS)-services biedt voor netwerken van ondernemingen en serviceproviders. Het implementeren van CPNR als een VNF in OpenStack vereist een zorgvuldige planning, vooral bij het gebruik van SR-IOV-poorten, cross-NUMA-configuraties en een Active-Backup-verbindingsinterface voor redundantie en prestaties.

In dit artikel wordt het stapsgewijze proces voor de implementatie van de CPNR VNF op OpenStack uitgelegd. Het omvat:

- De cross-NUMA-modus configureren, die essentieel is voor toegang tot SR-IOV NIC's vanaf meerdere NUMA-knooppunten.
- Het instellen van Active-Backup-bonding, het garanderen van hoge beschikbaarheid zonder configuraties aan de switch.

- OpenStack-netwerken, smaken en Glance configureren.
- IP-adresplanning, Linux-netwerken configureren met ifcfg-bestanden en de VNF implementeren met Cisco ESC.

Belangrijkste kenmerken van de implementatie

1. Cross-NUMA Bewustzijn:

    - De CPNR VNF omvat NUMA-knooppunten voor toegang tot SR-IOV-NIC's (sriov0 op NUMA 0 en sriov1 op NUMA 1).
    - De cross-NUMA-modus is vereist omdat OpenStack in de single-NUMA-modus een VNF alleen toestaat om verbinding te maken met NIC's die zich fysiek op dezelfde NUMA-node bevinden waar de VNF wordt gestart. Door cross-NUMA-modus in te schakelen, kan de VNF gebruikmaken van NIC's en bronnen van beide NUMA-knooppunten.

2. Active Backup-binding:

    - Een bond0-interface wordt gemaakt met behulp van SR-IOV NIC's (eth2 van sriov0 en eth3 van sriov1).
    - De Active Backup-modus garandeert redundantie en fouttolerantie zonder configuraties aan de switch.

3. OpenStack Networking:

    - Orchestratie- en beheernetwerken: OpenSwitch-gebaseerd voor beheer en administratief verkeer.
    - Applicatie-/servicenetwerken: SR-IOV-gebaseerd voor high-performance verkeer.

Waarom is de Cross-NUMA-modus vereist

1. NUMA-Aware Networking in OpenStack

NUMA is een geheugenarchitectuur waarbij elke CPU (en zijn lokale geheugen en apparaten) is gegroepeerd in een NUMA-node. In OpenStack zorgt NUMA-bewuste plaatsing ervoor dat VNF's optimaal worden toegewezen aan bronnen op dezelfde NUMA-node om de latentie te minimaliseren en de prestaties te maximaliseren.

- SR-IOV NIC's zijn NUMA-lokaal:
    - Elke fysieke NIC is gekoppeld aan een specifieke NUMA-node. Voorbeeld:
        - sriov0 is verbonden met NUMA-knooppunt 0.
        - sriov1 is verbonden met NUMA-knooppunt 1.
- Beperking van de Single-NUMA-modus:

    - Wanneer een VNF wordt gestart in single-NUMA-modus, staat OpenStack alleen toe dat de VNF verbinding maakt met NIC's die lokaal zijn voor de NUMA-node waar de VNF wordt gestart. Dit betekent:
        - Als de VNF wordt gestart op NUMA 0, kan deze alleen verbinding maken met NIC's op sriov0.
        - Als de VNF wordt gestart op NUMA 1, kan deze alleen verbinding maken met NIC's op seriov1.

2. Waarom is de cross-NUMA-modus noodzakelijk?

De CPNR VNF vereist toegang tot:

- Orchestratienetwerk (OpenSwitch, NUMA-agnostisch)
- Beheernetwerk (OpenSwitch, NUMA-agnostisch)
- SR-IOV Netwerk 1: Verbonden tosriov0(NUMA knooppunt 0)
- SR-IOV Netwerk 2: Verbonden tosriov1 (NUMA knooppunt 1).

In deze implementatie vereist de CPNR VNF toegang tot SR-IOV NIC's van zowel NUMA 0 (sriov0) als NUMA 1 (sriov1) om redundantie en hoge beschikbaarheid te bieden. Om dit te bereiken:

- De VNF moet worden gestart in de cross-NUMA-modus, waarmee OpenStack CPU, geheugen en NIC's van meerdere NUMA-knooppunten kan toewijzen.
- Dit zorgt ervoor dat de VNF kan aansluiten op NIC's op sriov0 en sriov1, waardoor het gebruik van beide SR-IOV-poorten in een Active-Backup bond configuratie.

Beperking van de grootte van de verbinding voor OVS-poorten

Wat is Conntrack?

Conntrack is een Linuxkernelfunctie die wordt gebruikt om netwerkverbindingen te volgen, met name voor Network Address Translation (NAT) en firewallregels. Voor op OVS gebaseerde poorten in OpenStack wordt contrack gebruikt om de verbindingsstatus te beheren en de regels voor beveiligingsgroepen af te dwingen.

Hoe Conntrack de OVS-poorten beïnvloedt

1. Contraktabel:

    - Elke actieve verbinding verbruikt een item in de tabel.
    - De grootte van de contrack tabel wordt beperkt door de f_contrack_maxparameter.

2. Standaardlimiet:

    - Standaard is de grootte van de verbindingstabel 65536 items. Voor werkbelastingen met hoge verbindingssnelheden (bijvoorbeeld VNF's met veel gelijktijdige stromen) kan

deze limiet snel worden uitgeput, wat resulteert in gedropte pakketten.

3. Gevolgen voor OVS-poorten:

   - Als de aansluittafel vol is, worden nieuwe verbindingen verwijderd, wat de VNF-prestaties ernstig kan beïnvloeden.
   - Dit is met name relevant voor de netwerken Orchestration en Management, die gebruikmaken van OVS-poorten.

Hoe te beperken Conntrack Beperkingen

1. Grootte van Contrasttabel vergroten:

   - De huidige limiet bekijken:

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

   - Verhoog de limiet:

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

   - Maak de wijziging blijvend:

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

2. Gebruik van Contrack controleren:

   Contactstatistieken controleren:

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

3. Regels voor beveiligingsgroepen optimaliseren:

   Verminder het aantal regels dat wordt toegepast op OVS-poorten om de overhead van het netwerk tot een minimum te beperken.

Hoe SR-IOV Conntrack-problemen oplost

1. Elimineert afhankelijkheid van contacten

SR-IOV poorten omzeilen de OVS datapath en Linux kernel functies zoals contrack. Hiermee wordt de overhead voor het bijhouden van verbindingen volledig verwijderd.

2. Hogere schaalbaarheid

In tegenstelling tot OVS-poorten, die beperkt zijn door de grootte van de aanmeldtabel (nf_conntrack_max), kunnen SR-IOV-poorten een vrijwel onbeperkt aantal verbindingen aan.

3. Verminderde latentie

Door pakketverwerking te offloaden naar de NIC-hardware, elimineren SR-IOV-poorten de latentie die wordt geïntroduceerd door op software gebaseerde contactverwerking.

Waarom de Active-Backup-modus is gekozen voor SR-IOV-poorten op de CPNR VM

De Active-Backup-bindingsmodus is bijzonder geschikt voor deze implementatie vanwege de eenvoud, fouttolerantie en compatibiliteit met SR-IOV-interfaces. Dit is waarom:

1. Redundantie zonder complexiteit

- Active-Backup-modus: slechts één interface (de actieve interface) verzendt en ontvangt verkeer op een bepaald moment. De andere interface(s) blijven in de stand-bymodus.
- Als de actieve interface uitvalt (bijvoorbeeld door een verbindingsfout of hardwareprobleem), wordt de koppeling automatisch naar een stand-byinterface switches. Dit zorgt voor continue netwerkconnectiviteit zonder handmatige tussenkomst.

2. Geen Link Aggregation Group (LAG) vereist

- In tegenstelling tot andere bindingsmodi (bijvoorbeeld 802.3adorbalance-alb), vereist de Active-Backup-modus geen Link Aggregation Control Protocol (LACP) of configuraties aan de switch.
- Dit is vooral belangrijk voor SR-IOV-poorten, omdat SR-IOV VF's meestal geen LACP- of LAG-configuraties ondersteunen.

3. Naadloze failover

- Failover is bijna onmiddellijk, met minimale verstoring van het verkeer.
- Wanneer de actieve interface faalt, promoot de binding onmiddellijk een stand-by-interface naar actieve status.

4. Hardware-onafhankelijkheid

De Active Backup-modus werkt onafhankelijk van de onderliggende switches of hardware. De failover-logica bevindt zich volledig in de Linux-kernel, waardoor deze zeer draagbaar en veelzijdig is.

5. Geoptimaliseerd voor SR-IOV

SR-IOV VF's zijn gekoppeld aan specifieke fysieke NIC's en NUMA-knooppunten. Door de Active-Backup-modus te gebruiken, kunt u VF's van verschillende NUMA-knooppunten combineren tot één logische bindingsinterface (bond0). Dit zorgt voor een hoge beschikbaarheid en maakt efficiënt gebruik van NUMA-middelen.

De Active-Backup-modus is een van de eenvoudigste en meest gebruikte modi in Linux-bonding. Het is ontworpen om een hoge beschikbaarheid te bieden door ervoor te zorgen dat het verkeer naadloos blijft stromen, zelfs als een van de gekoppelde interfaces uitvalt. Dit is een uitgebreide uitleg van hoe de Active-Backup-modus werkt, de belangrijkste kenmerken en voordelen.

Wat is een Linux Bond Interface?

Een bindingsinterface in Linux combineert twee of meer netwerkinterfaces in één logische interface. Deze logische interface, aangeduid als de binding (bijvoorbeeld bond0), wordt gebruikt om te voorzien:

- Redundantie: zorgen voor hoge beschikbaarheid van netwerkconnectiviteit.
- Prestatieverbetering: in andere modi (bijvoorbeeld balans-fout802.3ad) kan het ook bandbreedte aggregeren.

Hoe werkt de Active-Backup-modus?

In de Active-Backup-modus wordt slechts één interface (de zogenaamde actieve interface) op een gegeven moment gebruikt om verkeer te verzenden en te ontvangen. De andere interface(s) blijven in de stand-bymodus. Als de actieve interface uitvalt, wordt een van de standby-interfaces gepromoveerd naar de actieve status en wordt het verkeer automatisch omgeleid naar de nieuwe actieve interface.

Belangrijkste kenmerken van de Active-Backup-modus

1. Eén actieve interface:
    - Op elk moment is er slechts één fysieke interface in de binding actief voor het verzenden en ontvangen van verkeer.
    - Standby-interfaces zijn volledig passief, tenzij er een failover plaatsvindt.

2. Automatische failover:
    - Als de actieve interface uitvalt (bijvoorbeeld als gevolg van een hardwareprobleem, kabelonderbreking of verbindingsfout), wordt de koppeling automatisch naar een

stand-byinterface switches.
- Failover is naadloos en vereist geen handmatige interventie.

3. Failback-ondersteuning:

   Zodra de mislukte interface is hersteld, kan deze automatisch opnieuw actief worden (indien geconfigureerd om dit te doen) of in de stand-bymodus blijven, afhankelijk van de verbindingsconfiguratie.

4. Geen eisen aan de Switch:

   - In tegenstelling tot andere bindingsmodi (bijvoorbeeld 802.3adorbalance-rr), vereist de Active Backup-modus geen speciale configuratie op de fysische switches (bijvoorbeeld LAG of LACP).
   - Dit maakt het ideaal voor scenario's waarbij switch-side configuratie niet mogelijk is of bij het koppelen van SR-IOV virtual functies, die LAG niet ondersteunen.

5. Monitoring:

   - De bond bewaakt continu de gezondheid van alle lidinterfaces met behulp van `de` parameter `imimon` (Media Independent Interface Monitor).
   - Als een link defect wordt gedetecteerd, de band onmiddellijk switches naar een gezonde stand-by-interface.

Hoe verkeer stroomt in de Active-Backup-modus

normale werking

1. Actieve interface:

   - Het verkeer stroomt uitsluitend via de actieve interface (bijvoorbeeld eth2 in een binding van eth2en deth3).
   - De standby-interface (eth3) blijft inactief en verzendt of ontvangt geen verkeer.

2. Monitoring:

   - De bond controleert periodiek de status van alle lidinterfaces. Dit wordt gedaan met behulp van:
     - miimon: Controleert de linkstatus van elke interface met een configureerbaar interval (bijvoorbeeld elke 100ms).
     - Address Resolution Protocol (ARP)-bewaking (optioneel): verzendt ARP-verzoeken om ervoor te zorgen dat de actieve interface bereikbaar is.

failoverscenario

1. Linkfout op actieve interface:

   Als de actieve interface (eth2) uitvalt (bijvoorbeeld kabel losgekoppeld, NIC-hardwarefout of link down), detecteert de binding de fout onmiddellijk met behulp van minimale ARP-bewaking.

2. Automatische failover:

   - De bond-switch wordt gekoppeld aan de standby-interface (eth3), die de nieuwe actieve interface wordt.
   - Het verkeer wordt omgeleid via de nieuwe actieve interface zonder dat handmatige interventie nodig is.

3. Failover-tijdigheid:

   Het failover-proces verloopt vrijwel onmiddellijk (meestal binnen enkele milliseconden, afhankelijk van het minimuminterval).

failbackscenario

1. Herstel van mislukte interface:

   - Wanneer de eerder mislukte interface (eth2) wordt hersteld, kan deze:
     - De actieve rol automatisch terugwinnen (indien hiervoor geconfigureerd).
     - Blijf in de stand-bymodus (standaardgedrag).

2. Verkeerscontinuïteit:

   Failback is naadloos en zorgt ervoor dat de doorlopende verkeersstromen niet worden verstoord.

Use Case: Active-Backup-binding met SR-IOV-poorten

De Active Backup-modus is bijzonder geschikt voor SR-IOV-interfaces omdat:

- SR-IOV VF's ondersteunen doorgaans geen koppelingsaggregatieprotocollen zoals LACP.
- De binding in de Active Backup-modus kan redundantie bieden zonder configuratie aan de switch.

Voorbeeld:

- eth2 is toegewezen aan een SR-IOV VF onserov0(NUMA-knooppunt 0).
- eth3 is gekoppeld aan een SR-IOV VF onserov1 (NUMA-knooppunt 1).
- De obligatie (bond0) combineert deze interfaces en biedt naadloze failover tussen SR-IOV VF's.

Stap 1. OpenStack Networking

De CPNR VNF vereist de volgende vier netwerken:

1. Orchestratienetwerk: voor controle- en orkestratieverkeer (op basis van OpenSwitch).
2. Beheernetwerk: voor administratieve toegang (op basis van OpenSwitch).
3. SR-IOV Netwerk 1: Toepassing/dienstverkeer op sriov0.
4. SR-IOV Netwerk 2: Toepassing/dienstverkeer op sriov1.

Stapsgewijze implementatie:

Stap 1.1. OpenSwitch-netwerken maken

- Orchestratienetwerk:

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

- Beheernetwerk:

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

Stap 1.2. Subnetten voor OpenSwitch-netwerken maken

- Orchestratie-subnet:

```
openstack subnet create --network orchestration-network \
--subnet-range 192.168.100.0/24 orchestration-subnet
```

- Subnet beheer:

```
openstack subnet create --network management-network \
--subnet-range 10.10.10.0/24 management-subnet
```

Stap 1.3. SR-IOV-netwerken maken

- SR-IOV-netwerk 1:

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

- SR-IOV-netwerk 2:

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

Stap 2. OpenStack-smaken

Stap 2.1. Een Cross-NUMA-smaak maken

Om ervoor te zorgen dat de VNF toegang heeft tot SR-IOV NIC's van beide NUMA-knooppunten, kunt u een smaak creëren met cross-NUMA-ondersteuning:

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

Stap 2.2. NUMA-eigenschappen configureren

NUMA-specifieke eigenschappen instellen:

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

Stap 3. Bonding configureren in Active Backup-modus

Nadat u de VNF hebt gestart, configureert u de bindingsinterface voor SR-IOV-poorten (eth2 en eth3) op de VNF.

Stap 3.1. Bond-interfaceconfiguratie

Maak een bond-interface (bond0) in de Active-Backup-modus:

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

```
DEVICE=bond0
BOOTPROTO=static
ONBOOT=yes
BONDING_OPTS="mode=active-backup miimon=100"
IPADDR=172.16.1.10
NETMASK=255.255.255.0
GATEWAY=172.16.1.1
```

Stap 3.2. Slave-interfaces configureren

- 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
```

Stap 3.3. Configuratie toepassen

Start de netwerkservice opnieuw op om de configuratie toe te passen:

```
systemctl restart network
```

Verifiëren

Controleer na het implementeren van de VNF de functionaliteit met behulp van de volgende stappen:

1. VNF-status controleren

Controleer of de VNF-instantie actief is:

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

Zorg ervoor dat de status ACTIEF is.

2. Controleer de netwerkconnectiviteit

- Ping Test: Controleer of de VNF over alle netwerken kan communiceren:

```
ping
```

```
ping
```

- Bond-interface:
    - Bevestig dat bond0 actief is:

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

Zoeken naar:

- Momenteel actieve slave: geeft de actieve interface aan.
- Slave Interface: Bevestigt dat zowel 2 als 3 deel uitmaken van de binding.

3. Verifieer NUMA plaatsing

Zorg ervoor dat de VNF middelen gebruikt van beide NUMA-knooppunten:

```
nova show
        --human | grep numa
```

beste praktijken

- Controle en probleemoplossing: gebruik tools zoals cpdumpandethtool om de SR-IOV-interfaces te bewaken.
- Beveiliging: Beheer de toegang tot het fysieke netwerk zorgvuldig en handhaaf strikte isolatie tussen huurders.
- Schaalvergroting: plan fysieke NIC-capaciteit bij het schalen van SR-IOV-implementaties, aangezien het aantal beschikbare VF's wordt beperkt door de NIC-hardware.

Problemen oplossen

Als de implementatie niet werkt zoals verwacht, raadpleegt u de volgende stappen voor probleemoplossing:

1. SR-IOV-configuratie controleren

- Controleer of SR-IOV is ingeschakeld in het BIOS:

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

- Bevestig dat VF's zijn gemaakt op NIC's:

```
lspci | grep Ethernet
```

2. Verifieer NUMA plaatsing

Als de VNF geen toegang heeft tot beide NIC's, moet u ervoor zorgen dat de modus voor cross-NUMA is ingeschakeld:

- Controleer NUMA eigenschappen van de smaak:

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

3. Obligatie-interface-emissies

- Controleer de obligatiestatus:

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

- Als de band niet functioneert:
  - Zorg ervoor dat de slave-interfaces (eth2en th3) correct zijn geconfigureerd als onderdeel van de binding.
  - Start de netwerkservice opnieuw op:

```
systemctl restart network
```

4. Problemen met netwerkconnectiviteit

- Controleer de OpenStack-poortbindingen:

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

- Controleer of de IP-configuratie in de VNF correct is:

```
ip addr show
```

Conclusie

Voor de implementatie van CPNR VNF op OpenStack met SR-IOV-poorten is cross-NUMA-modus vereist om de VNF in staat te stellen verbinding te maken met NIC's van beide NUMA-knooppunten. Dit is van essentieel belang omdat OpenStack VNF's in single-NUMA-modus beperkt tot alleen toegang tot bronnen (NIC's, CPU's, geheugen) binnen de NUMA-node waar de VNF wordt gestart. De combinatie van de cross-NUMA-modus met Active Backup-binding zorgt voor hoge beschikbaarheid, fouttolerantie en efficiënt gebruik van bronnen, waardoor deze implementatie zeer veerkrachtig en krachtig is.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.