# Bereitstellung von DNS VNF mit SRIOV-Netzwerk auf OpenStack CVIM - Konfigurationsbeispiel für Prime Network Registrar (DNS)

## Inhalt


Einleitung

Voraussetzungen

- Anforderungen
- Verwendete Komponenten

Hintergrundinformationen

Konfiguration

- 1. Hardware-Anforderungen
- 2.Identifizieren von Intel NIC-Karten
  - Schritt 1: Verwenden des Befehls lspci
  - Schritt 2: Überprüfen von XL710
  - Schritt 3: Überprüfen von E810CQDA2
  - Schritt 4: Bestätigen der Treiberunterstützung
- 3. BIOS-/UEFI-Konfiguration
- 4. OpenStack-Setup
- 5. Cisco Prime Network Registrar (CPNR) VNF-Image
- 6. Verwaltungszugriff

Architekturübersicht

- VNF-Netzwerkschnittstellen-Verbindungsdiagramm
- Flussdiagramm
- Beispielkonfigurationen
- Wichtigste Punkte

Bereitstellung von CPNR VNF mit SR-IOV-Ports und Active-Backup Bond-Schnittstelle auf OpenStack

Wichtigste Merkmale der Bereitstellung

Warum der NUMA-übergreifende Modus erforderlich ist

- 1. NUMA-Aware Networking in OpenStack
- 2. Warum ist der NUMA-übergreifende Modus erforderlich?

Größenbeschränkung für OVS-Ports aufheben

- Was ist Contrack?
- Auswirkungen von Contratrack auf OVS-Ports
- Abwehr von Constracking-Einschränkungen

So behebt SR-IOV Verbindungsprobleme

- 1. Beseitigt ConnTrack-Abhängigkeit
- 2. Höhere Skalierbarkeit

- 3. Reduzierte Latenz

Gründe für die Wahl des Aktiv-Backup-Modus für SR-IOV-Ports auf CPNR VM

- 1. Redundanz ohne Komplexität
- 2. Keine Link Aggregation Group (LAG) erforderlich
- 3. Nahtloses Failover
- 4. Hardware-Unabhängigkeit
- 5. Optimiert für SR-IOV

Was ist eine Linux Bond-Schnittstelle?

Wie funktioniert der Active-Backup-Modus?

- Wichtigste Funktionen des Aktiv-Backup-Modus

Datenverkehrsfluss im Aktiv-Backup-Modus

- Normalbetrieb
- Failover-Szenario
- Failback-Szenario

Anwendungsfall: Active-Backup-Bonding mit SR-IOV-Ports

Schritt 1: OpenStack-Netzwerke

- Schritt 1.1. OpenSwitch-Netzwerke erstellen
- Schritt 1.2: Erstellen von Subnetzen für OpenSwitch-Netzwerke
- Schritt 1.3. Erstellen von SR-IOV-Netzwerken

Schritt 2: OpenStack-Varianten

- Schritt 2.1. Erstellen einer NUMA-übergreifenden Variante
- Schritt 2.2. Konfigurieren der NUMA-Eigenschaften

Schritt 3: Bonding im Aktiv-Backup-Modus konfigurieren

- Schritt 3.1: Konfiguration der Anbindungsschnittstelle
- Schritt 3.2. Konfigurieren von Slave-Schnittstellen
- Schritt 3.3: Konfiguration anwenden

Überprüfung

- 1. VNF-Status überprüfen
- 2. Netzwerkverbindung überprüfen
- 3. NUMA-Platzierung überprüfen

Best Practices

Fehlerbehebung

- 1. Überprüfen der SR-IOV-Konfiguration
- 2. NUMA-Platzierung überprüfen
- 3. Probleme mit Anleiheschnittstellen
- 4. Probleme mit der Netzwerkverbindung

Schlussfolgerung


---

# Einleitung

Dieses Dokument beschreibt die schrittweise Bereitstellung von CPNR auf OpenStack Cisco Virtualized Infrastructure Manager (CVIM) mithilfe von SR-IOV und Active-Backup-Bonding.

# Voraussetzungen

## Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Vertrautheit mit OpenStack- und SR-IOV-Konzepten (Single Root Input/Output Virtualization)
- Kenntnisse in den Bereichen Cisco Virtual Interface Manager (VIM), Cisco Elastic Services Controller, Linux-Befehle und -Netzwerke

## Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen

# Hintergrundinformationen

In der aktuellen Netzwerklandschaft spielen virtuelle Netzwerkfunktionen (Virtual Network Functions, VNFs) eine entscheidende Rolle bei der Bereitstellung flexibler, skalierbarer und effizienter Netzwerkservices. SR-IOV ist eine weit verbreitete Technologie für VNFs, die Hochleistungs-Netzwerkverbindungen benötigen. Mit SR-IOV können VNFs den virtuellen Hypervisor-Switch umgehen und direkt auf physische NIC-Ressourcen (Network Interface Controller) zugreifen. Dies reduziert die Latenz und erhöht den Durchsatz.

# Konfiguration

Bevor Sie mit der Bereitstellung fortfahren, stellen Sie sicher, dass diese Voraussetzungen erfüllt sind.

## 1. Hardware-Anforderungen

- SR-IOV-fähige NICs:
  - Mindestens zwei SR-IOV-fähige physische NICs mit aktiviertem SR-IOV im BIOS/Unified Extensible Firmware Interface (UEFI)
  - Beispiel: sriov0mapped to Non-Uniform Memory Access (NUMA) node 0 und sriov1mapped to NUMA node 1.

- NUMA-fähige Hosts:
  - Rechenknoten müssen die NUMA-Architektur unterstützen.
  - Die NUMA-Unterstützung muss auf dem Host-BIOS/UEFI aktiviert sein.

## 2. Identifizieren von Intel NIC-Karten

Intel XL710 und E810CQDA2 NIC-Karten werden häufig für Hochleistungs-SR-IOV-Netzwerke verwendet. Um das NIC-Kartenmodell auf dem Host zu überprüfen, gehen Sie wie folgt vor:

Schritt 1: Verwenden des Befehls lspci

Führen Sie diesen Befehl aus, um Peripheral Component Interconnect (PCI)-Geräte für Netzwerk-Controller aufzulisten:

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

Ausgabebeispiel:

```
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)
```

Schritt 2: Überprüfen von XL710

Wenn die Netzwerkkarte Intel XL710 ist, wird der Ethernet-Controller XL710 in der Ausgabe angezeigt.

Schritt 3: Überprüfen von E810CQDA2

Wenn die Netzwerkkarte Intel E810CQDA2 ist, sehen Sie die Ausgabe des Ethernet-Controllers E810-Cin.

Schritt 4: Bestätigen der Treiberunterstützung

Um den verwendeten Netzwerkkartentreiber zu überprüfen, führen Sie Folgendes aus:

```
ethtool -i
```

Ausgabe Beispiel für XL710:

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

Ausgabebeispiel für E810CQDA2:

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

Stellen Sie sicher, dass die Treiberversion mit der Kompatibilitätsmatrix für Ihre OpenStack- und Linux-Distribution übereinstimmt.

## 3. BIOS-/UEFI-Konfiguration

- SR-IOV aktivieren:

Stellen Sie sicher, dass SR-IOV auf den Servern BIOS/UEFI aktiviert ist.

- Virtualisierungstechnologie für Directed I/O (VT-d)/AMD-VI:

Intel VT-d oder AMD-Vi müssen für PCI-Passthrough und SR-IOV-Funktionalität aktiviert sein.

## 4. OpenStack-Setup

- Zentrale OpenStack-Services:

Stellen Sie sicher, dass OpenStack-Services wie Nova, Neutron, Glance und Keystone installiert und konfiguriert sind.

- Neutronenkonfiguration:

Neutron muss sowohl OpenSwitch (OVS) für Orchestrierungs-/Management-Netzwerke als auch SR-IOV für Anwendungs-/Service-Netzwerke unterstützen.

- SR-IOV-Konfiguration:

Die Rechenknoten müssen so konfiguriert werden, dass sie SR-IOV unterstützen, wobei auf den NICs virtuelle Funktionen (VFs) erstellt werden.

## 5. Cisco Prime Network Registrar (CPNR) VNF-Image

- VNF-Image-Kompatibilität:

Das CPNR-VNF-Image muss SR-IOV-Schnittstellen unterstützen und die erforderlichen Treiber enthalten.

- Auf Übersicht hochladen:

Stellen Sie sicher, dass das CPNR-VNF-Image auf OpenStack Glance verfügbar ist.

### 6. Verwaltungszugriff

- OpenStack-CLI:

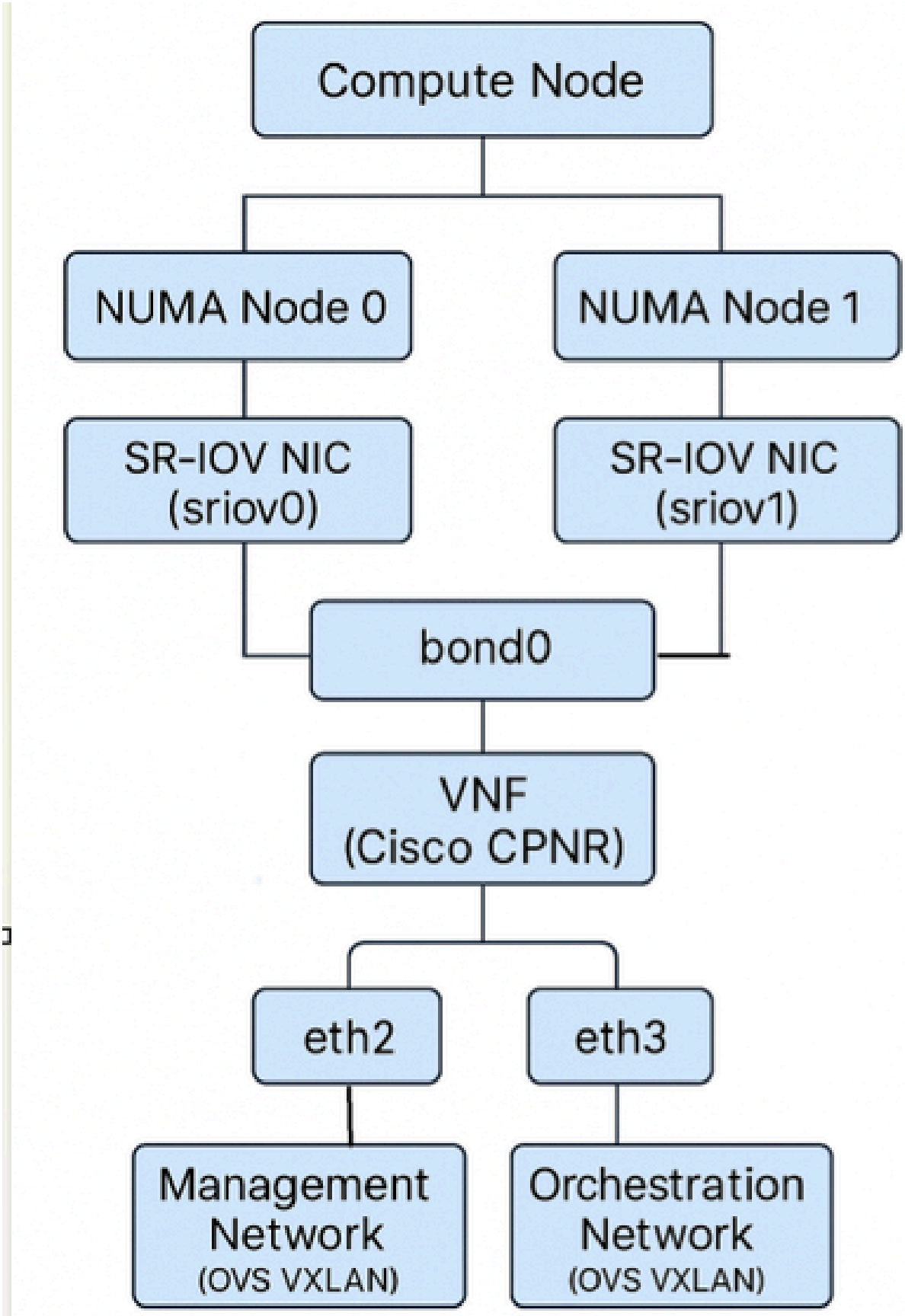
Stellen Sie den Zugriff auf die OpenStack-CLI sicher, um Netzwerke und Varianten zu erstellen und die VNF zu starten.

- Stamm- oder Administratorberechtigungen:

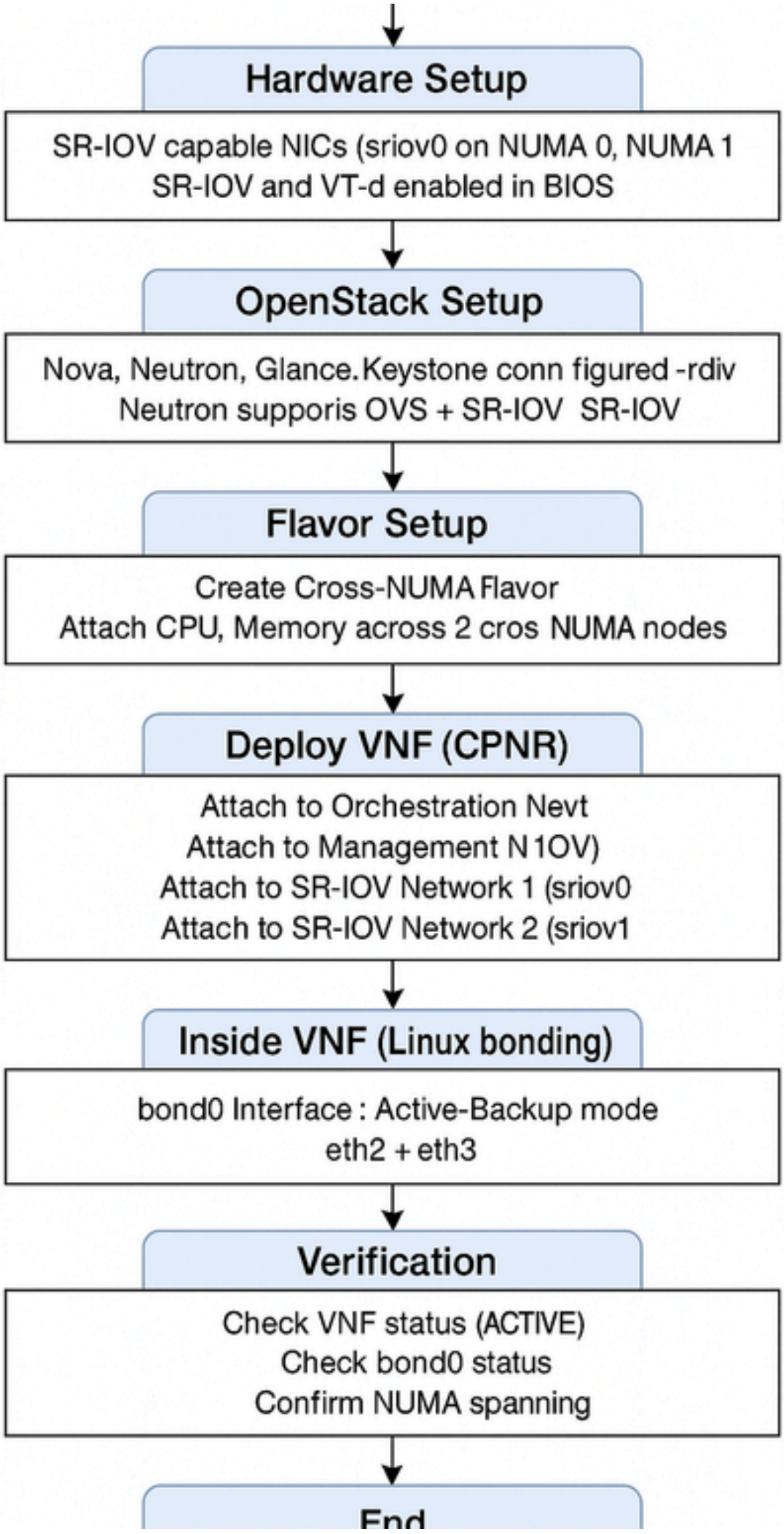
Root- oder Administratorzugriff zur Konfiguration des Netzwerks auf dem Linux-Host und innerhalb des VNF.

## Architekturübersicht

### VNF-Netzwerkschnittstellen-Verbindungsdiagramm




Flussdiagramm


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

2. Aktiv-Backup-Bonding:

    - Eine bond0-Schnittstelle wird mithilfe von SR-IOV NICs erstellt (eth2 von sriov0 und eth3 von sriov1).
    - Der Active-Backup-Modus stellt Redundanz und Fehlertoleranz sicher, ohne dass Switch-seitige Konfigurationen erforderlich sind.

3. OpenStack-Netzwerke:

    - Orchestrierungs- und Managementnetzwerke: OpenSwitch-basiert für Kontroll- und Verwaltungsdatenverkehr
    - Anwendungs-/Servicenetzwerke: SR-IOV-basiert für Hochleistungs-Datenverkehr

# Warum der NUMA-übergreifende Modus erforderlich ist

## 1. NUMA-Aware Networking in OpenStack

NUMA ist eine Speicherarchitektur, in der jede CPU (und ihr lokaler Speicher und ihre Geräte) in einem NUMA-Knoten gruppiert ist. Bei OpenStack wird durch die NUMA-sensitive Platzierung sichergestellt, dass VNFs optimal Ressourcen auf demselben NUMA-Knoten zugewiesen werden, um die Latenz zu minimieren und die Leistung zu maximieren.

- SR-IOV-NICs sind NUMA-lokal:
    - Jede physische NIC ist mit einem bestimmten NUMA-Knoten verknüpft. Beispiele:
        - sriov0 ist mit dem NUMA-Knoten 0 verbunden.
        - sriov1 ist mit NUMA-Knoten 1 verbunden.
- Einschränkung des Single-NUMA-Modus:
    - Wenn ein VNF im Single-NUMA-Modus gestartet wird, ermöglicht OpenStack dem VNF nur, sich mit NICs zu verbinden, die sich lokal im NUMA-Knoten befinden, von dem das VNF gestartet wird. Das bedeutet:
        - Wenn die VNF auf NUMA 0 gestartet wird, kann sie nur eine Verbindung zu den NICs auf sriov0 herstellen.
        - Wenn VNF auf NUMA 1 gestartet wird, kann es nur eine Verbindung zu NICs auf sriov1 herstellen.

## 2. Warum ist der NUMA-übergreifende Modus erforderlich?

CPNR VNF benötigt Zugang zu:

- Orchestrierungsnetzwerk(OpenSwitch, NUMA-unabhängig)
- Managementnetzwerk(OpenSwitch, NUMA-unabhängig)
- SR-IOV-Netzwerk 1: Verbunden mit sriov0(NUMA-Knoten 0)
- SR-IOV-Netzwerk 2: Verbunden mit sriov1 (NUMA-Knoten 1).

Bei dieser Bereitstellung benötigt der CPNR-VNF Zugriff auf SR-IOV NICs von NUMA 0 (sriov0) und NUMA 1 (sriov1), um Redundanz und hohe Verfügbarkeit zu gewährleisten. Um dies zu erreichen:

- Die VNF muss im NUMA-übergreifenden Modus gestartet werden, sodass OpenStack CPU, Speicher und NICs von mehreren NUMA-Knoten zuweisen kann.
- Dadurch wird sichergestellt, dass die VNF eine Verbindung zu den NICs auf sriov0 und sriov1 herstellen kann, sodass beide SR-IOV-Ports in einer Active-Backup-Bond-Konfiguration verwendet werden können.

# Größenbeschränkung für OVS-Ports aufheben

## Was ist Contrack?

Contrack ist eine Linux-Kernelfunktion, die zum Verfolgen von Netzwerkverbindungen verwendet wird, insbesondere für Network Address Translation (NAT) und Firewall-Regeln. Für OVS-basierte Ports in OpenStack wird contrack verwendet, um den Verbindungsstatus zu verwalten und Sicherheitsgruppenregeln durchzusetzen.

## Auswirkungen von Contratrack auf OVS-Ports

1. ConnTrack-Tabelle:

    - Für jede aktive Verbindung wird ein Eintrag in der conntrack-Tabelle verwendet.
    - Die Größe der conntrack-Tabelle wird durch den Parameter thenf_conntrack_maxbegrenzt.

2. Standardlimit:

    - Standardmäßig beträgt die Größe der conntrack-Tabelle 65536 Einträge. Bei Workloads mit hohen Verbindungsraten (z. B. VNFs mit vielen gleichzeitigen Datenflüssen) kann dieser Grenzwert schnell überschritten werden, sodass Pakete verworfen werden.

3. Auswirkungen auf OVS-Ports:

    - Wenn die conntrack-Tabelle voll ist, werden neue Verbindungen getrennt, was sich erheblich auf die VNF-Leistung auswirken kann.
    - Dies gilt insbesondere für die Orchestrierungs- undManagement-Netzwerke, die OVS-Ports verwenden.

## Abwehr von Constracking-Einschränkungen

1. Größe der Kontrolltabelle erhöhen:

- Aktuelles Limit anzeigen:

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

- Erhöhen Sie das Limit:

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

- Dauerhafte Änderung:

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

2. Überwachung der ConnTrack-Nutzung:

   Kontrahierungsstatistiken überprüfen:

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

3. Sicherheitsgruppenregeln optimieren:

   Reduzieren Sie die Anzahl der Regeln, die auf OVS-Ports angewendet werden, um den Overhead für die Zugriffe zu minimieren.

# So behebt SR-IOV Verbindungsprobleme

## 1. Beseitigt ConnTrack-Abhängigkeit

SR-IOV-Ports umgehen den OVS-Datenpfad und Linux-Kernel-Funktionen wie conntrack. Dadurch wird der Aufwand für die Verbindungsnachverfolgung vollständig beseitigt.

## 2. Höhere Skalierbarkeit

Im Gegensatz zu OVS-Ports, die durch die Größe der conntrack-Tabelle (nf_conntrack_max) begrenzt sind, können SR-IOV-Ports eine nahezu unbegrenzte Anzahl von Verbindungen verarbeiten.

## 3. Reduzierte Latenz

Durch die Auslagerung der Paketverarbeitung an die NIC-Hardware eliminieren die SR-IOV-Ports die Latenz, die durch die softwarebasierte Parallelverarbeitung entsteht.

# Gründe für die Wahl des Aktiv-Backup-Modus für SR-IOV-Ports auf CPNR VM

Der Active-Backup-Bonding-Modus eignet sich aufgrund seiner Einfachheit, Fehlertoleranz und Kompatibilität mit SR-IOV-Schnittstellen besonders gut für diese Bereitstellung. Hier die Gründe:

## 1. Redundanz ohne Komplexität

- Aktiv-Backup-Modus: Es wird zu einem bestimmten Zeitpunkt nur über eine Schnittstelle (die aktive Schnittstelle) Datenverkehr übertragen und empfangen. Die anderen Schnittstellen verbleiben im Standby-Modus.
- Wenn die aktive Schnittstelle ausfällt (z. B. aufgrund eines Verbindungsausfalls oder eines Hardwareproblems), schaltet die Bindung automatisch auf eine Standby-Schnittstelle um. So wird eine kontinuierliche Netzwerkkonnektivität ohne manuelle Eingriffe sichergestellt.

## 2. Keine Link Aggregation Group (LAG) erforderlich

- Im Gegensatz zu anderen Bonding-Modi (z. B. 802.3adorbalance-alb) ist für den Active-Backup-Modus kein Link Aggregation Control Protocol (LACP) oder Switch-seitige Konfigurationen erforderlich.
- Dies ist besonders für SR-IOV-Ports wichtig, da SR-IOV VFs in der Regel keine LACP- oder LAG-Konfigurationen unterstützen.

## 3. Nahtloses Failover

- Failover erfolgt zeitnah mit minimaler Unterbrechung des Datenverkehrs.
- Wenn die aktive Schnittstelle ausfällt, wird die Standby-Schnittstelle durch die Bindung sofort in den aktiven Status versetzt.

## 4. Hardware-Unabhängigkeit

Der Active-Backup-Modus funktioniert unabhängig von den zugrunde liegenden physischen Switches oder der Hardware. Die Failover-Logik befindet sich vollständig im Linux-Kernel und ist daher äußerst mobil und vielseitig.

## 5. Optimiert für SR-IOV

SR-IOV-VFs sind mit bestimmten physischen NICs und NUMA-Knoten verknüpft. Mit dem Active-Backup-Modus können Sie VFs von verschiedenen NUMA-Knoten zu einer einzigen logischen Bindungsschnittstelle (bond0) kombinieren. So wird eine hohe Verfügbarkeit sichergestellt und gleichzeitig die NUMA-Ressourcen effizient genutzt.

Der Active-Backup-Modus ist einer der einfachsten und am weitesten verbreiteten Modi beim

Linux-Bonding. Es wurde für hohe Verfügbarkeit entwickelt, indem sichergestellt wird, dass der Datenverkehr auch dann nahtlos weiterfließt, wenn eine der verbundenen Schnittstellen ausfällt. Dies ist eine ausführliche Erläuterung der Funktionsweise des Active-Backup-Modus, seiner Hauptmerkmale und Vorteile.

# Was ist eine Linux Bond-Schnittstelle?

Eine Bond-Schnittstelle in Linux kombiniert zwei oder mehr Netzwerkschnittstellen zu einer einzigen logischen Schnittstelle. Diese logische Schnittstelle, die als Bindung (z. B. bond0) bezeichnet wird, dient zur Bereitstellung von:

- Redundancy: Gewährleistung einer hohen Verfügbarkeit der Netzwerkverbindungen.
- Leistungssteigerung: In anderen Modi (z. B. balance-error802.3ad) kann es auch Bandbreite aggregieren.

# Wie funktioniert der Active-Backup-Modus?

Im Aktiv-Backup-Modus wird jeweils nur eine Schnittstelle (die aktive Schnittstelle) zum Senden und Empfangen von Datenverkehr verwendet. Die anderen Schnittstellen verbleiben im Standby-Modus. Wenn die aktive Schnittstelle ausfällt, wird eine der Standby-Schnittstellen in den aktiven Status hochgestuft, und der Datenverkehr wird automatisch auf die neue aktive Schnittstelle umgeleitet.

## Wichtigste Funktionen des Aktiv-Backup-Modus

1. Eine aktive Schnittstelle:

    - Zu jedem Zeitpunkt ist nur eine physische Schnittstelle der Verbindung zum Senden und Empfangen von Datenverkehr aktiv.
    - Standby-Schnittstellen sind vollständig passiv, es sei denn, es tritt ein Failover auf.

2. Automatisches Failover:

    - Wenn die aktive Schnittstelle ausfällt (z. B. aufgrund eines Hardwareproblems, Kabeltrennung oder Verbindungsausfalls), schaltet die Verbindung automatisch auf eine Standby-Schnittstelle um.
    - Failover ist nahtlos und erfordert keinen manuellen Eingriff.

3. Failback-Unterstützung:

    Sobald die ausgefallene Schnittstelle wiederhergestellt ist, kann sie je nach Bonding-Konfiguration automatisch wieder aktiviert werden (sofern konfiguriert) oder im Standby-Modus verbleiben.

4. Keine Switch-seitigen Anforderungen:

    - Im Gegensatz zu anderen Bonding-Modi (z. B. 802.3adorbalance-rr) ist für den Active-Backup-Modus keine spezielle Konfiguration der physischen Switches (z. B. LAG oder LACP) erforderlich.
    - Damit eignet sie sich ideal für Szenarien, in denen eine Switch-seitige Konfiguration nicht möglich ist oder virtuelle SR-IOV-Funktionen verbunden werden, die die LAG nicht unterstützen.

5. Überwachung:

    - Der Bond überwacht kontinuierlich den Zustand aller Member-Schnittstellen mit dem Parameter `miimon`(Media Independent Interface Monitor).
    - Wenn ein Verbindungsausfall erkannt wird, schaltet die Verbindung sofort auf eine intakte Standby-Schnittstelle um.

# Datenverkehrsfluss im Aktiv-Backup-Modus

## Normalbetrieb

1. Aktive Schnittstelle:

    - Der Datenverkehr fließt ausschließlich über die aktive Schnittstelle (z. B. eth2in einer Bindung ofeth2andeth3).
    - Die Standby-Schnittstelle (eth3) bleibt inaktiv und sendet oder empfängt keinen Datenverkehr.

2. Überwachung:

    - Die Bindung überwacht regelmäßig den Status aller Teilnehmer-Schnittstellen. Dies geschieht mithilfe von:
        - Mimon: Überprüft den Verbindungsstatus jeder Schnittstelle in einem konfigurierbaren Intervall (z. B. alle 100 ms).
        - Address Resolution Protocol (ARP)-Überwachung (optional): Sendet ARP-Anfragen, um sicherzustellen, dass die aktive Schnittstelle erreichbar ist.

## Failover-Szenario

1. Verbindungsfehler auf aktiver Schnittstelle:

    Wenn die aktive Schnittstelle (eth2) ausfällt (z. B. Kabel abgezogen, NIC-Hardwarefehler oder Verbindungsunterbrechung), erkennt die Verbindung den Fehler sofort mithilfe der Mini- oder ARP-Überwachung.

2. Automatisches Failover:

- Der Bond schaltet auf die Standby-Schnittstelle (eth3) um, die zur neuen aktiven Schnittstelle wird.
- Der Datenverkehr wird ohne manuellen Eingriff über die neue aktive Schnittstelle umgeleitet.

3. Failover-Aktualität:

   Der Failover-Prozess läuft nahezu unmittelbar ab (in der Regel innerhalb weniger Millisekunden, je nach Minimumintervall).

### Failback-Szenario

1. Wiederherstellung einer fehlerhaften Schnittstelle:

   - Wenn die zuvor ausgefallene Schnittstelle (eth2) wiederhergestellt wird, kann sie:
     - Die aktive Rolle wird automatisch wieder beansprucht (sofern konfiguriert).
     - Verbleiben im Standby-Modus (Standardverhalten)

2. Kontinuität des Datenverkehrs:

   Failback ist nahtlos und stellt keine Unterbrechung des laufenden Datenverkehrs sicher.

## Anwendungsfall: Active-Backup-Bonding mit SR-IOV-Ports

Der Active-Backup-Modus eignet sich aus folgenden Gründen besonders für SR-IOV-Schnittstellen:

- SR-IOV-VFs unterstützen in der Regel keine Link-Aggregation-Protokolle wie LACP.
- Die Verbindung im Aktiv-Backup-Modus kann ohne Switch-seitige Konfiguration für Redundanz sorgen.

Beispiele:

- eth2 ist einem SR-IOV VF onsriov0 (NUMA-Knoten 0) zugeordnet.
- eth3 wird einem SR-IOV VF onsriov1 (NUMA-Knoten 1) zugeordnet.
- Die Bindung (bond0) kombiniert diese Schnittstellen und sorgt für ein nahtloses Failover zwischen SR-IOV-VFs.

## Schritt 1: OpenStack-Netzwerke

Für die CPNR-VNF sind die folgenden vier Netzwerke erforderlich:

1. Orchestrierungs-Netzwerk: Für Kontroll- und Orchestrierungsdatenverkehr (OpenSwitch-basiert).
2. Managementnetzwerk: Für Administratorzugriff (OpenSwitch-basiert).
3. SR-IOV-Netzwerk 1: Anwendungs-/Dienstdatenverkehr auf sriov0.
4. SR-IOV-Netzwerk 2: Anwendungs-/Service-Datenverkehr auf sriov1.

Schrittweise Bereitstellung:

## Schritt 1.1. OpenSwitch-Netzwerke erstellen

- Orchestrierungs-Netzwerk:

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

- Managementnetzwerk:

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

## Schritt 1.2: Erstellen von Subnetzen für OpenSwitch-Netzwerke

- Orchestrierungs-Subnetz:

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

- Management-Subnetz:

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

## Schritt 1.3. Erstellen von SR-IOV-Netzwerken

- SR-IOV-Netzwerk 1:

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

- SR-IOV-Netzwerk 2:

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

# Schritt 2: OpenStack-Varianten

## Schritt 2.1. Erstellen einer NUMA-übergreifenden Variante

Um sicherzustellen, dass die VNF auf SR-IOV NICs von beiden NUMA-Knoten zugreifen kann, erstellen Sie eine Variante mit NUMA-übergreifender Unterstützung:

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

## Schritt 2.2. Konfigurieren der NUMA-Eigenschaften

NUMA-spezifische Eigenschaften festlegen:

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

# Schritt 3: Bonding im Aktiv-Backup-Modus konfigurieren

Konfigurieren Sie nach dem Starten der VNF die Bond-Schnittstelle für SR-IOV-Ports (eth2 und eth3) auf der VNF.

## Schritt 3.1: Konfiguration der Anbindungsschnittstelle

Erstellen Sie eine Bond-Schnittstelle (bond0) im 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
```

## Schritt 3.2. Konfigurieren von Slave-Schnittstellen

- 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
  ```

## Schritt 3.3: Konfiguration anwenden

Starten Sie den Netzwerkdienst neu, um die Konfiguration anzuwenden:

```
systemctl restart network
```

# Überprüfung

Überprüfen Sie nach der Bereitstellung des VNF dessen Funktionalität mithilfe der folgenden Schritte:

## 1. VNF-Status überprüfen

Überprüfen Sie, ob die VNF-Instanz aktiv ist:

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

Stellen Sie sicher, dass der Status AKTIV lautet.

## 2. Netzwerkverbindung überprüfen

- Ping-Test: Vergewissern Sie sich, dass die VNF über alle Netzwerke kommunizieren kann:

  ```
  ping
  ```

  ```
  ping
  ```

- Bindungsschnittstelle:
  - Bestätigen Sie, dass bond0 aktiv ist:

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

    Suchen nach:

- Aktuell aktiver Slave: Zeigt die aktive Schnittstelle an.
- Slave-Schnittstelle: Bestätigt, dass sowohl eth2andeth3 Teil der Bindung sind.

### 3. NUMA-Platzierung überprüfen

Vergewissern Sie sich, dass die VNF Ressourcen von beiden NUMA-Knoten verwendet:

```
nova show

        --human | grep numa
```

## Best Practices

- Überwachung und Fehlerbehebung: Verwenden Sie Tools wie likepdumpandethtool, um die SR-IOV-Schnittstellen zu überwachen.
- Sicherheit: Sorgfältige Verwaltung des Zugriffs auf das physische Netzwerk und Durchsetzung strikter Isolierung zwischen Tenants
- Skalierung: Planen Sie die physische NIC-Kapazität bei der Skalierung von SR-IOV-Bereitstellungen, da die Anzahl der verfügbaren VFs durch die NIC-Hardware begrenzt ist.

## Fehlerbehebung

Wenn die Bereitstellung nicht wie erwartet funktioniert, führen Sie die folgenden Schritte zur Fehlerbehebung durch:

### 1. Überprüfen der SR-IOV-Konfiguration

- Überprüfen Sie, ob SR-IOV im BIOS aktiviert ist:

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

- Vergewissern Sie sich, dass VFs auf NICs erstellt werden:

```
lspci | grep Ethernet
```

## 2. NUMA-Platzierung überprüfen

Wenn das VNF nicht auf beide NICs zugreifen kann, stellen Sie sicher, dass der NUMA-übergreifende Modus aktiviert ist:

- Überprüfen Sie die NUMA-Eigenschaften der Variante:

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

## 3. Probleme mit Anleiheschnittstellen

- Überprüfen Sie den Anleihestatus:

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

- Wenn die Anleihe nicht funktioniert:
  - Stellen Sie sicher, dass die Slave-Schnittstellen (eth2andeth3) als Teil der Bindung richtig konfiguriert sind.
  - Starten Sie den Netzwerkdienst neu:

```
systemctl restart network
```

## 4. Probleme mit der Netzwerkverbindung

- Überprüfen Sie die OpenStack-Port-Bindungen:

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

- Prüfen Sie, ob die IP-Konfiguration in der VNF korrekt ist:

```
ip addr show
```

# Schlussfolgerung

Für die Bereitstellung von CPNR-VNF auf OpenStack mit SR-IOV-Ports ist der NUMA-übergreifende Modus erforderlich, damit die VNF eine Verbindung zu NICs beider NUMA-Knoten herstellen kann. Dies ist wichtig, da OpenStack VNFs im Single-NUMA-Modus auf den Zugriff auf Ressourcen (NICs, CPUs, Speicher) innerhalb des NUMA-Knotens beschränkt, in dem die VNF gestartet wird. Durch die Kombination des NUMA-übergreifenden Modus mit dem Active-Backup-Bonding werden eine hohe Verfügbarkeit, Fehlertoleranz und eine effiziente Ressourcennutzung sichergestellt, sodass diese Bereitstellung äußerst ausfallsicher und leistungsfähig ist.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.