

# SDNとNuage Networksの統合

ORACLE WHITE PAPER | 2019年4月



## 免責事項

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

## 改訂履歴

このホワイト・ペーパーには、初版発行以来、次の改訂が加えられています。

---

更新日	改訂内容
2019年4月17日	初版発行

---

Oracle Cloud Infrastructureホワイト・ペーパーの最新版は、  
<https://cloud.oracle.com/iaas/technical-resources>でご覧いただけます。

## 目次

概要	4
ソフトウェア要件	4
前提事項	4
ターゲット・シナリオ	5
技術アーキテクチャ	5
Nuage NetworksとのSDN統合をOracle Cloud Infrastructureにデプロイする	7
ネットワーク・インフラ（VCNとサブネット）を作成する	7
VSC用のインスタンスを作成する	9
Virtualized Services Controllerのインストールと構成	16
VSCをインストールする	17
VSCを構成する	22
Virtual Routing and Switchingのインストール	29
前提条件	30
VRSをインストールする	30
Nuage Networks SDNのテスト	36
付録A：Oracle Cloud InfrastructureにセカンダリNICをアタッチする	39
付録B：Virtualized Service ControllerのBOFファイル	42
付録C：Virtualized Service Controllerの構成ファイル	42
付録D：Virtual Routing and Switchingの構成ファイル	46
参考資料	51

## 概要

このホワイト・ペーパーは、Oracle Cloud Infrastructure内のNuage Networks from Nokiaソフトウェア定義ネットワークング（SDN）ソリューションに関する、詳細なデプロイ・ガイドです。内容としては、リファレンス・アーキテクチャやインストール手順のほか、ビルド時に実行されるテスト手順について説明します。

このホワイト・ペーパーは、Nuage Networks SDNソリューションを使用してオンプレミス・サービスをクラウドへとシームレスに拡張したいと考えている、ネットワーク・アーキテクトやネットワーク管理者を対象読者としています。

## ソフトウェア要件

このホワイト・ペーパーは、次のソフトウェア要件に基づいて記述されています。

- Nuage Networks Virtual Routing and Switching（VRS）およびVirtualized Services Controller（VSC）for KVM（リリース5.3.3）
- Oracle Linux 7.4以降
- CentOS 7

## 前提事項

このホワイト・ペーパーでは、次のことを前提としています。

- 読者がKVMについての知識を持ち、ハイパーバイザーとの連携方法について理解している。
- 読者がLinuxシステムの管理についての知識を持ち、ネットワーク・ファイルの設定方法と編集方法を理解している。
- 読者がNuage Networks SDNソリューションについて知識を持っている（Virtual Routing and Switching（VRS）、Virtualized Services Controller（VSC）、およびVirtualized Services Directory（VSD）を含む）。
- 読者が、オペレーティング・システムをゲストとしてインストールする方法や、仮想ディスク・イメージをディスク間でコピーする方法について理解している。
- 読者が、ゲストによるストレージの共有方法について理解している。
- 読者が、使用する環境に必要なリソースをすでに作成している（仮想クラウド・ネットワーク（VCN）やネットワーク関連情報など）。
- 読者が、ベア・メタル・コンピュータ・インスタンスのプロビジョニング方法を知っている。
- 読者のKVMホストがインターネットにアクセスできる。
- 読者が、KVM用のNuage Networks VRSおよびVSC qcow2イメージを保有している。読者は、この仮想マシン・イメージをqcow2形式でインポートすることになります。

## ターゲット・シナリオ

このホワイト・ペーパーで使用しているシナリオは、SDNオーバーレイを通じたデータ・センター拡張です。

## 技術アーキテクチャ

Nuage Networks Virtualized Services Platform (VSP) は、データ・センターとクラウド・ネットワークの仮想化を提供するSDNソリューションです。VSPは、作成されたコンピュート・リソース間の接続を提供します。

Nuage Networksは、Oracle Cloud Infrastructure内のベア・メタル・サーバーの上層で迅速なOpen vSwitch (OVS) 交換を使用することにより、Nuage Networks Virtual Routing and Switching (VRS) コントローラに接続します。VRSがコントロール・プレーン (VSC) に接続され、そのコントロール・プレーンとデータ・プレーン (インフラ) の接続がIPSec VPNトンネル経由で確立されたら、その他のことはクラウドベースのソリューション内で定義されます。次の図は、アーキテクチャとインフラの要件を示したものです。

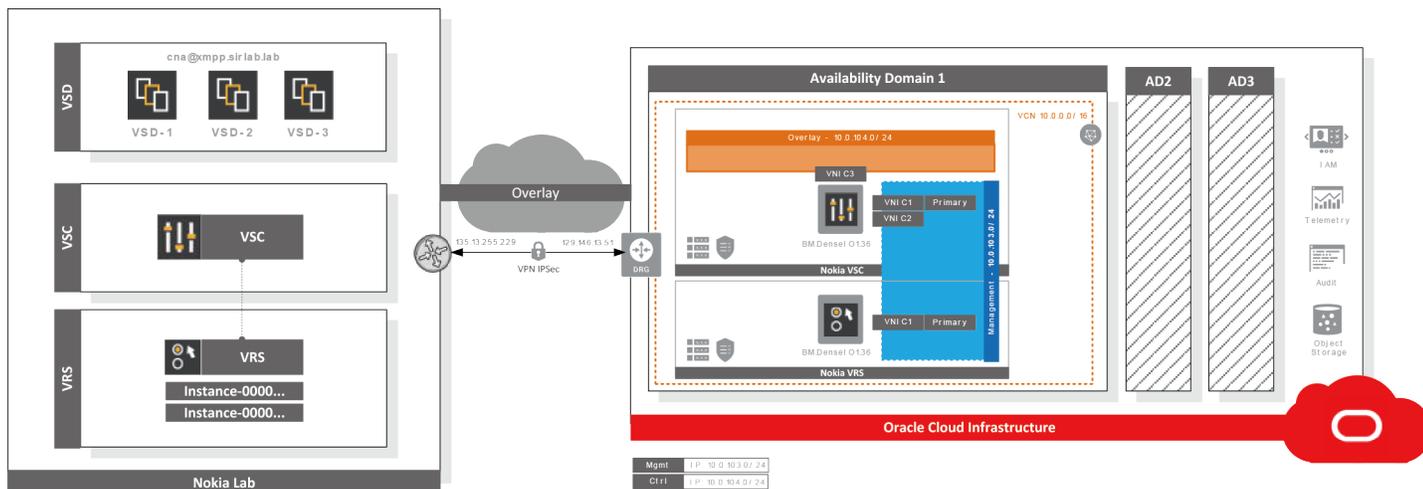


図1 : Nuage Networks SDNのアーキテクチャ

Nuage Networks VSPのソフトウェア・スイートには、次の主要製品が含まれます。

- **Virtualized Services Directory (VSD)** : ネットワーク・サービスの抽象定義をサポートする、ポリシー、ビジネス・ロジック、およびアナリティクス・エンジンです。管理者は、VSDに対するRESTful APIを使用して、サービス設計を定義し、エンタープライズ・ポリシーを実装することができます。
- **Virtualized Services Controller (VSC)** : データ・センター・ネットワーク用のコントロール・プレーンです。VSCは、ネットワークおよびサービス・トポロジの完全なテナント単位ビューを保持します。OpenFlowなどのネットワークAPIを使用することで、VSCはデータ・センターのネットワーキング・ハードウェアから独立したデータ・センター・ネットワークをプログラムします。

- **Virtual Routing and Switching (VRS)** : ネットワーク・サービスの仮想エンドポイントです。VRSは、コンピュート環境で変更が生じた際にそれを検知し、ポリシーベースの応答をトリガーすることで、アプリケーションに必要なネットワーク接続が確保されるようにします。

次の図は、このソリューションの各コンポーネントを説明したものです。

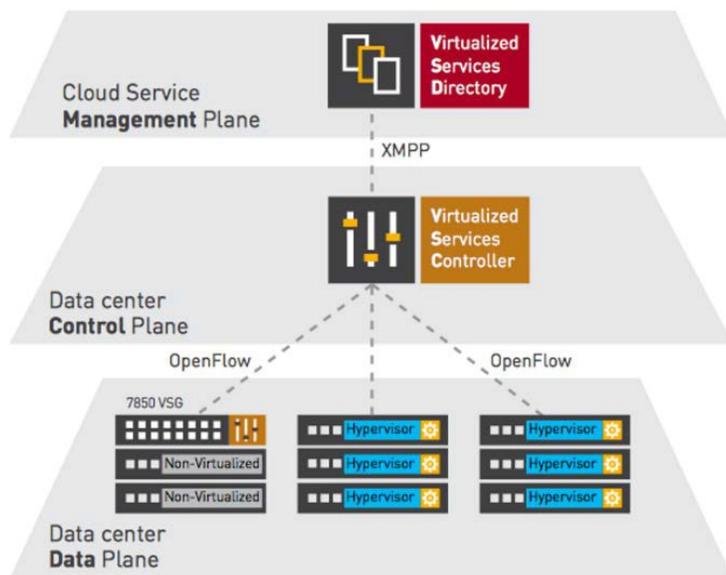


図2 : Nuage Networks VSP

このホワイト・ペーパーでは、Oracle Cloud InfrastructureにVSCとVRSの両方をデプロイして構成し、お客様のネットワークに接続します。マネジメント・プレーン（VSD）については、お客様の施設内に維持されるため、このホワイト・ペーパーでは説明しません。

次の図は、Nuage NetworksのVSCとVRSをOracle Cloud Infrastructureにデプロイするための手順を示したものです。このプロセスは、Terraformを通じて自動化できます。それでは始めていきましょう。

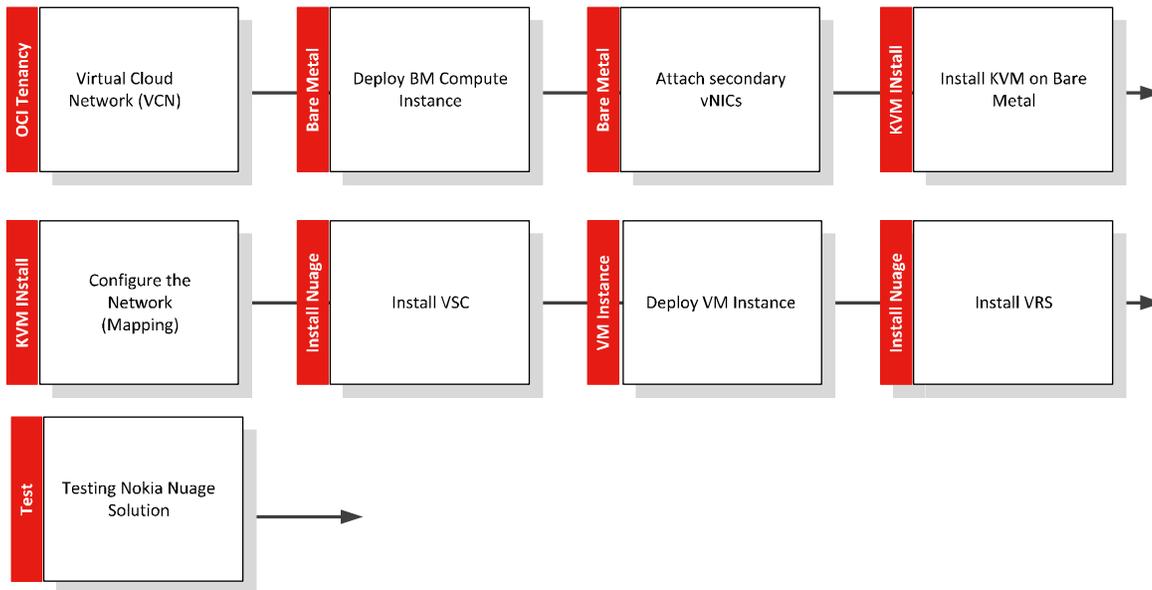


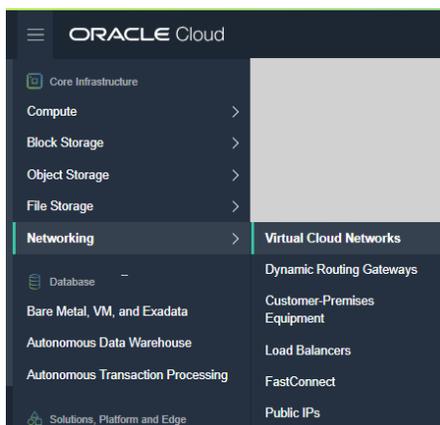
図3：デプロイの大まかな流れ

## Nuage NetworksとのSDN統合をOracle Cloud Infrastructureにデプロイする

Nuage NetworksとのSDN統合をOracle Cloud Infrastructureにデプロイするには、次のタスクを実行します。

### ネットワーク・インフラ（VCNとサブネット）を作成する

1. Oracle Cloud Infrastructureコンソールにログインします。
2. ナビゲーション・メニューで、「ネットワーキング」を選択し、「仮想クラウド・ネットワーク」を選択します。



3. 「仮想クラウド・ネットワークの作成」をクリックします。
4. 次の図に示すように、2つのパブリック・サブネットを持つ新しいVCNを作成します。

### Subnets in SDN Compartment

Create Subnet

Sort by: Created Date (Desc) Displaying 2 Subnets < Page 1 >

	<b>mgt-plain</b> OCID: ...czh4da <a href="#">Show</a> <a href="#">Copy</a>	<b>CIDR Block:</b> 10.0.103.0/24 <b>Virtual Router MAC Address:</b> 00:00:17:0D:62:FF	<b>Subnet Type:</b> Availability Domain-Specific <b>Availability Domain:</b> shPn:EU-FRANKFURT-1-AD-1 <b>DNS Domain Name:</b> mgtplain... <a href="#">Show</a> <a href="#">Copy</a> <b>Subnet Access:</b> Public Subnet	<b>Route Table:</b> <a href="#">Default Route Table for VCN</a> <b>Security Lists:</b> <a href="#">Default Security List for VCN</a>	<b>DHCP Options:</b> <a href="#">Default DHCP Options for VCN</a>
	<b>ctl-plain</b> OCID: ...mjw6q <a href="#">Show</a> <a href="#">Copy</a>	<b>CIDR Block:</b> 10.0.104.0/24 <b>Virtual Router MAC Address:</b> 00:00:17:0D:62:FF	<b>Subnet Type:</b> Availability Domain-Specific <b>Availability Domain:</b> shPn:EU-FRANKFURT-1-AD-1 <b>DNS Domain Name:</b> ctlplain... <a href="#">Show</a> <a href="#">Copy</a> <b>Subnet Access:</b> Public Subnet	<b>Route Table:</b> <a href="#">Default Route Table for VCN</a> <b>Security Lists:</b> <a href="#">Default Security List for VCN</a>	<b>DHCP Options:</b> <a href="#">Default DHCP Options for VCN</a>

この例では、一方のサブネットを**ctl-plain**という名前にし、CIDRブロック10.0.104.0/24を使用して、AD-1内に作成します。もう一方のサブネットは、**mgt-plain**という名前にし、CIDRブロック10.0.103.0/24を使用して、AD-1内に作成します。どちらのサブネットについても、デフォルトのルート表、セキュリティ・リスト、およびDHCPオプションが使用されます。

次の図は、ルート表の詳しい構成を示したものです。なお、この例では、VPNを使用してVCNをオンプレミス・ネットワークに接続するために、動的ルーティング・ゲートウェイ（DRG）も作成しました。CIDR 10.5.0.0/16は、VCNをVPN経由でオンプレミス環境に接続するために使用されたサブネットです。

### Default Route Table for VCN

Apply Tag(s)

Route Table Information [Tags](#)

OCID: ...igmfiq [Show](#) [Copy](#)      Compartment: (root)/SDN  
 Created: Mon, 11 Mar 2019 09:11:06 GMT

### Route Rules

Displaying 2 Route Rules

[Edit Route Rules](#)

Destination CIDR Block: 10.5.0.0/16      Target Type: Dynamic Routing Gateway  
 Target: DRG ...kjrakq [Show](#) [Copy](#)

次の図は、各セキュリティ・リストの詳しい構成を示したものです。VPNの作成前にインターネットからインスタンスへアクセスできるようにするために、ポート22は開けておく必要があります。また、すべてのプロトコルでオンプレミス環境からのトラフィックが許可されるようにするために、サブネット10.5.0.0/16を開けておく必要があります。

### Default Security List for VCN

Edit All Rules Terminate Apply Tag(s)

Security List Information Tags

OCID: ...rait3q [Show](#) [Copy](#)  
Created: Mon, 11 Mar 2019 09:11:06 GMT

*Instance traffic is controlled by firewall rules on each Instance in addition to this Security List*

### Ingress Rules

Stateless Rules

No Ingress Rules

There are no stateless Ingress Rules for this Security List.

Stateful Rules

Source	IP Protocol	Source Port Range	Destination Port Range	Allows
0.0.0.0/0	TCP	All	22	TCP traffic for ports: 22 SSH Remote Login Protocol
0.0.0.0/0	ICMP	Type and Code: 3, 4		ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
10.0.0.0/16	ICMP	Type and Code: 3		ICMP traffic for: 3 Destination Unreachable
10.5.0.0/16	TCP	All	All	TCP traffic for ports: all

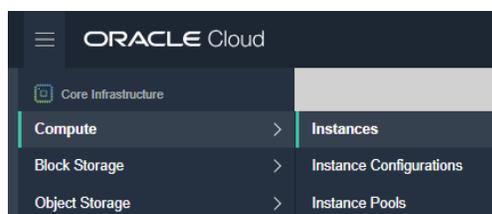
## VSC用のインスタンスを作成する

Virtualized Services Controller (VSC) 用のインスタンスを作成するには、次のタスクを実行します。

### Oracle Cloud Infrastructureにインスタンスをデプロイする

コンピュータ・インスタンスをデプロイするには、VCNとサブネットがすでにデプロイされている必要があります。

1. コンソールのナビゲーション・メニューから、「コンピュータ」を選択し、「インスタンス」を選択します。



2. 「インスタンスの作成」をクリックします。
3. インスタンスの名前（たとえば、**Instance-VSC**）を指定し、可用性ドメイン（**AD 1**）を選択した後、「イメージ・ソースの変更」を選択し、「**CentOS 7**」を選択します。

Create Compute Instance

Oracle Cloud Infrastructure Compute lets you provision and manage compute hosts, known as instances. You can launch instances as needed to meet your compute and application requirements.

Name your instance

Instance-VSC

Select an availability domain for your instance

AD 1

shPn:EU-FRANKFURT-1-AD-1 ✓

AD 2

shPn:EU-FRANKFURT-1-AD-2

AD 3

shPn:EU-FRANKFURT-1-AD-3

Choose an operating system or image source



CentOS 7  
Image Build: 2019.02.23-0

CentOS is a free, open-source Linux distribution that is suitable for use in enterprise cloud environments. For more information, see <https://www.centos.org>.

Change Image Source

4. 「ベア・メタル・マシン」を選択し、「シェイプの変更」をクリックして、「**BM.Standard1.36**」を選択した後、「シェイプを選択」をクリックします。

Browse All Shapes Close

A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance. See [Compute Shapes](#) for more information.

Shape Name	OCPU	Memory (GB)	Local Disk (TB)	Network Bandwidth	Max. Total VNICs
<input type="checkbox"/> BM.Standard2.52	52	768	Block storage only	2 x 25 Gbps	24 total (12 per physical NIC)
<input type="checkbox"/> BM.DenseIO2.52	52	768	51.2TB NVMe SSD	2 x 25 Gbps	24 total (12 per physical NIC)
<input checked="" type="checkbox"/> BM.Standard1.36	36	256	Block storage only	10 Gbps	16
<input type="checkbox"/> BM.DenseIO1.36	36	512	28.8TB NVMe SSD	10 Gbps	16

1 Selected Showing 4 Item(s)

Select Shape Cancel

5. 「SSHキーの追加」セクションで、SSH鍵ファイルを選択するか、テキスト・ボックスにSSH鍵を貼り付けます。

6. 「**ネットワーキングの構成**」セクションで、VCNのコンパートメント、VCN、サブネットのコンパートメント、およびサブネット (**mgt-plain**) を選択します。

Create Compute Instance

Configure networking

Virtual cloud network compartment  
SDN

Virtual cloud network  
VCN

Subnet compartment  
SDN

Subnet ⓘ  
mgt-plain

7. 「**作成**」をクリックします。

数分後、インスタンスが（次の画像のように）起動します。

Instance-VSC

Create Custom Image Start Stop Reboot Terminate Apply Tag(s)

Instance Information Tags

**Instance Information**

**Availability Domain:** shPn:EU-FRANKFURT-1-AD-1

**Fault Domain:** FAULT-DOMAIN-1

**Region:** eu-frankfurt-1

**Shape:** BM.Standard1.36

**Virtual Cloud Network:** [VCN](#)

**Maintenance Reboot:** -

## KVMソフトウェアをLinux (CentOS) にインストールする

1. PuTTYやMobaXtermなどのソフトウェアを使用して、インスタンスのSSH接続にログインします。

2. `/etc/default/grub` ファイルを編集し、次の行を（下の図のように）追加します：

```
intel_iommu=on.  
GRUB_CMDLINE_LINUX="crashkernel=auto LANG=en_US.UTF-8  
transparent_hugepage=never console=tty0 console=ttyS0,9600  
libiscsi.debug_libiscsi_ah=1 rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0  
ip=dhcp netroot=iscsi:169.254.0.2:::ign.2015-02.oracle.boot:uefi  
iscsi_param=node.session.timeo.replacement_timeout=6000  
net.ifnames=1 intel_iommu=on"
```

3. 次のコマンドを実行してKVMソフトウェアをインストールし、`libvirtd`サービスを起動して有効にします。

```
# sudo su -  
# yum install qemu-kvm qemu-img virt-manager libvirt libvirt-python  
libvirt-client virt-install virt-viewer bridge-utils  
# systemctl start libvirtd  
# systemctl enable
```

## ネットワークを準備する

1. 次のコマンドを実行して、2つのネットワーク・コントローラが接続されていることを確認します。

```
# sudo lspci | egrep -i --color 'network|ethernet'
```

```
[opc@instance-vsc ~]$ sudo lspci | egrep -i --color 'network|ethernet'  
03:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)  
03:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
```

2. 次のコマンドを実行して、アタッチされている物理NICが`ensf0`だけであることを確認します。

```
# sudo ip link show | grep ens
```

```
[opc@instance-vsc ~]$ sudo ip link show | grep ens  
2: ens3f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP mode DEFAULT group default qlen 1000  
3: ens3f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN mode DEFAULT group default qlen 1000
```

3. 次のコマンドを実行して、KVM用のネットワークを初期化するためのスクリプトを作成します。

```
# sudo su -  
# vi /usr/bin/initialize-kvm-network.sh
```

4. スクリプトに次のテキストを追加します。

```
#!/bin/bash  
  
function build_sriov_vf  
{  
    number_vfs=2  
    vnic_json=`curl -s http://169.254.169.254/opc/v1/vnics/`
```

```

vnic_count=`echo ${vnic_json} | jq -r 'length'`
count=0

for field in macAddr vlanTag
do
    read -ra ${field} <<< `echo ${vnic_json} | jq -r '.[0:length] | .["${field}"]'`
done
while [ ${count} -lt ${vnic_count} ]
do
    if [ ${vlanTag[${count}]} -eq 0 ]
    then
        physdev=`ip -o link show | grep ${macAddr[${count}]} | awk -F: '{gsub(/\s+/, ""), $2}; print $2'`
        echo ${number_vfs} > /sys/class/net/${physdev}/device/sriov_numvfs
        wait
        bridge link set dev ${physdev} hwmode vepa
    fi

    if [ ${vlanTag[${count}]} -gt 0 ]
    then
        (( vf_index = count - 1 ))
        ip link set ${physdev} vf ${vf_index} mac ${macAddr[${count}]} spoofchk off
    fi

    (( count = count + 1 ))
done
}

build_sriov_vf

#wait 30s to OS enable VFs
sleep 30s

```

5. 次のコマンドを実行して、ファイルに対する権限を変更し、ファイルを実行できるようにします。

```
# chmod +x /usr/bin/initialize-kvm-network.sh
```

6. 次のコマンドを実行してスクリプトを実行し、仮想機能デバイスを有効にします。

```
# /usr/bin/initialize-kvm-network.sh
```

7. 次のコマンドを実行して、作成された仮想デバイスを表示します。

```
# lshw -c network -businfo
```

```

[root@instance-vsc ~]# lshw -c network -businfo
Bus info          Device          Class           Description
=====
pci@0000:03:00.0  ens3f0         network        82599ES 10-Gigabit SFI/SFP+ Network Connection
pci@0000:03:00.1  ens3f1         network        82599ES 10-Gigabit SFI/SFP+ Network Connection
pci@0000:03:10.0  enp3s16       network        82599 Ethernet Controller Virtual Function
pci@0000:03:10.2  enp3s16f2     network        82599 Ethernet Controller Virtual Function

```

出力を見ると、追加された仮想機能デバイスは、enp3s16とenp3s16f2であることが確認できます。

8. 次のコマンドを実行して、これらの仮想機能のMACアドレスを表示します。

```
# ip -o link show | grep enp
```

```
[root@instance-vsc ~]# ip -o link show | grep enp
10: enp3s16: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000\ link/ether 02:00:17:01:c1:d5 brd ff:ff:ff:ff:ff:ff
11: enp3s16f2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000\ link/ether 02:00:17:01:9b:e2 brd ff:ff:ff:ff:ff:ff
```

この例の場合、MACアドレスは次のようになります。

仮想機能	MACアドレス
enp3s16	02:00:17:01:c1:d5
enp3s16f2	02:00:17:01:9b:e2

9. 次の情報を使用して、各仮想機能の新しい構成ファイルを作成します。

ファイル	詳細
/etc/sysconfig/network-scripts/ifcfg-enp3s16	DEVICE= <b>enp3s16</b> BOOTPROTO=none ONBOOT=yes MACADDR=" <b>02:00:17:01:c1:d5</b> " NM_CONTROLLED=no MTU=9000
/etc/sysconfig/network-scripts/ifcfg-enp3s16f2	DEVICE= <b>enp3s16f2</b> BOOTPROTO=none ONBOOT=yes MACADDR=" <b>02:00:17:01:9b:e2</b> " NM_CONTROLLED=no MTU=9000

10. 各ファイルを作成するには、次のコマンドを実行して、各ファイルにコンテンツを含めます。

```
# vi /etc/sysconfig/network-scripts/ifcfg-enp3s16
```

```
# vi /etc/sysconfig/network-scripts/ifcfg-enp3s16f2
```

11. 次の情報を使用して、各仮想機能デバイスのVLAN構成ファイルを作成します。

ファイル	詳細
/etc/sysconfig/network-scripts/ifcfg-enp3s16.vlan1	DEVICE=vlan1 PHYSDEV=enp3s16 BOOTPROTO=none ONBOOT=yes NM_CONTROLLED=no VLAN=yes
/etc/sysconfig/network-scripts/ifcfg-enp3s16f2.vlan2	DEVICE=vlan2 PHYSDEV=enp3s16f2 BOOTPROTO=none ONBOOT=yes NM_CONTROLLED=no VLAN=yes

12. 各ファイルを作成するには、次のコマンドを実行して、各ファイルにコンテンツを含めます。

```
# vi /etc/sysconfig/network-scripts/ifcfg-enp3s16.vlan1
```

```
# vi /etc/sysconfig/network-scripts/ifcfg-enp3s16f2.vlan2
```

13. /usr/bin/initialize-kvm-network.shファイルに、次のエントリを追加します。

```
ifup enp3s16
ifup
enp3s16f2
ifup vlan1
```

14. 手順を完了してKVMネットワークをサービスとして確立するには、次のコマンドを実行します。

```
# systemctl daemon-reload
# systemctl enable kvm-network.service
# systemctl start kvm-network.service
```

## Virtualized Services Controllerのインストールと構成

Virtualized Services Controller (VSC) は、Nuage Networks VSPソリューションのコントロール・プレーンです。VSCはハイパーバイザと通信して、仮想マシン (VM) 情報を収集します (MACアドレスやIPアドレスなど)。

VSCは、TCPポート6333で実行されるOpenFlowを使用して、Virtual Routing and Switching (VRS) モジュールを制御します。VSCがXMPPプロトコルを通じてVirtualized Services Directory (VSD) と通信することにより、VM用の新しいポリシーや、ポリシーの更新がダウンロードできるようになります。VSC間の通信は、マルチプロトコル・ボーダー・ゲートウェイ・プロトコル (MP-BGP) を通じて実行されます。これは、VSC間におけるVMのMAC/IP到達可能性情報を配信するために使用されます。

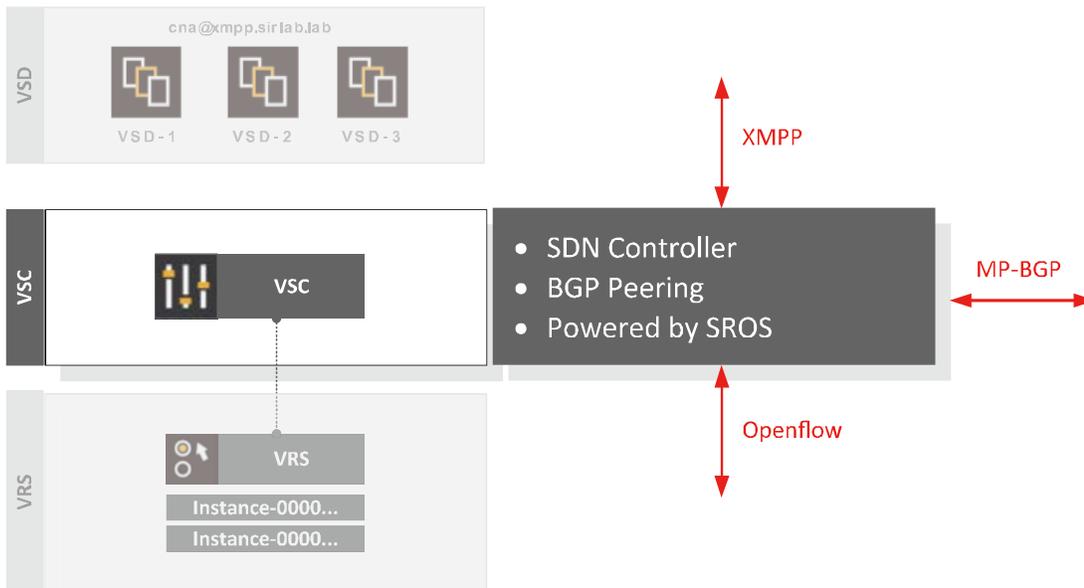


図4 : VSCコンポーネント

VSCはベア・メタル・インスタンスにデプロイされます。VSCには2つのサブネットがあります。基盤ネットワークに接続されたコントロール・インターフェースと、別のコンポーネント (VRS) に接続するマネジメント・ネットワークです。

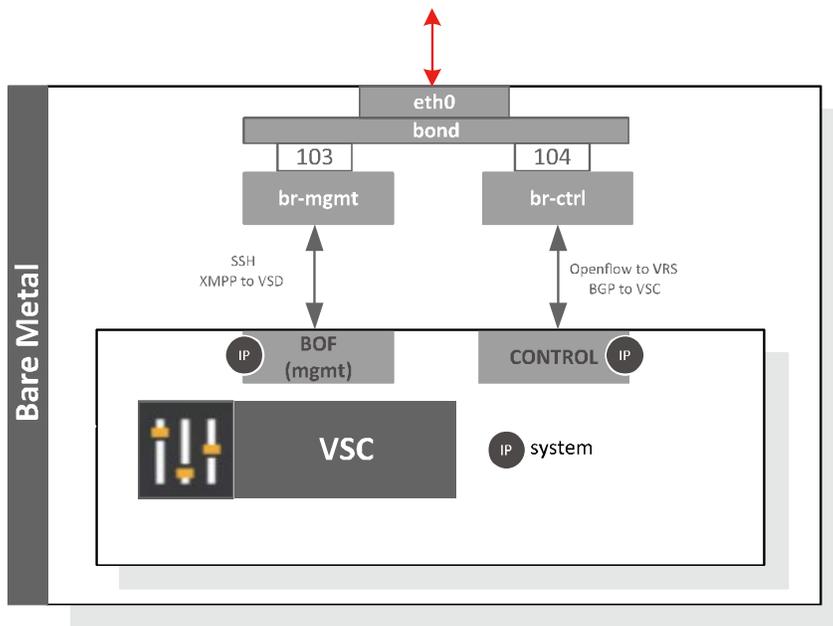


図5 : VSCのネットワーク接続

## VSCをインストールする

このセクションでは、Oracle Cloud Infrastructure内のベア・メタル・サーバーにNuage Networks VSCソフトウェアをインストールするプロセスについて説明します。プロセスを完了すると、VSCイメージがサーバーで実行され、ログインを求めるプロンプトが表示されます。

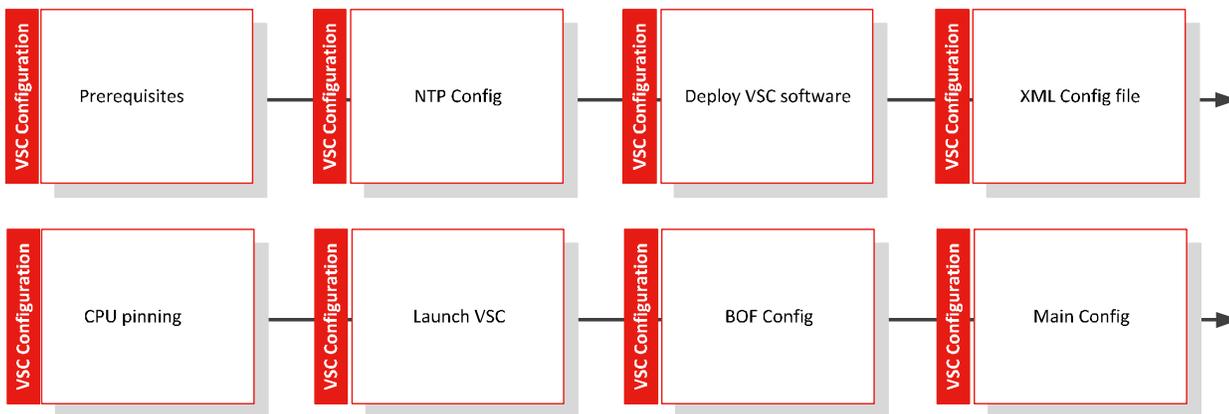


図6 : VSCのインストールの大まかな流れ

## 前提条件

VSCをデプロイする前に、次の要件を満たす必要があります。計画実施の一部として、必要なタスクを実行してください。

- マネジメント・ネットワーク用のIPアドレスがすでに割り当てられている。
- マネジメント用とデータ・トラフィック用に2つのネットワーク・インタフェースが個別に設定され、2つのLinuxブリッジ・インタフェースに接続されている。この指示は、マネジメント用のブリッジbr0とデータ用のbr1が作成され、アタッチされていることを前提としたものです。
- 少なくとも1つのNTPサーバーが構成され、同期されている。サーバーを設定する際には、すべてのコンポーネントに対してNTPサーバーを設定する必要があります。VMを定義すると、VMがタイムスタンプを取得しますが、これが10秒以上ずれないようにする必要があります。
- VSCソフトウェア・ファイルをサーバーにコピーするための手段が必要。

これらの要件を満たしたら、次のようにして依存項目をインストールします。

```
# yum install kvm libvirt bridge-utils service libvirtd start chkconfig libvirtd on
```

## NTPサーバーをインストールするには

1. NTPサーバーをインストールします。

```
[opc@instance-vsc ~]$ sudo su  
[root@instance-vsc opc]# yum install ntp
```

2. タイム・ゾーンを設定するには、まず/etc/localtimeを削除しなければならない場合があります。/etc/ntp.confファイルをチェックし、必要な値と同期してください。

このソリューションでは、次の行を追加します。

```
[root@instance-vsc opc]# cat >> /etc/ntp.conf << EOF  
server 0.centos.pool.ntp.org iburst  
server 1.centos.pool.ntp.org iburst  
server 2.centos.pool.ntp.org iburst  
server 3.centos.pool.ntp.org iburst  
EOF
```

3. NTPデーモンを再起動します。

```
[root@instance-vsc opc]# service ntpd restart  
[root@instance-vsc ntp]# date  
Tue Mar 12 13:32:32 GMT 2019
```

## VSCをインストールするには

この構成を試行する前に、前のセクションを完了していることを確認してください。KVMの管理に使用されるlibvirt APIには、VMの作成や管理を行うためのツール・セットが含まれています。

1. libvirtdを起動して、実行中であることを確認します。

```
[root@instance-vsc opc]# systemctl start libvirtd
```

注：ブート時にlibvirtdを自動的に起動するには、「# systemctl enable libvirtd」と入力します。

```
[root@instance-vsc opc]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-02-01 12:09:43 GMT; 1 months 8 days ago
```

2. VSCソフトウェア・ファイルを宛先ホストにコピーします。

```
[root@instance-vsc opc]# cd /var/lib/libvirt/images/
[root@instance-vsc images]# scp admin@source_host :/share/nfs/nuage/5.3.3/
Nuage-VSC-5.3.3-128.tar.gz ./ nuage-vsc- 5.3.3-128.tar.gz
```

3. ホストのVSCソフトウェア・ファイルを解凍します。このデプロイでは、単一ディスクを実装します。

```
[root@instance-vsc images]# tar xzvf nuage-vsc- 5.3.3-128.tar.gz
[root@instance-vsc images]# cd single_disk/
[root@instance-vsc single_disk]#
[root@instance-vsc single_disk]# cp vsc_singledisk.qcow2 ./vsc1.qcow2
```

4. qcow2のインストールを開始します。

```
[root@instance-vsc single_disk]# chown qemu:qemu vsc1.qcow2
```

5. Nuage Networksのソフトウェア・リリースとともに提供されたVsc.xmlファイルを使用して、新しいVMを定義します。VSC XML構成を編集して、VMの名前を変更するか、ディスク・パスとファイル名を変更します。

注：次の構成では、利用可能な物理CPUに対してvCPUをピンングしています。cpupin要素は、CPUのチューニング可能なパラメータに関する詳細を提供するものです。vcpupinは、ベア・メタル・インスタンスの物理CPUのうち、どれがドメインのvCPUに対してピンングされるかを指定するために使用します。CPUのチューニングの詳細については、<https://libvirt.org/formatdomain.html>にアクセスしてください。

```
[root@instance-vsc images]# cat vsc.xml
<domain type='kvm'>
  <name>vsc</name>
  <description>Timos VM</description>
  <memory>4147483</memory>
  <currentMemory>4147483</currentMemory>
```

```

<vcpu current='4'>4</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='0'/>
  <vcpupin vcpu='1' cpuset='1'/>
  <vcpupin vcpu='2' cpuset='2'/>
  <vcpupin vcpu='3' cpuset='3'/>
</cputune>
<os>
  <type arch='x86_64' machine='rhel6.0.0'>hvm</type>
  <smbios mode='sysinfo'/>
</os>
<sysinfo type='smbios'>
  <system>
    <entry name='product'>Nuage Networks Virtual Services
Controller</entry>
  </system>
</sysinfo>
<features>
  <apic/>
</features>
<cpu>
  <topology sockets='4' cores='1' threads='1'/>
</cpu>
<clock offset='utc'>
  <timer name='pit' tickpolicy='catchup'/>
  <timer name='rtc' tickpolicy='catchup'/>
</clock>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>coredump-destroy</on_crash>
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <controller type='ide' index='0'>
    <alias name='ide0'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
function='0x1'/>
  </controller>
  <controller type='usb' index='0'>
    <alias name='usb0'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
function='0x2'/>
  </controller>
  <disk type='file' device='disk' snapshot='no'>
    <driver name='qemu' type='qcow2' cache='writethrough'/>
    <source file='/var/lib/libvirt/images/vsc.qcow2'/>
    <target dev='hda' bus='ide'/>
    <boot order='1'/>
  </disk>
  <interface type='bridge'>
    <source bridge='brV2MGMT'/>
    <model type='virtio'/>

```

```

        <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
function='0x0' />
    </interface>
    <interface type='bridge'>
        <source bridge='brV1CTRL' />
        <model type='virtio' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
function='0x0' />
    </interface>
    <serial type='pty'>
        <source path='/dev/pts/1' />
        <target port='0' />
        <alias name='serial0' />
    </serial>
    <console type='pty' tty='/dev/pts/1'>
        <source path='/dev/pts/1' />
        <target type='serial' port='0' />
        <alias name='serial0' />
    </console>
</devices>
<seclabel type='none' />
</domain>

```

```
[root@instance-vsc images]#
```

## 6. VSCを定義します。

```
[root@instance-vsc opc]# virsh define vsc.xml
```

## 7. autostartを構成します。

```
[root@instance-vsc opc]# virsh autostart vscl
```

## 8. コンソールにログインします。デフォルトでは、ユーザー名とパスワードは「admin/admin」です。

```
[root@instance-vsc opc]# virsh console vscl
```

```

login as: opc
Authenticating with public key "rsa-key-20181119"
Last login: Tue Mar 12 11:20:04 2019 from 156.151.8.1
[opc@instance-vsc ~]$ sudo su
[root@instance-vsc opc]# virsh console vsc
Connected to domain vsc
Escape character is ^]

Login: admin
Password:

*A:vsc-ocip# █

```

## VSCを構成する

次に、VSC自体を構成します。使用されているコマンドの詳細については、VSPインストール・ガイドを参照してください。

VSCコントローラの構成には、次のコンポーネントが含まれています。

- ブート・オプション・ファイル（BOF）：デバイスをブートするのに必要なパラメータが含まれています。Nuage NetworksのVSCでは、システム・ブート時に読み取られる、`bof.cfg`というファイルが使用されます、このファイルは、VSCを正常にブートするために必要となる、基本的な低レベル・システム構成に使用されます。
- メイン構成：メインの構成が含まれています（LAG設定やBGP設定など）。

### ブート・オプション・ファイルの構成を実行する

このソリューションでは、すべての構成とブート・イメージがCF1ディスク（ユーザー・ディスク）に保存される、単一ディスクのインストールを使用します。それでは、BOFファイルを更新していきます。

1. ブート・オプション・ファイルのコンテキストに移動するには、「`bof<Enter>`」と入力します。プロンプトに、`bof`コンテキストへの変更が示されます。

```
*A:vsc-ocip# bof
*A:vsc-ocip>bof#
```

2. マネジメントIPアドレスを割り当てます。

```
*A:vsc-ocip>bof# address 10.0.103.101/24 active
```

3. DNSサーバーを構成します。

```
*A:vsc-ocip>bof# primary-dns 10.5.0.50
```

---

**注：** DNSサーバーは最大3つまで構成できます（プライマリ、セカンダリ、およびターシャリ）。

---

4. DNSドメインを構成します。

```
*A:vsc-ocip>bof# dns-domain sirlab.lab
```

5. マネジメントIPネットワークの静的ルートを構成します。

```
*A:vsc-ocip>bof# static-route 0.0.0.0/1 next-hop 10.0.103.1
*A:vsc-ocip>bof#128.0.0.0/1 next-hop 10.0.103.1
```

---

**注：** 静的ルート0.0.0.0/0は、BOF構成によって受け入れられません。デフォルトのルートが必要な場合は、2つの静的ルート（0.0.0.0/1と128.0.0.0/1）を代わりに構成してください。

---

6. マネジメント・ゲートウェイに対する接続を検証します。

```
*A:vsc-ocip>bof# ping router "management" 10.0.103.1
```

```
*A:vsc-ocip>bof# ping router "management" 10.0.103.1
PING 10.0.103.1 56 data bytes
64 bytes from 10.0.103.1: icmp_seq=1 ttl=64 time=0.492ms.
64 bytes from 10.0.103.1: icmp_seq=2 ttl=64 time=0.357ms.
64 bytes from 10.0.103.1: icmp_seq=3 ttl=64 time=0.444ms.
64 bytes from 10.0.103.1: icmp_seq=4 ttl=64 time=0.429ms.
64 bytes from 10.0.103.1: icmp_seq=5 ttl=64 time=0.409ms.

---- 10.0.103.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 0.357ms, avg = 0.426ms, max = 0.492ms, stddev = 0.044ms
*A:vsc-ocip>bof#
```

7. プライマリ構成の場所とネットワーク設定が正しく設定されていることを確認します。

```
*A:vsc-ocip>bof# primary-config cfl:\config.cfg
*A:vsc-ocip>bof# autonegotiate
*A:vsc-ocip>bof# wait 3
```

---

注：システムは、primary-configで指定された構成を使用するよう試みます。指定されたファイルが見つからない場合、システムは自動的に、secondary-configで指定された場所から構成を取得しようとし、そこでも見つからない場合は、tertiary-configから取得しようとしています。

---

8. 構成をCF1に保存します。

```
*A:vsc-ocip>bof# save
```

9. VSCをリブートして、保存したブート・オプションをロードします。

```
*A:vsc-ocip>bof# exit
*A:vsc-ocip# admin reboot
WARNING: Configuration and/or Boot options may have changed since the last
save. Are you sure you want to reboot (y/n)? y
```

## メイン構成を実行する

最も基本的な構成として、VSCには次のセクションが含まれます。

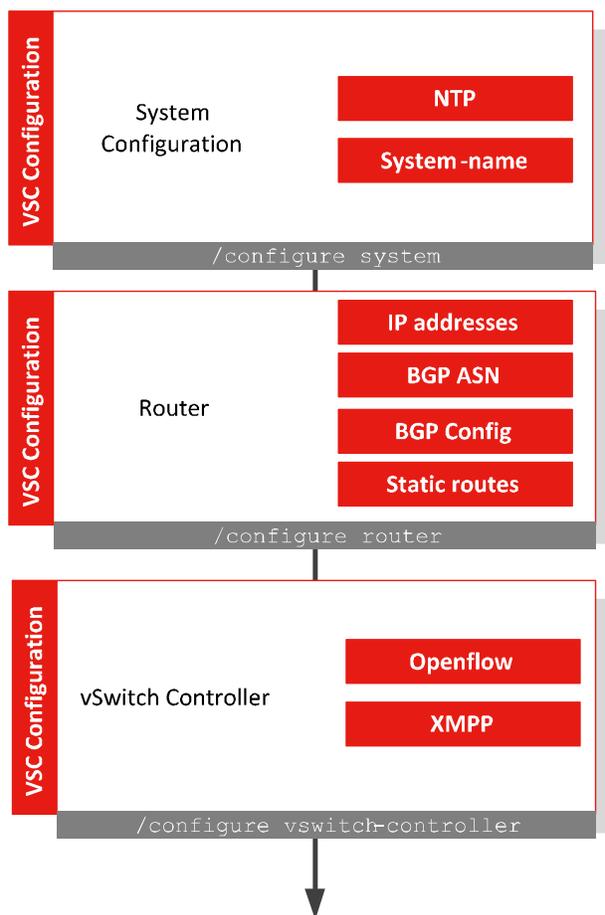


図7 : VSCの構成フロー

## システム構成

このセクションでは、基本的なシステム情報を構成します（システム名、連絡先情報、タイム・ゾーン、ゾーンに応じて時刻を表示するためのNTPパラメータなど）。

構成	パラメータ
システム名	vsc-ocip
連絡先情報	EMEA Cloud Pursuit Team
所在地	40.5214579,-3.8913381
NTPサーバー	10.5.0.50
タイム・ゾーン	UTC

システム・パラメータを構成するには、次のコマンドを実行します。

```
*A:vsc-ocip# configure system
*A:vsc-ocip>config>system# name vsc-ocip
*A:vsc-ocip>config>system# contact "EMEA Cloud Pursuit Team"
*A:vsc-ocip>config>system# location "40.5214579,-3.8913381"

*A:vsc-ocip>config>system# time
*A:vsc-ocip>config>system>time#ntp
*A:vsc-ocip>config>system>time>ntp# server 10.5.0.50
*A:vsc-ocip>config>system>time>ntp# no shutdown
*A:vsc-ocip>config>system>time>ntp# exit

*A:vsc-ocip>config>system>time# zone UTC
*A:vsc-ocip>config>system>time#
```

## ルーター構成

このセクションでは、VSPのコントロール・インタフェース、ASN番号、およびデフォルト・ルート構成します。

構成	パラメータ
コントロールIPアドレス	10.0.104.101/24
ASN番号	65005
ルーターID	10.0.104.101
ルート	デフォルトのルート

1. システムのコントロール・インタフェースを構成し、ステータスをチェックします。

```
*A:vsc-ocip# configure router
*A:vsc-ocip>config>router# interface "control" address 10.0.104.101/24
```

```
*A:vsc-ocip# show router interface

=====
Interface Table (Router: Base)
=====
Interface-Name      Adm      Opr (v4/v6)  Mode      Port/SapId
IP-Address          PfxState
-----
control             Up       Up/Down     Network  A/2:0
  10.0.104.101/24  n/a
system              Down    Down/Down   Network  system
  -                  -
-----
Interfaces : 2
=====
*A:vsc-ocip#
```

- 構成で使用されるBGP ASNを構成します。

```
*A:vsc-ocip>config>router#autonomous-system 65005
```

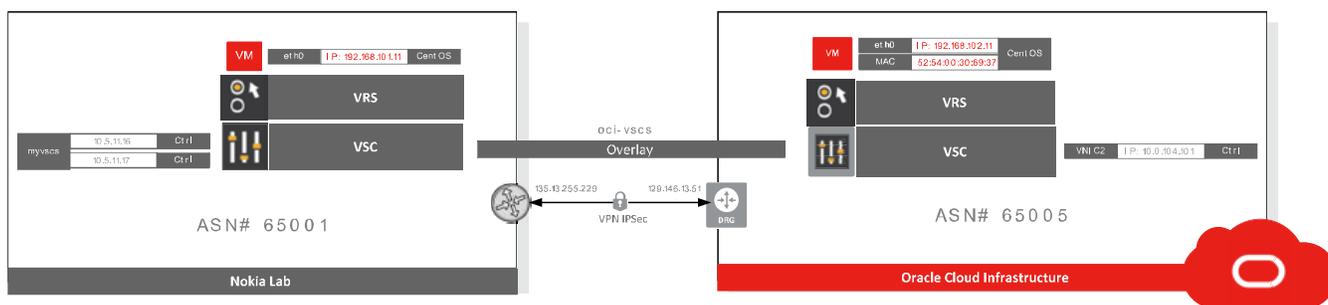
- 仮想ルーターのルーターIDを構成します。

```
*A:vsc-ocip>config>router# router-id 10.0.104.101
```

- デフォルトのルーターを構成します。

```
*A:vsc-ocip>config>router# static-route 0.0.0.0/0 next-hop 10.0.104.1
```

マルチプロトコル・ボーダー・ゲートウェイ・プロトコル (MP-BGP) は、VSC間におけるVMのMAC/IP到達可能性情報を配信するために使用されます。2つの環境間の接続を確立してください。



構成	パラメータ
BGPグループ	myvscs
BGPピアASN	65001
BGPネイバー	10.5.11.16、10.5.11.17
マルチホップ	TTL値 : 5
接続試行回数	2
ピア・トラッキング	有効
ラピッド・ウィズドロー	有効

- 次のコマンドを実行します。

```
*A:vsc-ocip>config>router# bgp
*A:vsc-ocip>config>router>bgp# connect-retry 2
*A:vsc-ocip>config>router>bgp# enable-peer-tracking
*A:vsc-ocip>config>router>bgp# rapid-withdrawal
*A:vsc-ocip>config>router>bgp# group "myvscs"
*A:vsc-ocip>config>router>bgp>group$ family evpn
*A:vsc-ocip>config>router>bgp>group$ type external
*A:vsc-ocip>config>router>bgp>group$ multihop 5
```

```
*A:vsc-ocip>config>router>bgp>group$ peer-as 65001
*A:vsc-ocip>config>router>bgp>group$ neighbor 10.5.11.16
*A:vsc-ocip>config>router>bgp>group$ neighbor 10.5.11.17
*A:vsc-ocip>config>router>bgp>group$ exit
*A:vsc-ocip>config>router>bgp# no shutdown
*A:vsc-ocip>config>router>bgp# exit
```

注：他の構成パラメータについては、「付録C：Virtualized Service Controllerの構成ファイル」を参照してください

## vSwitchの構成

VSCをSDNコントローラとして動作させるには、次の行を構成します。

```
*A:vsc-ocip>config# vswitch-controller
*A:vsc-ocip>config>vswitch-controller# xmpp-server "vsc-ocip@xmpp.sirlab.lab"
*A:vsc-ocip>config>vswitch-controller# exit
```

XMPPを構成すると、VSCはVSDサーバーのFQDNに対するejabberd接続を開始します。この接続は、新しいVMのポリシー情報をダウンロードしたり、ポリシーの更新を受信したりするために必要です。一方、OpenFlowは、VRSからのOpenFlow接続のリスニングを開始するために必要です。

XMPPサーバーは、指定されたユーザー名を使用して、VSC用のユーザーを自動的に作成します。

```
*A:vsc-ocip# show vswitch-controller xmpp-server detail

=====
XMPP Server Table
=====
XMPP FQDN           : xmpp.sirlab.lab
XMPP User Name     : vsc-ocip
Last changed since : 0d 03:44:33
State              : Functional
IQ Tx.             : 123                IQ Rx.             : 123
IQ Error           : 0                  IQ Timed Out       : 0
IQ Min. Rtt       : 20                  IQ Max. Rtt       : 120
IQ Ack Rcvd.      : 123
Nuage Updates Rcvd.: 2                  VSD Updates Rcvd. : 688
Nuage Msg Tx.     : 98                  Nuage Msg Rx.     : 98
Nuage Msg Ack. Rx.: 98                  Nuage Msg Error   : 0
Nuage Msg Min. Rtt: 30                  Nuage Msg Max. Rtt: 120
Nuage Sub Tx.     : 4                  Nuage UnSub Tx.   : 0
Nuage Msg Timed Out: 0
Encryption Type   : none

=====
*A:vsc-ocip#
```

VSDへの接続をテストします。

```
*A:vsc-ocip# show vswitch-controller vsd detail

=====
VSD Server Table
=====
VSD User Name      : cna@xmpp.sirlab.lab/vsd1
Uptime            : 9d 14:38:09
Status            : available
Nuage Msg Tx.     : 1467
Nuage Msg Rx.     : 1467
Nuage Msg Ack. Rx.: 1467
Nuage Msg Error   : 0
Nuage Msg TimedOut : 0
Nuage Msg MinRtt  : 40
Nuage Msg MaxRtt  : 11080

VSD User Name      : cna@xmpp.sirlab.lab/vsd3
Uptime            : 9d 14:38:14
Status            : available
Nuage Msg Tx.     : 1546
Nuage Msg Rx.     : 1546
Nuage Msg Ack. Rx.: 1546
Nuage Msg Error   : 0
Nuage Msg TimedOut : 0
Nuage Msg MinRtt  : 40
Nuage Msg MaxRtt  : 1040

VSD User Name      : cna@xmpp.sirlab.lab/vsd2
Uptime            : 9d 14:38:33
Status            : available
Nuage Msg Tx.     : 2298
Nuage Msg Rx.     : 2298
Nuage Msg Ack. Rx.: 2298
Nuage Msg Error   : 0
Nuage Msg TimedOut : 0
Nuage Msg MinRtt  : 40
Nuage Msg MaxRtt  : 13100

=====
*A:vsc-ocip#
```

VSDダッシュボードから操作する場合：

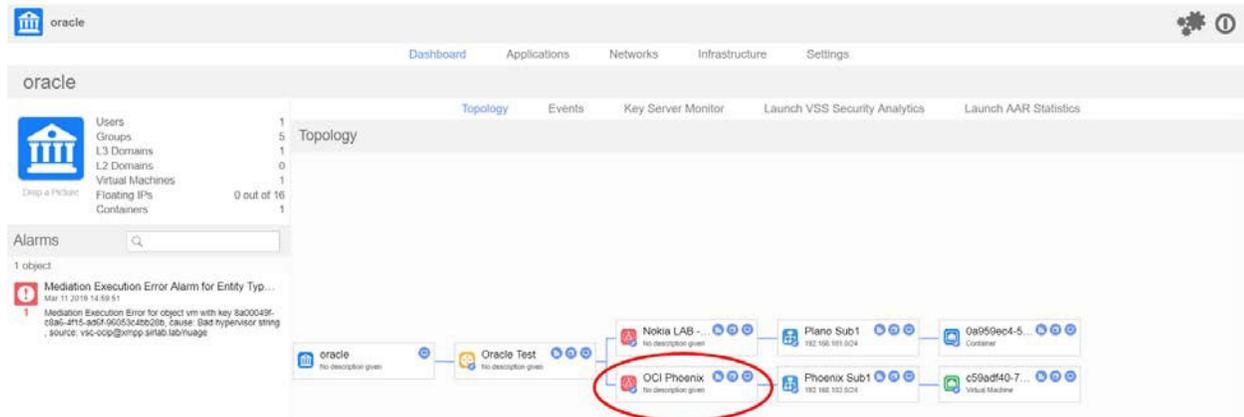


図8：VSDからのVSCビュー

## Virtual Routing and Switchingのインストール

Virtual Routing and Switching（VRS）は、L2/L3転送を処理します。VRSでは、外部エンドポイントとの通信を可能にする、幅広いL2/L3カプセル化手法（VLANからVxLANおよびGREまで）がサポートされます。

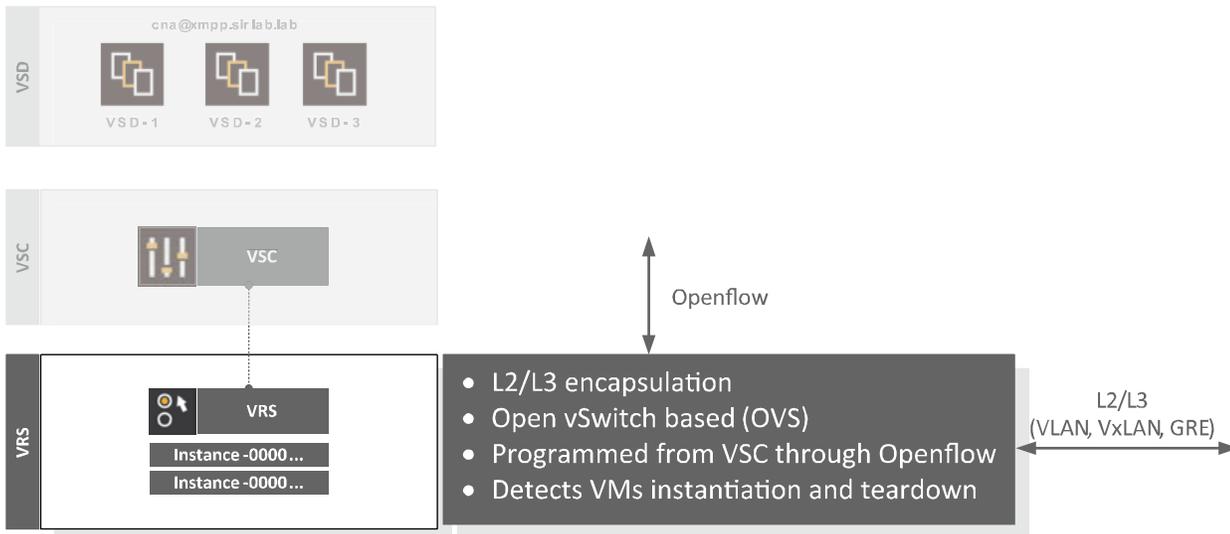


図9：VRSコンポーネント

これがインストールの最終ステップです。次の手順（およびフロー）は、デプロイ時のガイダンスとして使用できます。

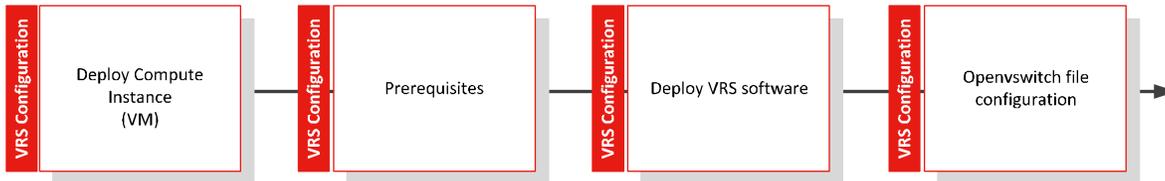


図10 : VRSのインストール・フロー

## 前提条件

VRSをインストールする前に、ターゲット・ホストで次の依存項目が揃っていることを確認してください。

- VRSによって必要とされるパッケージ：
  - Python twistedライブラリ
  - Perl JSONライブラリ
  - vconfigパッケージ
- 追加ソフトウェア：
  - KVM
  - libvirt

---

注 : CentOS 7イメージ（VRSの動作確認済）も必要となります。

---

## VRSをインストールする

1. コンピュート・インスタンスをデプロイするには、Oracle Cloud Infrastructureコンソールのナビゲーション・メニューから、「コンピュート」を選択し、「インスタンス」を選択します。
2. 「インスタンスの作成」をクリックします。
3. インスタンスの名前（たとえば、Instance-VRS）を指定し、可用性ドメイン（AD 3）を選択した後、「イメージ・ソースの変更」を選択し、「CentOS 7」を選択します。

ORACLE Cloud

### Create Compute Instance

Oracle Cloud Infrastructure Compute lets you provision and manage compute hosts, known as instances. You can launch instances as needed to meet your compute and application requirements.

Name your instance  
instance-VRS

Select an availability domain for your instance

AD 1  
shPn.PHX-AD-1

AD 2  
shPn.PHX-AD-2

AD 3  
shPn.PHX-AD-3 ✓

Choose an operating system or image source

CentOS 7  
Image Build: 2019.02.23-0  
CentOS is a free, open-source Linux distribution that is suitable for use in enterprise cloud environments. For more information, see <https://www.centos.org>

Change Image Source

- 「仮想マシン」を選択し、「シェイプの変更」をクリックして、「VM.Standard2.2」を選択した後、「シェイプを選択」をクリックします。

Browse All Shapes

A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance. See [Compute Shapes](#) for more information.

Shape Name	OCPU	Memory (GB)	Local Disk (TB)	Network Bandwidth	Max. Total VNICs
<input type="checkbox"/> VM.Standard2.1	1	15	Block Storage only	1 Gbps	2
<input checked="" type="checkbox"/> VM.Standard2.2	2	30	Block Storage only	2 Gbps	2
<input type="checkbox"/> VM.Standard2.4	4	60	Block Storage only	4.1 Gbps	4
<input type="checkbox"/> VM.Standard2.8	8	120	Block Storage only	8.2 Gbps	8
<input type="checkbox"/> VM.Standard2.16	16	240	Block Storage only	16.4 Gbps	16
<input type="checkbox"/> VM.Standard2.24	24	320	Block Storage only	24.6 Gbps	24
<input type="checkbox"/> VM.DenseIO2.8	8	120	6.4 TB NVMe SSD	8.2 Gbps	8
<input type="checkbox"/> VM.DenseIO2.16	16	240	12.8 TB NVMe SSD	16.4 Gbps	16
<input type="checkbox"/> VM.DenseIO2.24	24	320	25.6 TB NVMe SSD	24.6 Gbps	24

1 Selected

Select Shape Cancel

Showing 9 item(s)

- 「SSHキーの追加」セクションで、SSH鍵ファイルを選択するか、テキスト・ボックスにSSH鍵を貼り付けます。

6. 「**ネットワーキングの構成**」セクションで、VCNのコンパートメント、VCN、サブネットのコンパートメント、およびサブネット (**mgt-plain**) を選択します。

Create Compute Instance

Configure networking

Virtual cloud network compartment  
SDN  
telefonicacloud2 (root)/SDN

Virtual cloud network  
VCN

Subnet compartment  
SDN  
telefonicacloud2 (root)/SDN

Subnet ⓘ  
mgt-plain

7. 「**作成**」をクリックします。

数分後、インスタンスが（次の画像のように）起動します。

Compute > Instances > Instance Details

Instance-VRS

Create Custom Image Start Stop Reboot Terminate Apply Tag(s) Create Instance Configuration

Instance Information Tags

**Instance Information**

Availability Domain: [shPrPHX-AD-1](#)  
Fault Domain: [FAULT-DOMAIN-3](#)  
Region: [pdx](#)  
Shape: [VM.Standard2.2](#)  
Virtual Cloud Network: [VCN](#)  
Maintenance Reboot: -

**Primary VNIC Information**

Private IP Address: [10.0.103.3](#)  
Public IP Address:

Image: [Instance-VRS](#)  
OCID: [~332jq](#) [Show Copy](#)  
Launched: [Wed, 30 Jan 2019 21:30:22 GMT](#)  
Compartment: [telefonicacloud2 \(root\)/SDN](#)  
Launch Mode: [NATIVE](#)

Internal FQDN: [Instance-vs-](#) [Show Copy](#)  
Subnet: [mgt-plain](#)

This instance's traffic is controlled by its firewall rules in addition to the associated [Subnet's](#) Security Lists.

8. VMが起動したら、インスタンスにログインし、「前提条件」セクションで示したリポジトリをインストールまたは更新します。

```
[root@instance-vrs opc]# yum install libvirt
[root@instance-vrs opc]# yum install qemu-kvm
```

---

**警告** : Nuage Networksのリリース・ノートで、サポートされているカーネルのリストを必ずチェックしてください。システム更新（yum更新）によって、サポートされていないオペレーティング・システム・バージョンに更新されることがあります。

---

9. 現在実行されているカーネルをチェックします。

```
[root@instance-vrs opc]# uname -r
3.10.0-957.1.3.el7.x86_64
```

10. サポートされている各オペレーティング・システムには、VRSの.tar.gzファイルがあります。VRSソフトウェア・ファイルを宛先ホストにコピーします。

```
[root@instance-vrs opc]# mkdir nuage
[root@instance-vrs opc]# cd /home/opc/nuage
[root@instance-vrs nuage]# scp admin@source_host
:/share/nfs/nuage/5.3.3/nuage-VRS- 5.3.3-128.tar.gz ./ nuage-VRS- 5.3.3-
128.tar.gz
```

11. ホストのNuage VRSソフトウェア・ファイルを解凍します。

```
[root@instance-vrs nuage]# tar xzvf nuage-VRS- 5.3.3-128.tar.gz
```

12. nuage-openvswitchパッケージとnuage-bgpパッケージをインストールします。

```
[root@instance-vrs nuage]# yum localinstall nuage-openvswitch- 5.3.3-
128.el7.x86_64.rpm
[root@instance-vrs nuage]# yum localinstall nuage-bgp- 5.3.3-
128.el7.x86_64.rpm
```

13. パッケージがインストールされたことを確認します。

```
[root@instance-vrs nuage]# yum list installed | grep nuage
nuage-metadata-agent.x86_64          5.3.3-128.el7
installed
nuage-openvswitch.x86_64            5.3.3-128.el7
installed
```

14. /etc/default/openvswitchを編集して、パーソナリティ、プラットフォーム (KVM) 、およびコントローラIPアドレスを設定します。

```
# PERSONALITY: vrs/vrs-g/vrs-b/nsg/nsg-br/nsg-duc/vdf/vdf-g/none (default:
vrs)
PERSONALITY=vrs

# PLATFORM: kvm/xen/esx-i/lxc. Only apply when in VRS personality.
# lxc and kvm can exist at the same time as a , separated list like so:
# PLATFORM: "kvm, lxc"
PLATFORM="kvm"

# ACTIVE_CONTROLLER: Primary controller IP. Only valid IP addresses will
be
# accepted. To delete the controller comment out the ACTIVE_CONTROLLER
# variable below
ACTIVE_CONTROLLER=10.0.104.101
#
```

15. VRSを再起動します。

```
[root@instance-vrs opc]# service openvswitch restart
```

16. VRSが起動し、VSCコントローラに接続されたことを確認します。

```
[root@instance-vrs opc]# ovs-vsctl show
66870816-6a7c-4f30-b341-68f56eaef19c
    Bridge "alubr0"
        Controller "ctrl1"
            target: "tcp:10.0.104.101:6633"
            role: master
            is_connected: true
        Port svc-pat-tap
            Interface svc-pat-tap
                type: internal
        Port "svc-rl-tap1"
            Interface "svc-rl-tap1"
        Port "vnet0"
            Interface "vnet0"
        Port nuage-bgp
            Interface nuage-bgp
                type: internal
        Port svc-spat-tap
            Interface svc-spat-tap
                type: internal
        Port "svc-rl-tap2"
            Interface "svc-rl-tap2"
        Port "alubr0"
            Interface "alubr0"
                type: internal
    ovs_version: "5.3.3-128-nuage"
```

```

other_config: {acl-non-tcp-timeout="180", acl-tcp-timeout="3600",
connid-type="", connid-val="", connobj-limit="320000", control-cos="7",
control-dscp="56", controller-less-duration="", "disable-dhcp4=no, dual-
vtep=no, flow-collection="true", flow-limit="200000", fp-ports="", head-
less-duration="", nat-traversal-enabled=no, network-namespace=default, nw-
uplink="ens4f0", openflow_audit_timer="180", personality=vrs,
platform=kvm, revertive-controller=no, revertive-timer="300", stats-
collector="10.5.0.11:39090,10.5.0.12:39090,10.5.0.13:39090", stats-
collector-type=ip, stats-enable="true", sticky-ecmp-timeout="0", syslog-
dest=localhost, syslog-dest-port="514", sysmon-timer="3600", tcp-mss="0",
vdf_uplink="", vport-init-stateful-timer="300", vss-stats-interval="30"}
[root@instance-vrs opc]#

```

## 17. VSCからVRSへの接続を確認します。

```

*A:vsc-ocip# show vswitch-controller vswitches

=====
VSwitch Table
=====

Legend: * -> Primary Controller ! -> NSG in Graceful Restart
-----
Vswitch-Instance      Personality  Uptime                Num
VM/Host/Bridge/Cont
                               Num Resolved
-----
*va-10.0.103.3/1      VRS         37d 21:08:39         1/0/0/0
                               1/0/0/0
-----

No. of virtual switches: 1
-----

*A:vsc-ocip#

```

特定のシェル・コマンドをVRSに送信し、出力をキャプチャして、それをコントローラで表示することにより、VSCから直接VRSを問い合わせることができます。

## 18. VRSの背後にあるVMをリストします。

```

*A:vsc-ocip# tools vswitch 10.0.103.3 command "ovs-appctl vm/show"

```

```

*A:vsc-ocip# tools vswitch 10.0.103.3 command "ovs-appctl vm/show"
Name: centos      UUID: eeac7c9c-159b-476e-8fd5-b4081d77b1d8
State: running Reason: booted event_id: 0x3
event_ts: 0x5c5476e9
no_of_nics: 1 flags: 0x0 xml_length: 625

*A:vsc-ocip# █

```

19. VRSで構成されたルーティング表をチェックすることもできます。

```
*A:vsc-ocip# tools vswitch 10.0.103.3 command "ovs-appctl vrf/list alubr0"
```

```
*A:vsc-ocip# tools vswitch 10.0.103.3 command "ovs-appctl vrf/list alubr0"
vrf: 1506137622
```

Routes	Duration	Cookie	Pkt Count	Pkt Bytes	EVPN-Id or Local/remote Out port
192.168.102.1	3276057s	0x1	0	0	2 (MPLS-GRE: )
192.168.102.1	3276057s	0x1	0	0	2 (MPLS-GRE: )
192.168.102.1	3276057s	0x1	0	0	2 (MPLS-GRE: )
192.168.102.11	3276057s	0x1	456280	44265261	1578805703
)					
192.168.101.11	2048580s	0x1	2046365	200543770	)
192.168.102.0/24	3276057s	0x1	0	0	1578805703
)					
192.168.101.0/24	2048580s	0x1	0	0	)
0.0.0.0/0	3276057s	0x1	24261	2134660	)
0.0.0.0/0	3276057s	0x1	0	0	)

## Nuage Networks SDNのテスト

次に示すのは、このホワイト・ペーパーで使用されたネットワーク・アーキテクチャです。VMの起動には、環境（Nokia LabまたはOracle Cloud Infrastructure）に応じて、DockerとCentOSが使用されました。

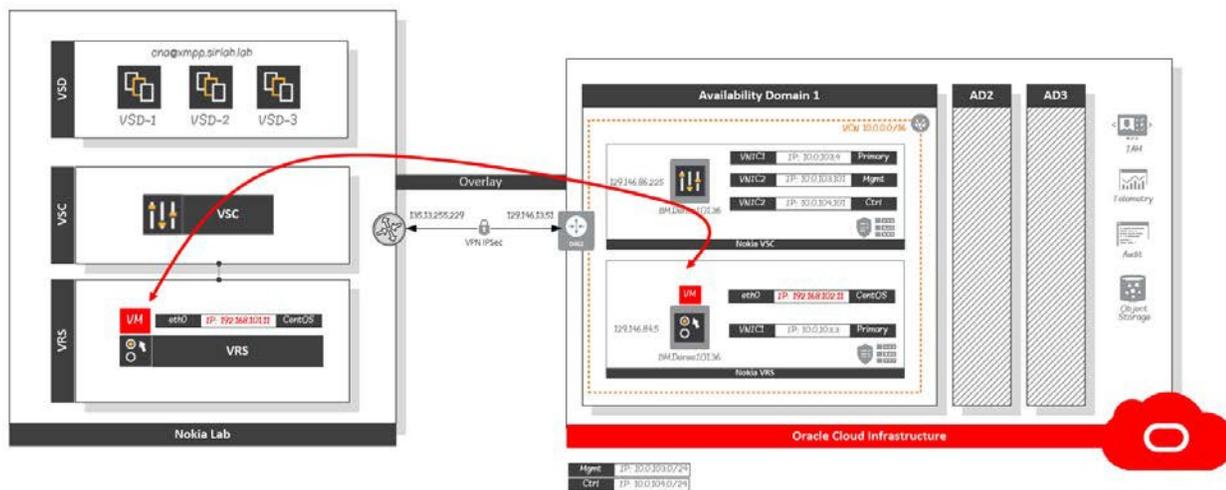


図11：エンドツーエンドの通信

このホワイト・ペーパーでは、両方のサイト間の通信で暗号化を必須としました。そのため、ソリューションではIPSecトンネルが構成されました。なお、これを行うと、既存のIPネットワークを介してEthernetトラフィックを運ぶために、オーバーレイ・ネットワークがデプロイされるため、パフォーマンス低下が生じる可能性があります。

IPSecトンネルを使用せずに両サイト間の接続を実行することもできますが、通信を保護することを強くお勧めします。

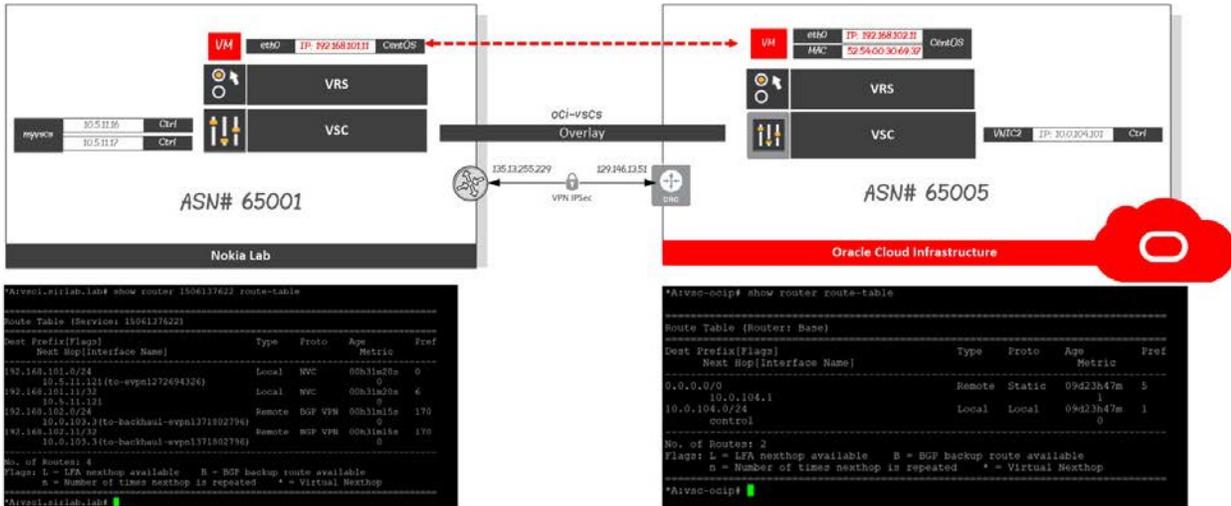


図12: ルーティング表

次の画像は、Oracle Cloud InfrastructureでのCentOS VMのネットワーク構成を示したものです。

```

[root@ocip-centos ~]# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.102.11 netmask 255.255.255.0 broadcast 192.168.102.255
    inet6 fe80::5054:ff:fe30:6937 prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:30:69:37 txqueuelen 1000 (Ethernet)
    RX packets 464193 bytes 44271911 (42.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4868765 bytes 7699728176 (7.1 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@ocip-centos ~]#
  
```

図13: CentOS VMのネットワーク構成

速度と帯域幅のネットワーク・パフォーマンス・テストは、本番環境と非本番環境のいずれについても必ず必要です。次に示すのは、ネットワークで実行されたいくつかのテストの結果です。

```
[root@ocip-centos ~]# iperf -c 192.168.101.11
-----
Client connecting to 192.168.101.11, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[  3] local 192.168.102.11 port 39426 connected with 192.168.101.11 port 5001
Throughput
[ ID] Interval      Transfer      Bandwidth
[  3] 0.0-10.1 sec  68.8 MBytes  57.2 Mbits/sec
[root@ocip-centos ~]#

--- 192.168.101.11 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99171ms
Latency
rtt min/avg/max/mdev = 37.786/37.872/38.868/0.112 ms
[root@ocip-centos ~]#

[root@ocip-centos ~]# traceroute 192.168.101.11
Traceroute
traceroute to 192.168.101.11 (192.168.101.11), 30 hops max, 60 byte packets
 1 gateway (192.168.102.1)  1.555 ms  1.849 ms  1.495 ms
 2 192.168.101.1 (192.168.101.1)  39.872 ms  38.717 ms  39.823 ms
 3 192.168.101.11 (192.168.101.11)  38.723 ms * *
```

図14：ネットワーク・パフォーマンス

最後に、私たちはポリシー・エンジン（VSD）から、Oracle Cloud Infrastructureにデプロイされた複数のVSCを制御できるかどうかをチェックしました。次の図は、環境にデプロイされたすべてのコントローラを、オンプレミスかクラウドかにかかわらず、1つのコンソールから管理する方法を示したものです。

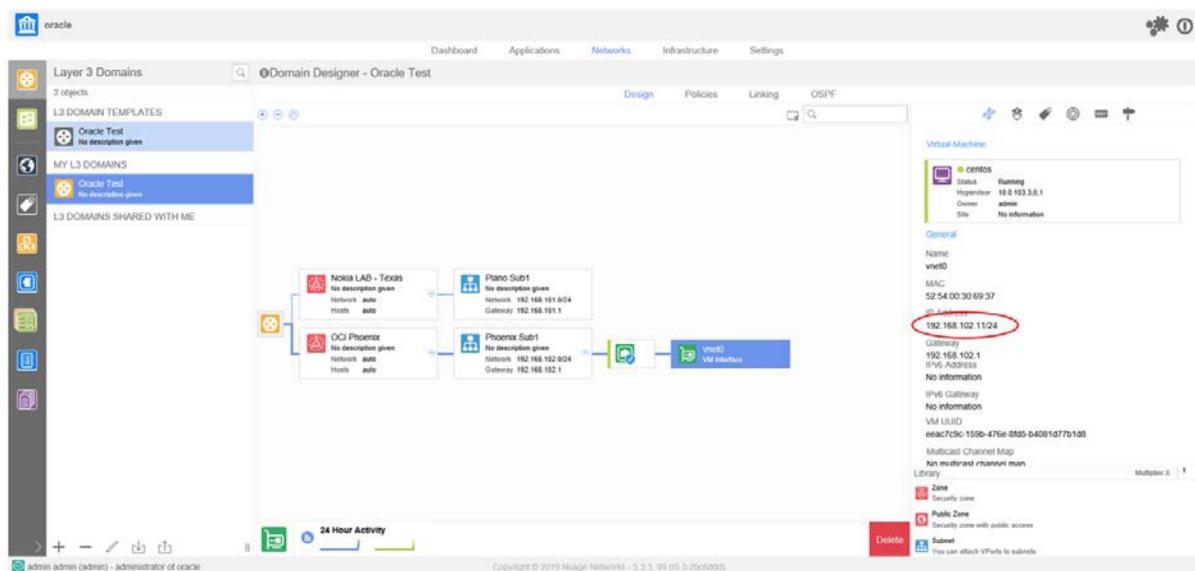


図15 : VSDでのレイヤー3ドメイン

## 付録A : Oracle Cloud InfrastructureにセカンダリNICをアタッチする

Oracle Cloud Infrastructureコンソールを使用して、セカンダリVNICを追加することができます。

1. ナビゲーション・メニューから、「コンピュー」を選択し、「インスタンス」を選択します。
2. インスタンスの名前（このケースでは、**Instance-VSC**）をクリックして、詳細を表示します。
3. 「リソース」で、「アタッチされたVNIC」をクリックします。
4. 「VNICの作成」をクリックします。
5. 次の情報を入力します。
  - 名前 : vf-mgt-nic
  - 仮想クラウド・ネットワーク : VCN
  - サブネット : mgt-plain
  - コントロールIPアドレス : 10.0.103.101

残りのエントリは空白のままにします。

Click here.'. Below this, there are three input fields: 'NAME (Optional)' with the value 'vf-mgt-nic', 'VIRTUAL CLOUD NETWORK' with a dropdown menu showing 'VCN', and 'SUBNET (Optional)' with a dropdown menu showing 'mgt-plain'. There is also a checkbox labeled 'Skip Source/Destination Check' which is currently unchecked. Below the checkbox is a note: 'The source/destination check causes this VNIC to drop any network traffic whose source or destination is not this VNIC. Only check the checkbox if you want this VNIC to skip the check and forward that traffic (for example, to perform Network Address Translation)'. The 'Primary IP Information' section has a 'PRIVATE IP ADDRESS (Optional)' field with the value '10.0.103.101'. Below this field is a note: 'Must be within 10.0.103.2 to 10.0.103.254. Cannot be in current use.'"/>

6. 上記の手順を繰り返し、次の情報を使用して、2つ目のVNICを追加します。

- 名前 : vf-ctl-nic
- 仮想クラウド・ネットワーク : VCN
- サブネット : ctl-plain
- コントロールIPアドレス : 10.0.104.101

新しい2つのVNICを作成した後は、インスタンス情報が次の図のようになります。

## Attached VNICs

Displaying 3 Attached VNICs

NIC 0			
 ATTACHED	<a href="#">Instance-VSC</a> (Primary VNIC) OCID: ...hwlgva <a href="#">Show</a> <a href="#">Copy</a> Attached: Mon, 11 Mar 2019 17:49:22 GMT Compartment: SDN	Private IP Address: 10.0.104.2 Fully Qualified Domain Name: instance-vsc... <a href="#">Show</a> <a href="#">Copy</a> Public IP Address: 130.61.118.245	Subnet: <a href="#">ctt-plain</a> Skip Source/Destination Check: No MAC Address: 90:E2:BA:3B:E1:08 VLAN Tag: 0
 ATTACHED	<a href="#">vf-mgt-nic</a> OCID: ...kzpcha <a href="#">Show</a> <a href="#">Copy</a> Attached: Tue, 12 Mar 2019 11:15:44 GMT Compartment: SDN	Private IP Address: 10.0.103.101 Fully Qualified Domain Name: <i>Unavailable</i> Public IP Address:	Subnet: <a href="#">mgt-plain</a> Skip Source/Destination Check: No MAC Address: 02:00:17:01:C1:D5 VLAN Tag: 1
 ATTACHED	<a href="#">vf-ctl-nic</a> OCID: ...jefusq <a href="#">Show</a> <a href="#">Copy</a> Attached: Tue, 12 Mar 2019 11:22:48 GMT Compartment: SDN	Private IP Address: 10.0.104.101 Fully Qualified Domain Name: <i>Unavailable</i> Public IP Address:	Subnet: <a href="#">ctt-plain</a> Skip Source/Destination Check: No MAC Address: 02:00:17:01:9B:E2 VLAN Tag: 2

7. ディレクトリを作成し、`secondary_vnic_all_configure.sh`スクリプトをダウンロードします。SSHを使用してインスタンスを接続し、次のコマンドを実行します。

```
mkdir /opt/secondary_vnic
cd /opt/secondary_vnic
wget
https://docs.cloud.oracle.com/iaas/Content/Resources/Assets/secondary_vnic_all_configure.sh
chmod u+x secondary_vnic_all_configure.sh
```

8. ユニット・ファイルを作成します。

```
# vi /etc/systemd/system/secondary_vnic_all_configure.service
```

9. ファイルに次の行を貼り付けます。

```
[Unit]
Description=Add the secondary VNIC at boot
After=basic.target
[Service]
Type=oneshot
ExecStart=/opt/secondary_vnic/secondary_vnic_all_configure.sh -c
[Install]
WantedBy=default.target
```

10. ユニット・ファイルを有効にします。

```
# chmod 664 /etc/systemd/system/secondary_vnic_all_configure.service
# systemctl enable
/etc/systemd/system/secondary_vnic_all_configure.service
# systemctl list-unit-files|egrep secondary_vnic_all_configure.service
```

11. Oracle Cloud Infrastructureコンソールのインスタンス詳細ページで「リブート」ボタンをクリックして、インスタンスをリブートします。

12. 2つ目のVNICが自動的に構成されていることを確認します。

```
# uptime; ip address
```

## 付録B : Virtualized Service ControllerのBOFファイル

```
*A:vsc-ocip# show bof
=====
BOF (Memory)
=====
primary-image          cf1:\timos\cpm.tim
primary-config         cf1:\config.cfg
address                10.0.103.101/24 active
primary-dns            10.5.0.50
dns-domain             sirlab.lab
static-route           0.0.0.0/1 next-hop 10.0.103.1
static-route           128.0.0.0/1 next-hop
autonegotiate
duplex                 full
speed                  100
wait                   3
persist                off
no li-local-
save no li-
separate no
fips-140-2             115200
=====
*A:vsc-ocip#
```

## 付録C : Virtualized Service Controllerの構成ファイル

メイン構成について、「admin display-config」と入力します。

```
*A:vsc-ocip# admin display-config
# TiMOS-DC-C-5.3.3-100 cpm/i386 NUAGE VSC Copyright (c) 2000-2018 Nokia.
# All rights reserved. All use subject to applicable license agreements.
# Built on Wed Oct 31 13:42:50 PDT 2018 [d429da] by builder in /rel5.3-
DC/release/panos/main

# Generated MON MAR 11 13:40:49 2019 UTC

exit all
configure
#-----
echo "System Configuration"
#-----
name "vsc-ocip"
```



```
contact "EMEA Cloud Pursuit Team"
location "40.5214579,-3.8913381b"
snmp
exit
time
    ntp
        ntp-server
        server 10.5.0.50
        no shutdown
    exit
    snmp
        shutdown
    exit
    zone UTC
exit
thresholds
    rmon
    exit
exit

#-----
echo "System Security Configuration"
#-----
    system
        security
            user "admin"
                password "L8PI6XXQN0W1jz.nZ92v2E" hash2
                access console
                console
                    member "administrative"
            exit
        exit
    exit
exit

#-----
echo "Log Configuration"
#-----
    log
    exit

#-----
echo "System Security Cpm Hw Filters and PKI Configuration"
#-----
    system
        security
        exit
    exit

#-----
echo "QoS Policy Configuration"
#-----
    qos
    exit
```

```
#-----
echo "Card Configuration"
#-----
#-----
echo "Service Configuration"
#-----
    service
    exit
#-----
echo "LAG Configuration"
#-----
    lag 98
        description "Multichassis interconnect LAG"
        encap-type dot1q
        qos
        exit
        lacp active administrative-key 36864
        no shutdown
    exit
#-----
echo "Management Router Configuration"
#-----
    router management
    exit
#-----
echo "Router (Network Side) Configuration"
#-----
    router
        interface "control"
            address 10.0.104.101/24
            no shutdown
        exit
        interface "system"
            shutdown
        exit
        vxlan
        exit
        autonomous-system 65005
        router-id 10.0.104.101
#-----
echo "Static Route Configuration"
#-----
    static-route 0.0.0.0/0 next-hop 10.0.104.1
#-----
echo "Web Portal Protocol Configuration"
#-----
    exit
```

```
-----
echo "Service Configuration"
-----
    service
        customer 1 create
            description "Default customer"
        exit
    exit
-----
echo "Router (Service Side) Configuration"
-----
    router
-----
echo "BGP Configuration"
-----
    bgp
        connect-retry 2
        enable-peer-tracking
        rapid-withdrawal
        rapid-update evpn
        group "myvscs"
            family evpn
            type external
            multihop 5
            peer-as 65001
            neighbor 10.5.11.16
            exit
            neighbor 10.5.11.17
            exit
        exit
    no shutdown
    exit
exit
-----
echo "System Time NTP Configuration"
-----
    system
        time
            ntp
            exit
        exit
    exit
-----
echo "Virtual Switch Controller Configuration"
-----
    vswitch-controller
        xmpp-server "vsc-ocip@xmpp.sirlab.lab"
        open-flow
        exit
    xmpp
```

```
    exit
    ovssdb
    exit
    init
    exit
    exit

exit all

# Finished MON MAR 11 13:41:04 2019 UTC
*A:vsc-ocip#
```

## 付録D : Virtual Routing and Switchingの構成ファイル

```
[root@instance-vrs opc]# cat /etc/default/openvswitch
### Configuration options for openvswitch

# Copyright (C) 2009, 2010, 2011 Nicira, Inc.

# FORCE_COREFILES: If 'yes' then core files will be enabled.
# FORCE_COREFILES=yes

# OVSDB_SERVER_PRIORITY: "nice" priority at which to run ovsdb-server.
#
# OVSDB_SERVER_PRIORITY=-10

# VSWITCHD_PRIORITY: "nice" priority at which to run ovs-vswitchd.
# VSWITCHD_PRIORITY=-10

# VSWITCHD_MLOCKALL: Whether to pass ovs-vswitchd the --mlockall option.
#   This option should be set to "yes" or "no". The default is "yes".
#   Enabling this option can avoid networking interruptions due to
#   system memory pressure in extraordinary situations, such as multiple
#   concurrent VM import operations.
# VSWITCHD_MLOCKALL=yes

# OVS_CTL_OPTS: Extra options to pass to ovs-ctl. This is, for example,
# a suitable place to specify --ovs-vswitchd-wrapper=valgrind.
# OVS_CTL_OPTS=
# DELETE_BRIDGES: Delete the previously existing ones, default is "no".
# DELETE_BRIDGES=no

# PERSONALITY: vrs/vrs-g/vrs-b/nsg/nsg-br/nsg-duc/vdf/vdf-g/none (default: vrs)
PERSONALITY=vrs

# UUID: uuid assigned to nsg
UUID=

# CPE_ID: 4 byte id assigned to nsg
```

```
CPE_ID=

# DATAPATH_ID: Datapath id of the nsg
DATAPATH_ID=

# UPLINK_ID: uplink id assigned to nsg
UPLINK_ID=

# NETWORK_UPLINK_INTF: uplink interface of the host
NETWORK_UPLINK_INTF=ens4f0
# NETWORK_NAMESPACE: namespace to create pat interfaces, iptables & route rules
NETWORK_NAMESPACE=

# VDF_UPLINK: Adds intf to use as uplink for vdf for creating vlan interfaces
VDF_UPLINK=

#
# VRSG_PEER_IP: Applies only when in GateWay mode
# VRSG_PEER_IP=0.0.0.0

# PLATFORM: kvm/xen/esx-i/lxc. Only apply when in VRS personality.
# lxc and kvm can exist at the same time as a , separated list like so:
# PLATFORM: "kvm, lxc"
PLATFORM="kvm"

# DEFAULT_BRIDGE: Nuage managed bridge
DEFAULT_BRIDGE=alubr0

# BRIDGE_MTU: Configurable bridge MTU
#BRIDGE_MTU=

# MCAST_UNDERLAY_TX_INTF: mcast tx interface
#MCAST_UNDERLAY_TX_INTF=

# GW_HB_BRIDGE: Name of the gateway heartbeat bridge
GW_HB_BRIDGE=

# GW_HB_VLAN: vlan for heart beat exchange in gateways
GW_HB_VLAN=

# GW_HB_TIMEOUT: timeout for heart beat exchange in gateways in milliseconds
GW_HB_TIMEOUT=2000

# MGMT_ETH: Comma separated names of management Ethernet interfaces
MGMT_ETH=

# UPLINK_ETH: Comma separated names of Ethernet interfaces used for uplink
UPLINK_ETH=

# GW_PEER_DATAPATH_ID: Datapath ID of peer gateway to which access resiliency
# will be established
```

```

GW_PEER_DATAPATH_ID=

# GW_ROLE: Specify role of a gateway.
# Set to "master" if all access link ports of the gateway should act as
# a master in a resilient setup, "backup" if it should act as a backup.
GW_ROLE="backup"

#Sample Mcast Underlay interface and range configuration
# MCAST_UNDERLAY_INTF_1: mcast interface
#MCAST_UNDERLAY_INTF_1=

# MCAST_UNDERLAY_INTF_RANGE_1: mcast interface range
#MCAST_UNDERLAY_INTF_RANGE_1=

# CONNID_TYPE: This could be set to type uuid or string
# CONNID_TYPE=

# CONNID_VAL: This could be a uuid value or a string
# CONNID_VAL=

# CLIENT_KEY_PATH: SSL client key file path
# CLIENT_KEY_PATH=

# CLIENT_CERT_PATH: SSL client certificate file path
# CLIENT_CERT_PATH=

# CA_CERT_PATH: CA certificate file path
# CA_CERT_PATH=

# CONN_TYPE: ssl or tcp
CONN_TYPE=tcp

# ACTIVE_CONTROLLER: Primary controller IP. Only valid IP addresses will be
# accepted. To delete the controller comment out the ACTIVE_CONTROLLER
# variable below
ACTIVE_CONTROLLER=10.0.104.101
#
# STANDBY_CONTROLLER: Secondary controller IP. Only valid IP addresses
# will be accepted. To delete the controller comment out the STANDBY_CONTROLLER
# variable below
# STANDBY_CONTROLLER=
#
# NUAGE_MONITOR_PRIORITY:
# NUAGE_MONITOR_PRIORITY= -10
#
# VM_MONITOR_PRIORITY:
# VM_MONITOR_PRIORITY= -10
#
# MANAGEMENT_INTERFACE: Management interface (example: eth0)
# MANAGEMENT_INTERFACE=eth0

```

```
# DHCP_RELAY_ADDRESS: IP Address of the DHCP relay server
#DHCP_RELAY_ADDRESS=

# STATS_COLLECTOR_ADDRESS: IP or FQDN of the STATS relay server (applicable only
for NSG)
# STATS_COLLECTOR_ADDRESS=

# STATS_COLLECTOR_TYPE: IP or FQDN (default: FQDN) (applicable only for NSG)
# STATS_COLLECTOR_TYPE=

# STATS_COLLECTOR_PORT: ssl port of the STATS relay server (applicable only for
NSG)
# STATS_COLLECTOR_PORT=
#
# DB_FILE: OVSDB file location (default: /etc/openvswitch)
# DB_FILE=

# FLOW_EVICTION_THRESHOLD: Number of flows at which eviction from
kernel flow table will be triggered (default : 2500)
#FLOW_EVICTION_THRESHOLD=

# DATAPATH_SYNC_TIMEOUT: Datapath flow stats sync timeout
# specified in milliseconds (default: 1000)
#DATAPATH_SYNC_TIMEOUT=

# DATAPATH_FLOW_IDLE_TIMEOUT : Datapath flow idle timeout
# specified in milliseconds (default: 5000)
#DATAPATH_FLOW_IDLE_TIMEOUT=

# SKB_LRO_MOD_ENABLED: enable or disable LRO modification in skb for
# improving performance. Allowed values: 'yes' or 'no'
SKB_LRO_MOD_ENABLED=no

# PROBE_INTERVAL : Configurable openflow echo timer
# specified in milliseconds (default: 5000)
#PROBE_INTERVAL=
#
# DEFAULT_LOG_LEVEL: default log level at openvswitch start
# DEFAULT_LOG_LEVEL=any:file:dbg
DEFAULT_LOG_LEVEL=

# REVERTIVE_CONTROLLER: Revertive behavior of VRS (default : no)
REVERTIVE_CONTROLLER=no

# REVERTIVE_TIMER: Revert timer for the revertive behavior of VRS (default: 300
seconds)
# Valid range : 10 - 7200 seconds
REVERTIVE_TIMER=300

# CONTROLLER_LESS_DURATION : Controller-less duration of VRS (applicable only
for NSG)
```

```
# (default is 3600 seconds. Valid Range: 3600 seconds(1 hr) - 86400 seconds(24
hr))
# -1 indicates infinite duration
#CONTROLLER_LESS_DURATION=3600

# Service IPV4 subnet for kubernetes
K8S_SERVICE_IPV4_SUBNET=0.0.0.0/8
# Pod IPV4 subnet for kubernetes
K8S_POD_NETWORK_CIDR=0.0.0.0/8

# FP_PORTS: List of fast-path ports to be recognized as Network ports
(applicable only for Advanced VRS)
#FP_PORTS=

# DUAL_VTEP_VRS: VRS supports dual-uplinks (default:no) (applicable only for DC
environments)
#DUAL_VTEP_VRS=

# DISABLE_DHCP4: VRS will not act as dhcp server (default:no) (applicable only
# for DC environments)
#DISABLE_DHCP4=

# UPLINK1: Uplink 1 name (applicable only when DUAL_VTEP_VRS is enabled)
#UPLINK1=

# Controller configuration (applicable only when DUAL_VTEP_VRS is enabled)
# UPLINK1_ACTIVE_CONTROLLER: Active controller of Uplink 1 (applicable only when
DUAL_VTEP_VRS is enabled)
#UPLINK1_ACTIVE_CONTROLLER=

# UPLINK1_STANDBY_CONTROLLER: Standby controller of Uplink 1 (applicable only
when DUAL_VTEP_VRS is enabled)
#UPLINK1_STANDBY_CONTROLLER=

# UPLINK1_UNDERLAY_ID: Underlay ID of Uplink 1 (applicable only when
DUAL_VTEP_VRS is enabled)
#UPLINK1_UNDERLAY_ID=

# UPLINK2: Uplink 2 name (applicable only when DUAL_VTEP_VRS is enabled)
#UPLINK2=

# UPLINK2_ACTIVE_CONTROLLER: Active controller of Uplink 2 (applicable only when
DUAL_VTEP_VRS is enabled)
#UPLINK2_ACTIVE_CONTROLLER=

# UPLINK2_STANDBY_CONTROLLER: Standby controller of Uplink 2 (applicable only
when DUAL_VTEP_VRS is enabled)
#UPLINK2_STANDBY_CONTROLLER=

# UPLINK2_UNDERLAY_ID: Underlay ID of Uplink 2 (applicable only when
DUAL_VTEP_VRS is enabled)
```



```
#UPLINK2_UNDERLAY_ID=  
[root@instance-vrs opc]#
```

## 参考資料

- [Installing and Configuring KVM on Bare Metal Instances with Multi-VNIC](#)
- Nuage Networks VSP 5.3.3リリース・ノート
- Nuage Networks VSP 5.3.3インストール・ガイド



**ORACLE®**

**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

CONNECT WITH US



[blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)



[facebook.com/oracle](https://facebook.com/oracle)



[twitter.com/oracle](https://twitter.com/oracle)



[oracle.com](https://oracle.com)

**Integrated Cloud Applications & Platform Services**

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。0419

SDNとNuage Networksの統合

2019年4月

著者：Oracle Corporation



Oracle is committed to developing practices and products that help protect the environment