Deploying Citrix NetScaler VPX on Oracle Cloud Infrastructure

TECHNICAL DETAIL DOCUMENT | MAY 2018

Contents

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Revision History

The following revisions have been made to this white paper since its initial publication:

Date	Revision
May 10, 2018	Initial release
May 21, 2018	Added Software Requirements section

Overview

This is a technical detail document for deploying Citrix NetScaler VPX to run as a guest image running on top of KVM sitting on Bare Metal, on Oracle Cloud Infrastructure.

Software Requirements

This document was written based on the following software requirements:

- Citrix NetScaler VPX for KVM (RHEL), release 12.0 (Build 57.19+) – 1000 Platinum license
- Oracle Linux 7.4+

Assumptions

This document assumes the following:

- You have a passing knowledge of KVM and some of the core concepts of working with this hypervisor
- You understand the impact of guests sharing block storage devices and can determine how your guests should share storage
- You understand how to install an operating system as a guest or you know how to copy a virtual disk image between systems
- You have a working knowledge of Linux system administration and can navigate your way around Linux and edit files
- You have created a Virtual Cloud Network (VCN) within your environment and you have provisioned one or more subnets within this VCN.
- You have provisioned (or know how to provision) an Oracle Bare Metal Compute instance
- Your KVM Host should have access to Internet
- You have access to Citrix NetScaler qcow2 image for KVM. You will have to import this virtual machine image in qcow2 format.

Audience

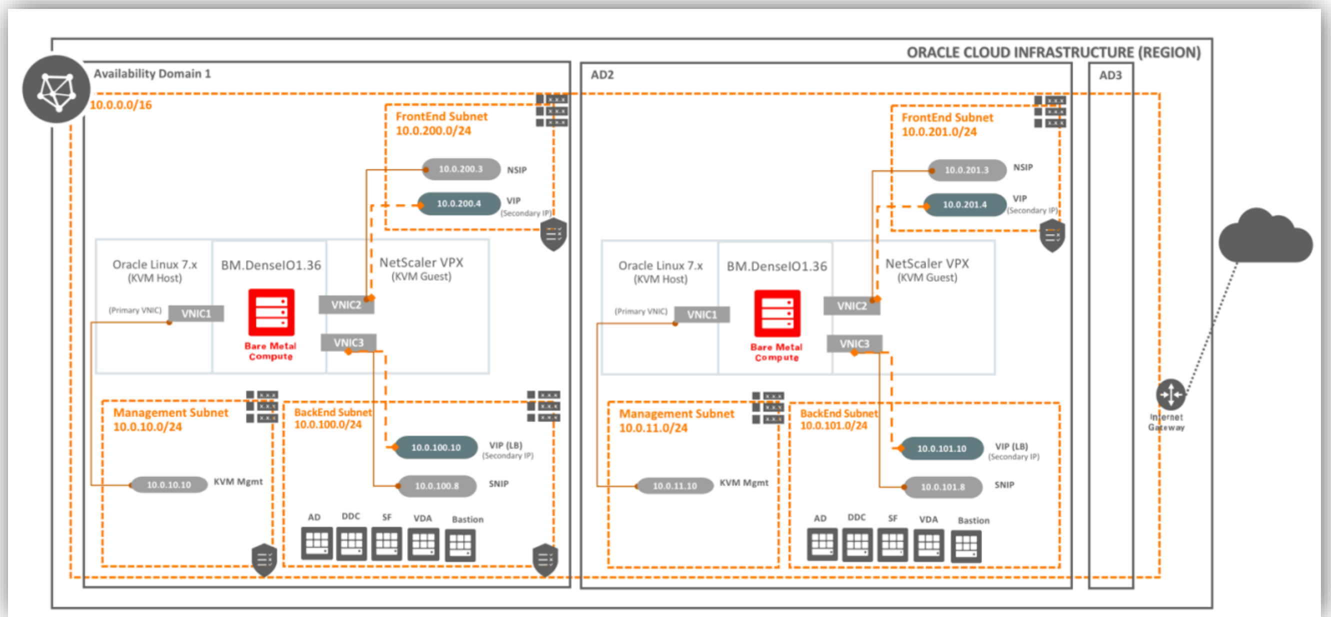
Customers who want to deploy Citrix NetScaler VPX on OCI

Target Scenarios

- Secure remote access to XenApp, XenDesktop or XenMobile
- General server load balancing

Technical Architecture

OCI VCN, frontend & backend subne



The picture above highlight a typical architecture for deploying Citrix NetScaler VM to OCI. We have BM instances in Frontend subnet (or “front-end” subnet), private instances in Backend subnet (or “back-end” subnet). In addition to these subnets, you will also require a subnet for the KVM host that can be completely separated from the Guest VM topology.

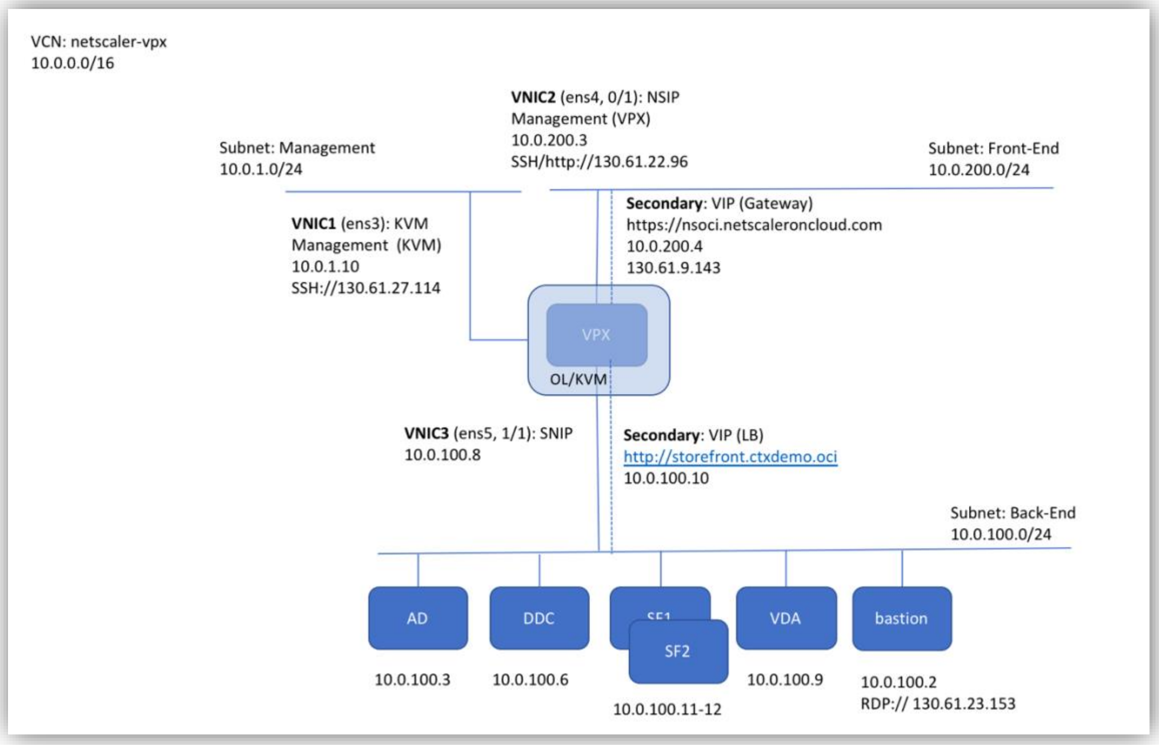
Preparing the OCI Tenancy

This guide is composed of 10 steps to setup the OCI Tenancy and deploy Citrix NetScaler VPX as a KVM guest running on a BM instance:

1- Create Oracle Virtual Cloud Network
2- Create Internet Gateway, Security Lists & Route Tables
3- Create Subnets
4- Launch Bare Metal Instance (KVM Host)
5- Attach Secondary Virtual Network Interface Cards (vNICS)
6- Install KVM on the Bare Metal Host (OS Setup + Network cards with support to Virtual Functions)
7- Upload qcow2 image file to Object Storage (Bucket) and create a PAR
8- Create KVM Domain
9- Attach Network Interfaces to KVM domain
10- Access NetScaler-VM as a KVM Guest

Sample Scenario

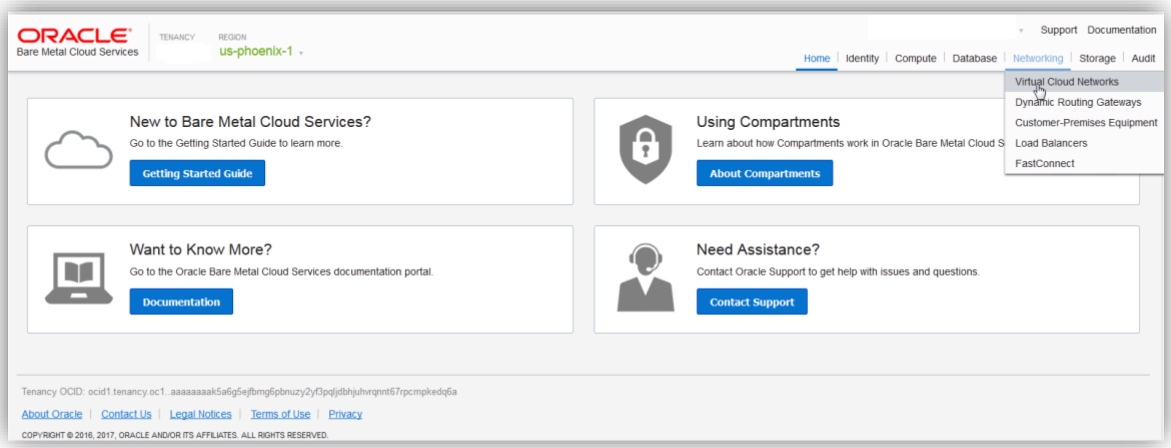
To illustrate how to setup OCI and the Citrix NetScaler VPX VM, we are providing a sample logical diagram from a Customer Project that will be explored in the next sections:



Create Virtual Cloud Network using the OCI Console

Oracle Virtual Cloud Network is a software-defined network that you set up in Oracle data centers. A subnet is a subdivision of a cloud network. Each subnet exists in a single Availability Domain and consists of a contiguous range of IP addresses that do not overlap with other subnets in the cloud network.

In the Console, click **Networking**.
Choose a compartment you have permission to work in (on the left side of the page). The page will update to display only the resources in that compartment. If you're not sure which compartment to use, contact an administrator



New to Bare Metal Cloud Services?
Go to the Getting Started Guide to learn more.

Using Compartments
Learn about how Compartments work in Oracle Bare Metal Cloud S

Want to Know More?
Go to the Oracle Bare Metal Cloud Services documentation portal.

Need Assistance?
Contact Oracle Support to get help with issues and questions.

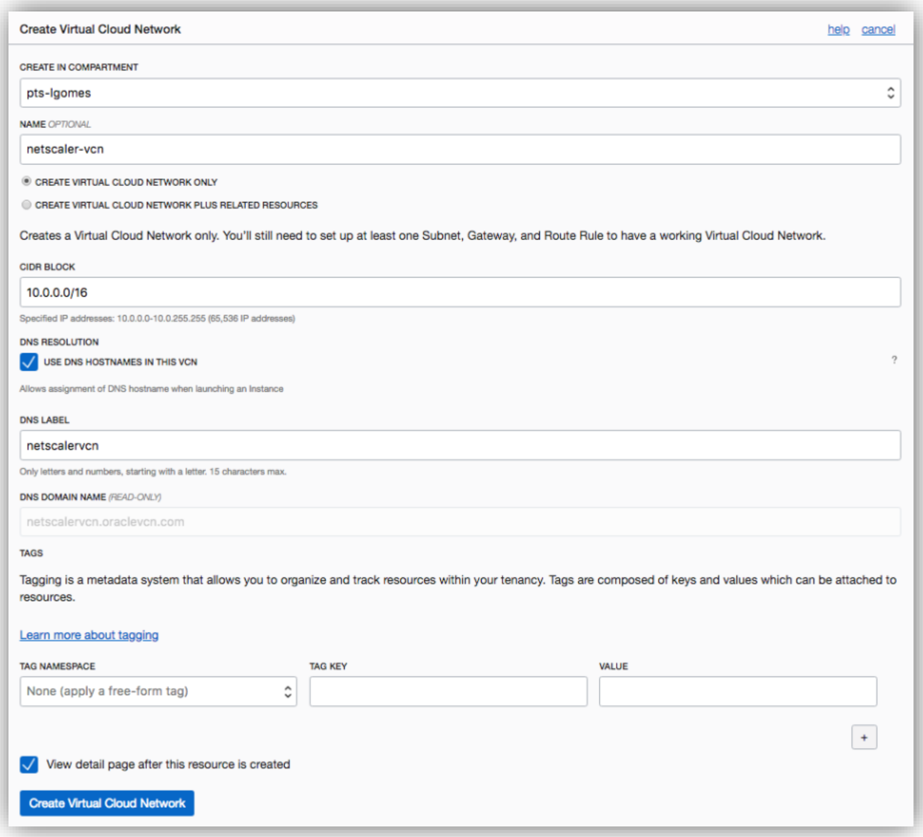
Tenancy OCID: ocid1.tenancy.oc1..aaaaaaaak5a6g5ejfbmg6pbnuzy2yf3pqljdbhjuhvrqnnt67rpcmpkedq6a

COPYRIGHT © 2016, 2017, ORACLE AND/OR ITS AFFILIATES. ALL RIGHTS RESERVED.

Click **Create Virtual Cloud Network**. **Enter the following:**

- **Enter Create in Compartment:** Leave as is
- **Name:** A friendly name for the cloud network. It doesn't have to be unique, and it cannot be changed later in the Console (but you can change it with the API)
- **Create Virtual Cloud Network Only:** Make sure this radio button is selected.
- **CIDR Block:** A single, contiguous CIDR block for the cloud network. For example: 10.0.0.0/16. You *cannot* change this value later. For reference, here's a CIDR calculator
- **Use DNS Hostnames in this VCN:** If you want the instances in the VCN to have DNS hostnames (which can be used with the *Internet and VCN Resolver*, a built-in DNS capability in the VCN), select the check box for **Use DNS Hostnames in this VCN**. Then you may specify a DNS label for the VCN, or the Console will generate one for you. The dialog box will automatically display the corresponding **DNS Domain Name** for the VCN (<*VCN DNS label*>.oraclevcn.com)

- Click Create Virtual Cloud Network



Create Virtual Cloud Network

CREATE IN COMPARTMENT

pts-lgomes

NAME *OPTIONAL*

netscaler-vcn

● CREATE VIRTUAL CLOUD NETWORK ONLY

○ CREATE VIRTUAL CLOUD NETWORK PLUS RELATED RESOURCES

Creates a Virtual Cloud Network only. You'll still need to set up at least one Subnet, Gateway, and Route Rule to have a working Virtual Cloud Network.

CIDR BLOCK

10.0.0.0/16

Specified IP addresses: 10.0.0.0-10.0.255.255 (65,536 IP addresses)

DNS RESOLUTION

☑ USE DNS HOSTNAMES IN THIS VCN

Allows assignment of DNS hostname when launching an Instance

DNS LABEL

netscalervcn

Only letters and numbers, starting with a letter. 15 characters max.

DNS DOMAIN NAME *(READ-ONLY)*

netscalervcn.oraclevcn.com

TAGS

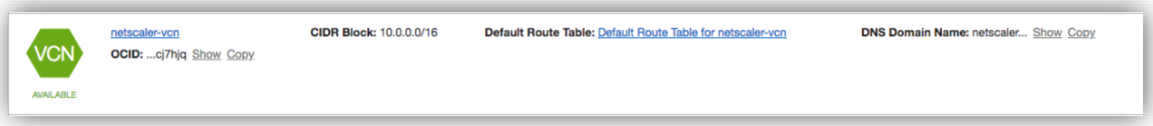
Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

☑ View detail page after this resource is created

- The cloud network is then created and displayed on the **Virtual Cloud Networks** page in the compartment you chose. Next you should create all the required resources that is required by the subnets (Internet Gateway, Security Lists, etc).



netscaler-vcn

OCID: ...cj7hjq Show Copy

CIDR Block: 10.0.0.0/16

Default Route Table: Default Route Table for netscaler-vcn

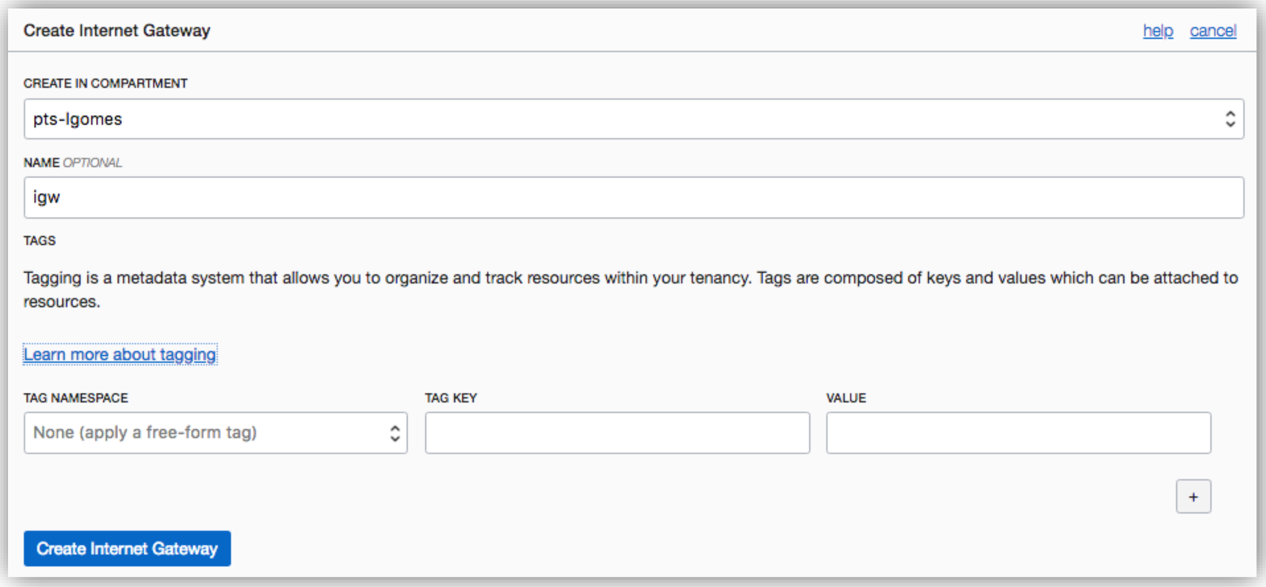
DNS Domain Name: netscaler... Show Copy

Create Internet Gateway

- Click on the VCN link “netscaler-vcn”
- Click on Internet Gateways on the left hand side
- Click on Create Internet Gateway
- Enter the following to Create the Internet Gateway

  Compartment: In the default VCN Compartment.

  Name: igw



Create Internet Gateway

CREATE IN COMPARTMENT: pts-lgomes

NAME *OPTIONAL*: igw

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

Create FrontEnd, BackEnd & Management Security Lists

Create FrontEnd Security List

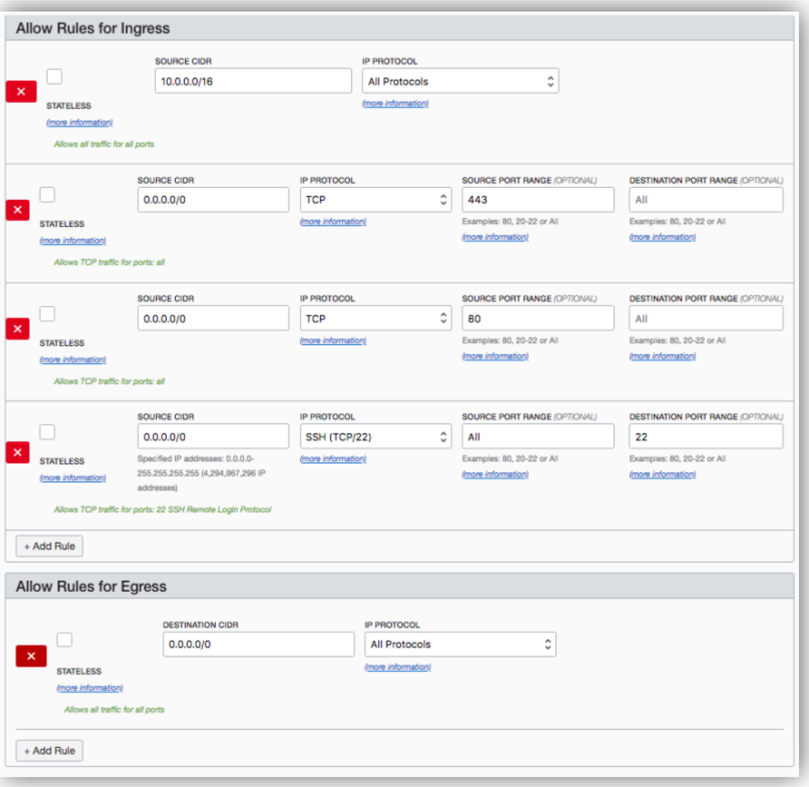
- Click on the VCN link “netscaler-vcn”
- Click on Security Lists on the left hand side
- Click on Create Security List

- Enter the following to Create the FrontEnd Security List
  Compartment: In the default VCN Compartment.
  Name: front-end-sec-list
  Under Allow Rules for Ingress, enter the following values:

Stateless	Source CIDR	IP Protocol	Source Port Range	Destination Port Range
unchecked	0.0.0.0/0	TCP	-	80
unchecked	0.0.0.0/0	TCP	-	443
unchecked	0.0.0.0/0	SSH (TCP/22)	-	22
unchecked	10.0.0.0/16	All Protocols	-	-

Under Allow Rules for Egress, enter the following values:

Stateless	Source CIDR	IP Protocol	Source Port Range	Destination Port Range
unchecked	0.0.0.0/0	All Protocols	-	-



Allow Rules for Ingress

☐ STATELESS
SOURCE CIDR: 10.0.0.0/16
IP PROTOCOL: All Protocols
Allows all traffic for all ports

☐ STATELESS
SOURCE CIDR: 0.0.0.0/0
IP PROTOCOL: TCP
SOURCE PORT RANGE (OPTIONAL): 443
Examples: 80, 20-22 or All
DESTINATION PORT RANGE (OPTIONAL): All
Examples: 80, 20-22 or All
Allows TCP traffic for ports: all

☐ STATELESS
SOURCE CIDR: 0.0.0.0/0
IP PROTOCOL: TCP
SOURCE PORT RANGE (OPTIONAL): 80
Examples: 80, 20-22 or All
DESTINATION PORT RANGE (OPTIONAL): All
Examples: 80, 20-22 or All
Allows TCP traffic for ports: all

☐ STATELESS
SOURCE CIDR: 0.0.0.0/0
Specified IP addresses: 0.0.0.0-255.255.255.255 (4,294,967,296 IP addresses)
IP PROTOCOL: SSH (TCP/22)
SOURCE PORT RANGE (OPTIONAL): All
Examples: 80, 20-22 or All
DESTINATION PORT RANGE (OPTIONAL): 22
Examples: 80, 20-22 or All
Allows TCP traffic for ports: 22 SSH Remote Login Protocol

Allow Rules for Egress

☐ STATELESS
DESTINATION CIDR: 0.0.0.0/0
IP PROTOCOL: All Protocols
Allows all traffic for all ports

Create **BackEnd Security List**

- Click on Create Security List
- Enter the following to Create the BackEnd Security List

Compartment: In the default VCN Compartment.

Name: back-end-sec-list

Under Allow Rules for Ingress, enter the following values:

Stateless	Source CIDR	IP Protocol	Source Port Range	Destination Port Range
unchecked	10.0.0.0/16	All Protocols	-	-

Under Allow Rules for Egress, enter the following values:

Stateless	Source CIDR	IP Protocol	Source Port Range	Destination Port Range
Unchecked	0.0.0.0/0	All Protocols	-	-

Create **Management Security List**

- Click on Create Security List
- Enter the following to Create the Management Security List

Compartment: In the default VCN Compartment.

Name: mgmt-sec-list

Under Allow Rules for Ingress, enter the following values:

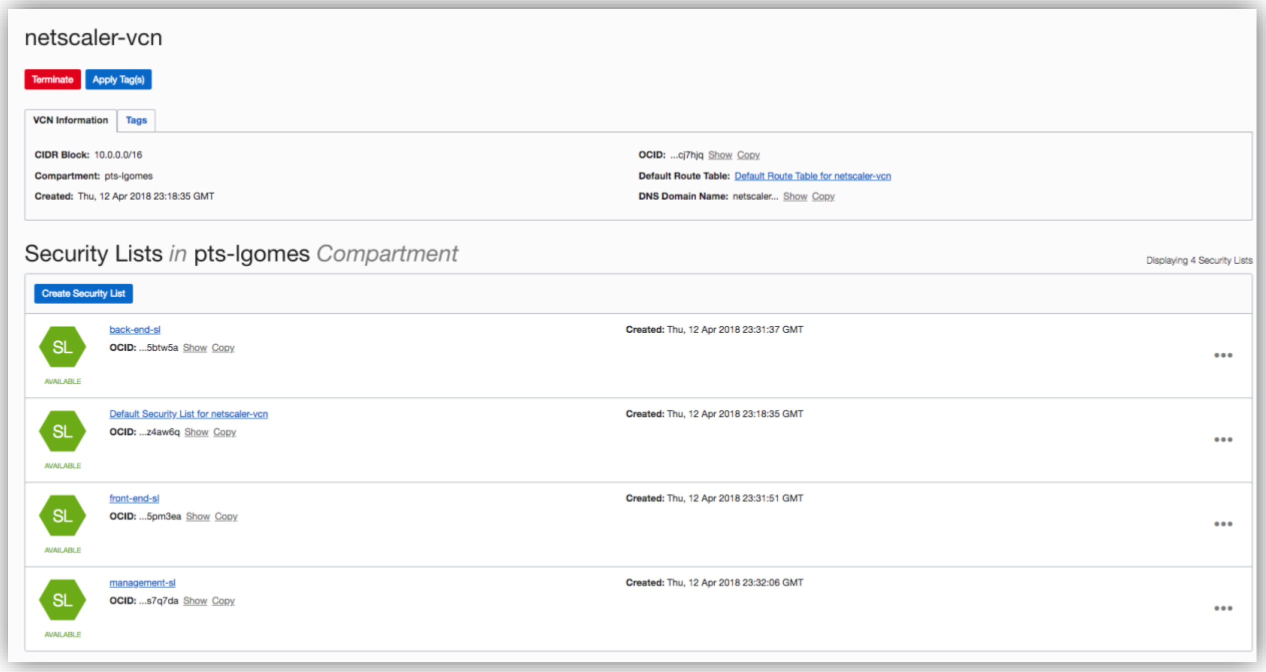
Stateless	Source CIDR	IP Protocol	Source Port Range	Destination Port Range
unchecked	0.0.0.0/0	SSH (TCP/22)	-	22
unchecked	10.0.0.0/16	All Protocols		

Under Allow Rules for Egress, enter the following values:

Stateless	Source CIDR	IP Protocol
unchecked	0.0.0.0/0	All Protocols

Note: You can modify all the security lists accordingly to the customer requirements/needs, e.g. only allow access from one particular IP/range from a trusted source to access management interface, etc.

At the end you will have the following Security Lists created:



netscaler-vcn

CIDR Block: 10.0.0.0/16

Compartment: pts-lgomes

Created: Thu, 12 Apr 2018 23:18:35 GMT

OCID: ...cj7hjq Show Copy

Default Route Table: Default Route Table for netscaler-vcn

DNS Domain Name: netscaler... Show Copy

Security Lists *in* pts-lgomes *Compartment*

Displaying 4 Security Lists

SL AVAILABLE
back-end-sl
OCID: ...5btw5a Show Copy
Created: Thu, 12 Apr 2018 23:31:37 GMT

SL AVAILABLE
Default Security List for netscaler-vcn
OCID: ...z4aw6q Show Copy
Created: Thu, 12 Apr 2018 23:18:35 GMT

SL AVAILABLE
front-end-sl
OCID: ...5pm3ea Show Copy
Created: Thu, 12 Apr 2018 23:31:51 GMT

SL AVAILABLE
management-sl
OCID: ...s7q7da Show Copy
Created: Thu, 12 Apr 2018 23:32:06 GMT

Create FrontEnd, BackEnd & Management Route Tables

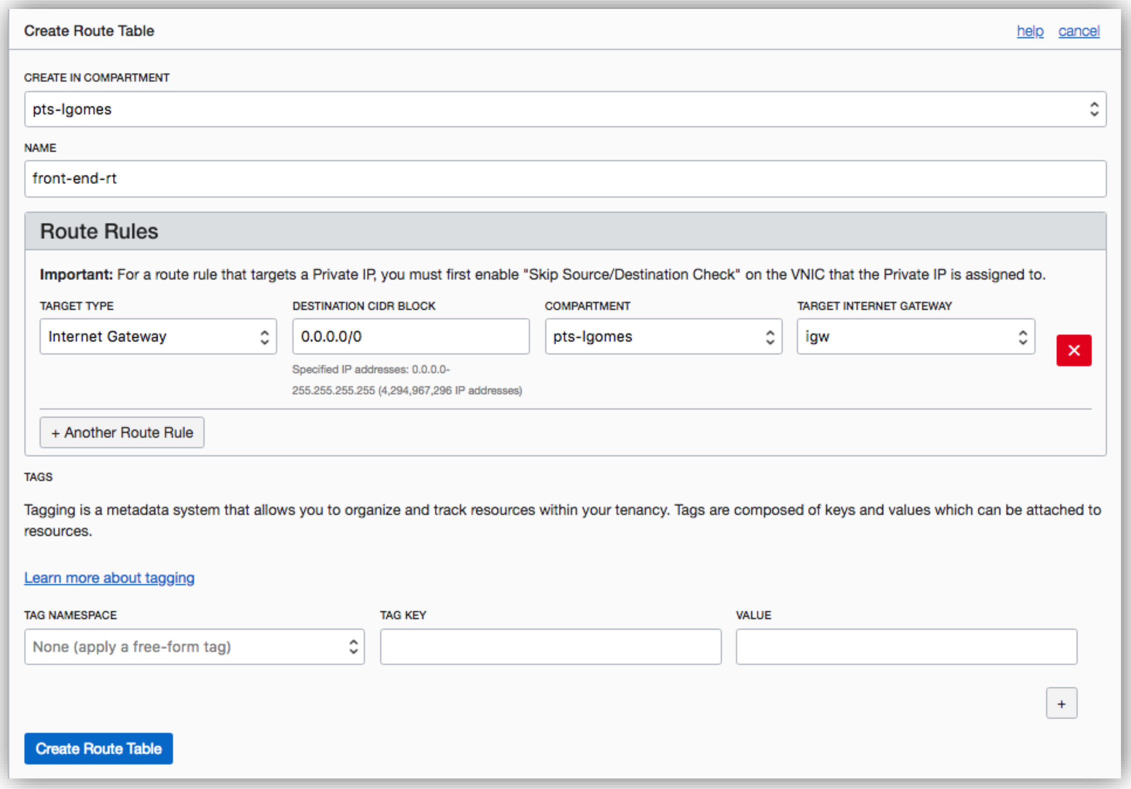
Create FrontEnd Route Table

- Click on the VCN link “netscaler-vcn”
- Click on Route Tables on the left hand side
- Click on Create Route Table

- Enter the following to Create the FrontEnd Route Table

  Compartment: In the default VCN Compartment.
  Name: front-end-rt
  Under Route Rules, enter the following values:

Destination CIDR Block	Target Type	Compartment	Target Internet Gateway
0.0.0.0/0	Internet Gateway	<Default VCN compartment>	Select <DefaultInternetGateway>



Create Route Table

CREATE IN COMPARTMENT: pts-lgomes

NAME: front-end-rt

Route Rules

Important: For a route rule that targets a Private IP, you must first enable "Skip Source/Destination Check" on the VNIC that the Private IP is assigned to.

TARGET TYPE	DESTINATION CIDR BLOCK	COMPARTMENT	TARGET INTERNET GATEWAY
Internet Gateway	0.0.0.0/0	pts-lgomes	igw

Specified IP addresses: 0.0.0.0-255.255.255.255 (4,294,967,296 IP addresses)

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

Create BackEnd Route Table:

- Click on Create Route Table
- Enter the following to Create the BackEnd Route Table

  Compartment: In the default VCN Compartment.
  Name: back-end-rt

Under Route Rules, delete the existing row. NetScaler will be responsible for routing the packages from back-end to front-end/public internet.

Create Management Route Table

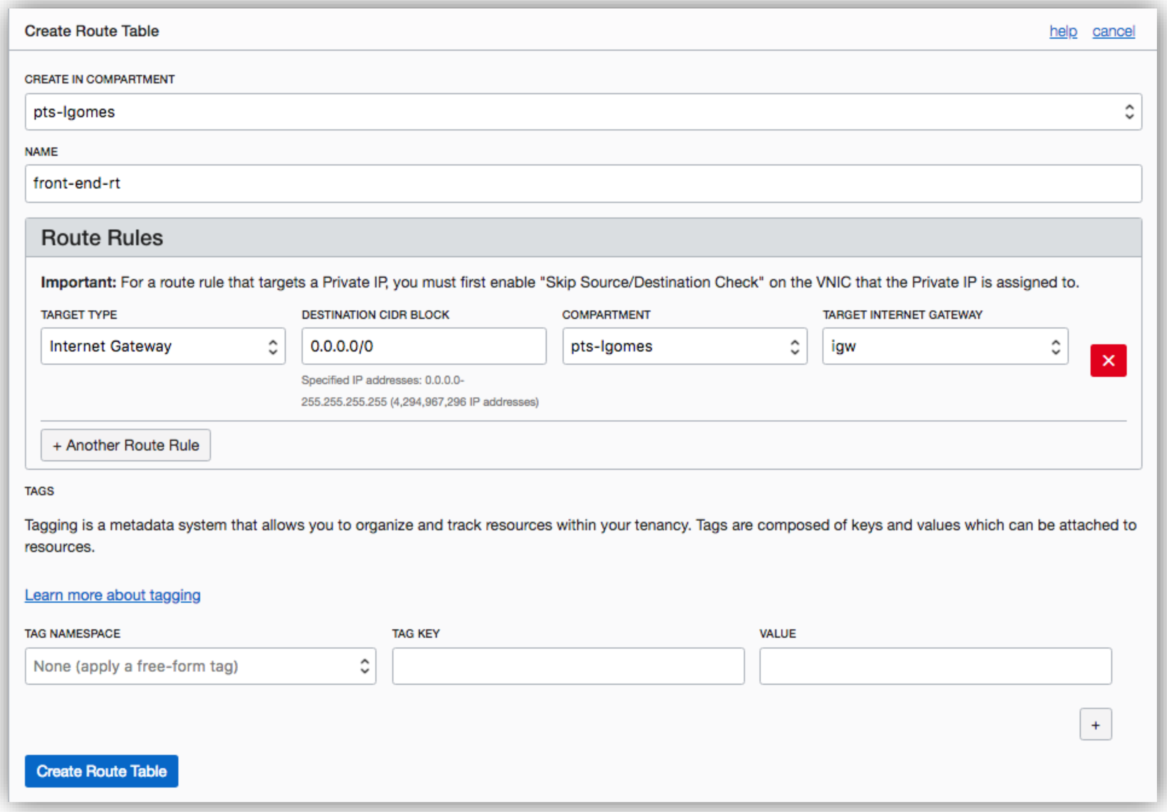
- Click on the VCN link "netscaler-vcn"
- Click on Route Tables on the left hand side
- Click on Create Route Table

- Enter the following to Create the Management Route Table

  Compartment: In the default VCN Compartment.
  Name: mgmt-rt
  Under Route Rules, enter the following values:

Destination CIDR Block	Target Type	Compartment	Target Internet Gateway
0.0.0.0/0	Internet Gateway	<Default VCN compartment>	Select <DefaultInternetGateway>

At the end you will have the following Route Tables created on your environment:



Create Route Table

CREATE IN COMPARTMENT: pts-lgomes

NAME: front-end-rt

Route Rules

Important: For a route rule that targets a Private IP, you must first enable "Skip Source/Destination Check" on the VNIC that the Private IP is assigned to.

TARGET TYPE: Internet Gateway
DESTINATION CIDR BLOCK: 0.0.0.0/0
Specified IP addresses: 0.0.0.0-255.255.255.255 (4,294,967,296 IP addresses)
COMPARTMENT: pts-lgomes
TARGET INTERNET GATEWAY: igw

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE: None (apply a free-form tag)
TAG KEY:
VALUE:

Create FrontEnd, BackEnd, Management & KVMHost Subnets

Create **FrontEnd Subnet**

- Click on **Create Subnet**
- Enter the Following for creating the Subnet:

  Name: front-end

  Availability Domain: US-ASHBURN-AD-1 (or pick up another according to the region you are deploying your solution)

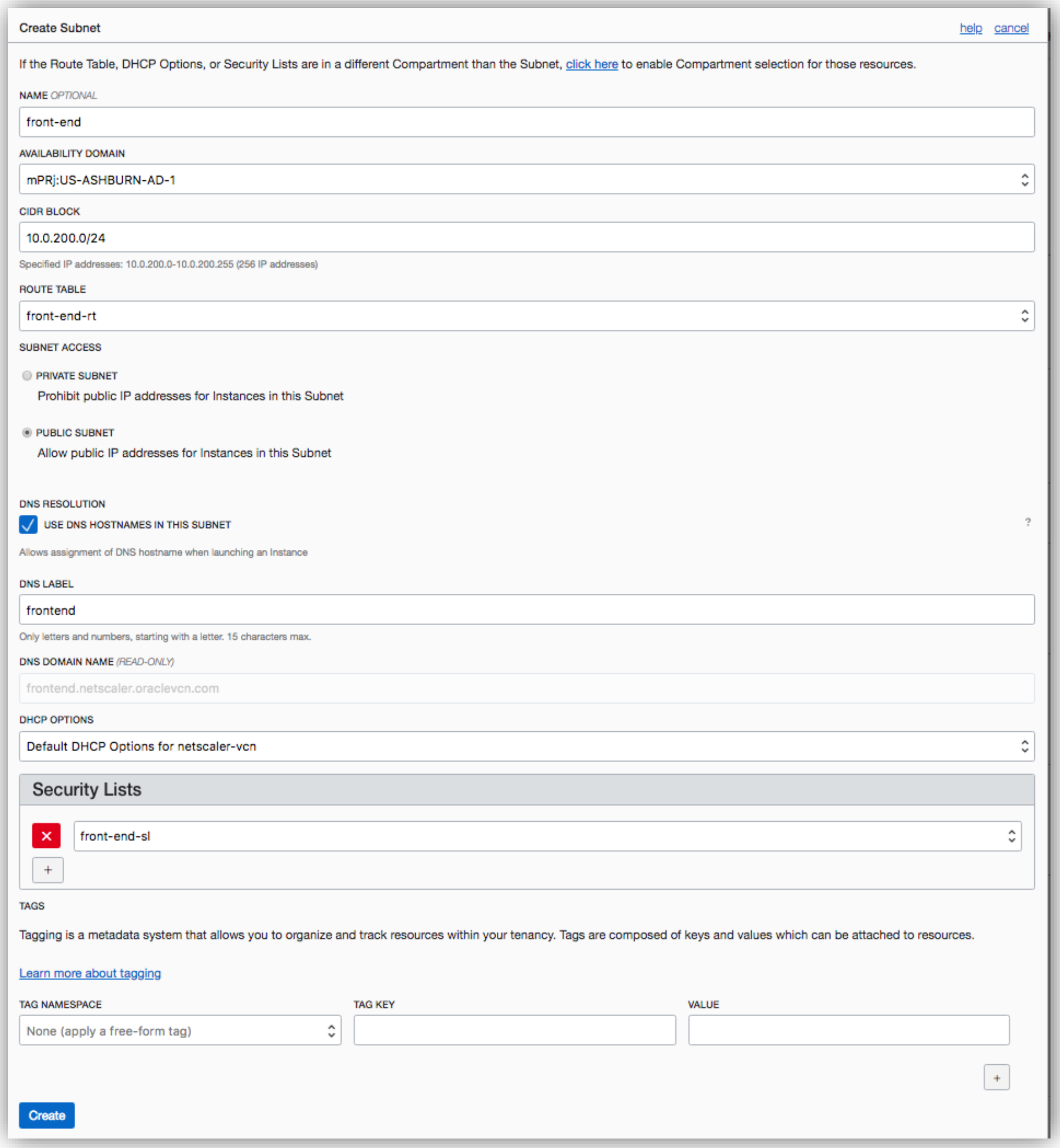
  CIDR Block: A single, contiguous CIDR block for the cloud network. For example: 10.0.200.0/24

  DHCP Options: Default DHCP Options for netscaler-vcn

  Route Table: <Select front-end-rt>

  Subnet Access: PUBLIC Subnet

  Security Lists: <Select front-end-sec-list>



Create Subnet

If the Route Table, DHCP Options, or Security Lists are in a different Compartment than the Subnet, click here to enable Compartment selection for those resources.

NAME *OPTIONAL*
front-end

AVAILABILITY DOMAIN
mPRj:US-ASHBURN-AD-1

CIDR BLOCK
10.0.200.0/24
Specified IP addresses: 10.0.200.0-10.0.200.255 (256 IP addresses)

ROUTE TABLE
front-end-rt

SUBNET ACCESS

○ PRIVATE SUBNET
Prohibit public IP addresses for Instances in this Subnet

● PUBLIC SUBNET
Allow public IP addresses for Instances in this Subnet

DNS RESOLUTION
☑ USE DNS HOSTNAMES IN THIS SUBNET
Allows assignment of DNS hostname when launching an Instance

DNS LABEL
frontend
Only letters and numbers, starting with a letter. 15 characters max.

DNS DOMAIN NAME *(READ-ONLY)*
frontend.netscaler.oraclevcn.com

DHCP OPTIONS
Default DHCP Options for netscaler-vcn

Security Lists

front-end-sl

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

Create **BackEnd Subnet**

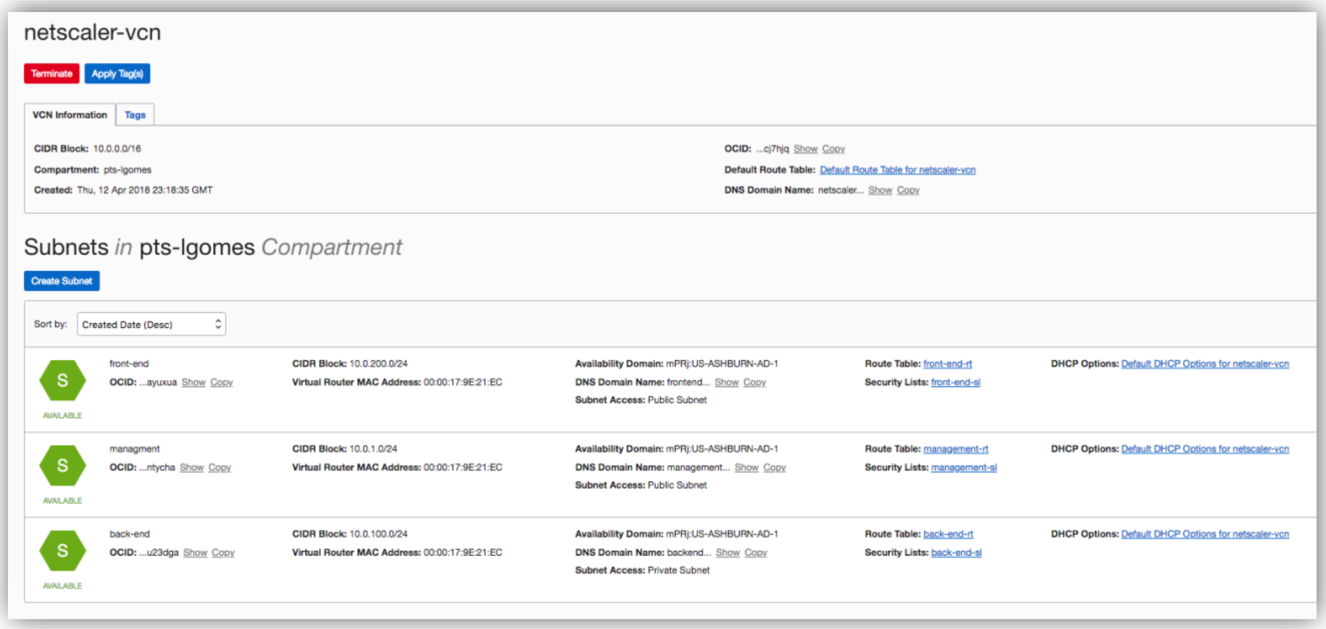
- Enter the Following for creating the Subnet:
  Name: back-end
  Availability Domain: US-ASHBURN-AD-1 (or pick up another according to the region you are deploying your solution)
  CIDR Block: A single, contiguous CIDR block for the cloud network. For example: 10.0.100.0/24
  DHCP Options: Default DHCP Options for netscaler-vcn
  Route Table: <Select back-end-rt>
  Subnet Access: PRIVATE Subnet
  Security Lists: <Select back-end-sec-list>

Create **Management Subnet** which should be used for managing the KVM host.

- Enter the Following for creating the Subnet:

  Name: mgmt-subnet
  Availability Domain: US-ASHBURN-AD-1 (or pick up another according to the region you are deploying your solution)
  CIDR Block: A single, contiguous CIDR block for the cloud network. For example: 10.0.1.0/24
  DHCP Options: Default DHCP Options for netscaler-vcn
  Route Table: <Select mgmt-rt>
  Subnet Access: PUBLIC Subnet
  Security Lists: <Select mgmt-sec-list>

At the end you will have the following Subnets created:



CIDR Block: 10.0.0.0/16
Compartment: pts-lgomes
Created: Thu, 12 Apr 2018 23:18:35 GMT

OCID: ...cj7hjq Show Copy
Default Route Table: Default Route Table for netscaler-vcn
DNS Domain Name: netscaler... Show Copy

Sort by: Created Date (Desc)

front-end
OCID: ...ayuxua Show Copy
CIDR Block: 10.0.200.0/24
Virtual Router MAC Address: 00:00:17:9E:21:EC
Availability Domain: mPRj:US-ASHBURN-AD-1
DNS Domain Name: frontend... Show Copy
Subnet Access: Public Subnet
Route Table: front-end-rt
Security Lists: front-end-sl
DHCP Options: Default DHCP Options for netscaler-vcn

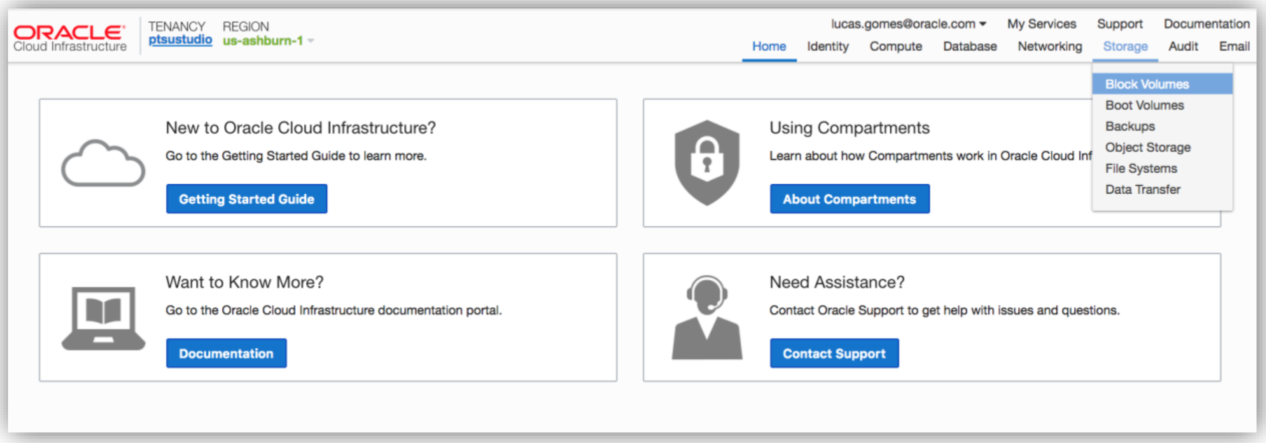
managment
OCID: ...ntycha Show Copy
CIDR Block: 10.0.1.0/24
Virtual Router MAC Address: 00:00:17:9E:21:EC
Availability Domain: mPRj:US-ASHBURN-AD-1
DNS Domain Name: management... Show Copy
Subnet Access: Public Subnet
Route Table: management-rt
Security Lists: management-sl
DHCP Options: Default DHCP Options for netscaler-vcn

back-end
OCID: ...u23dga Show Copy
CIDR Block: 10.0.100.0/24
Virtual Router MAC Address: 00:00:17:9E:21:EC
Availability Domain: mPRj:US-ASHBURN-AD-1
DNS Domain Name: backend... Show Copy
Subnet Access: Private Subnet
Route Table: back-end-rt
Security Lists: back-end-sl
DHCP Options: Default DHCP Options for netscaler-vcn

Create a Block Volume to hold the NetScaler VPX guest

You can either create a Block volume to hold the guest image data in case you select a BM.Standard compute shape or save the NetScaler data direct into a local NVMe disk in case you select a DenseIO shape. In the latter case, you should be responsible to protect the data by following the process described here since the disks are not protected against failure by default.

In order to create a new Block Volume, Click on Block Volumes under Storage:



New to Oracle Cloud Infrastructure?

Go to the Getting Started Guide to learn more.

Using Compartments

Learn about how Compartments work in Oracle Cloud Inf

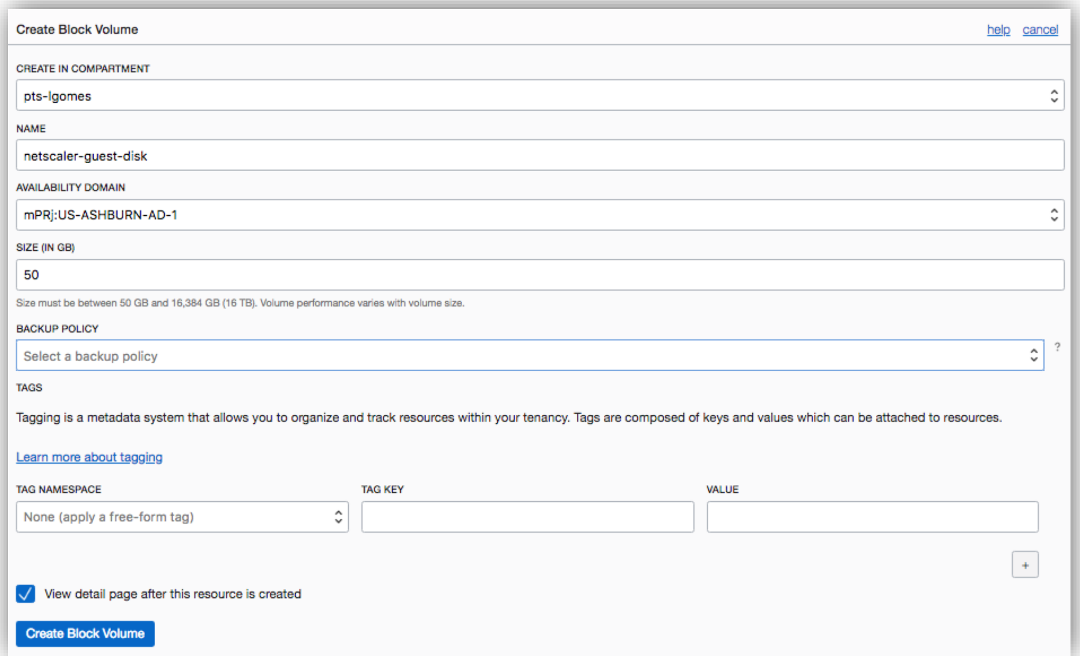
Want to Know More?

Go to the Oracle Cloud Infrastructure documentation portal.

Need Assistance?

Contact Oracle Support to get help with issues and questions.

- Click on **Create Block Volume**
- Enter the following to launch the OCI bare metal instance
  Name : netscaler-guest-disk
  Availability Domain: US-ASHBURN-AD-1 (or pick up another according to the region you are deploying your solution)
  Size (in GB): 50
  Backup Policy: <Select the most appropriated according to customer requirement>



Create Block Volume

CREATE IN COMPARTMENT: pts-lgomes

NAME: netscaler-guest-disk

AVAILABILITY DOMAIN: mPRj:US-ASHBURN-AD-1

SIZE (IN GB): 50

Size must be between 50 GB and 16,384 GB (16 TB). Volume performance varies with volume size.

BACKUP POLICY: Select a backup policy

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

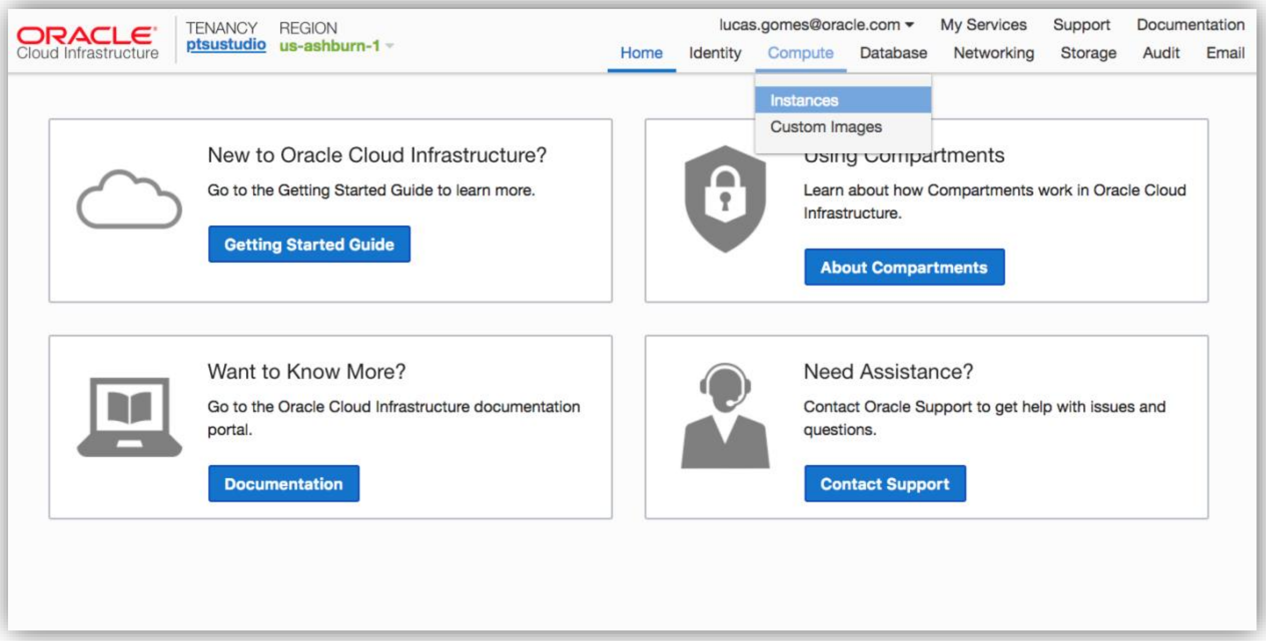
Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

☑ View detail page after this resource is created

Launch the Bare metal instance

Click on Instances under Compute



New to Oracle Cloud Infrastructure?

Go to the Getting Started Guide to learn more.

Using Compartments

Learn about how Compartments work in Oracle Cloud Infrastructure.

Want to Know More?

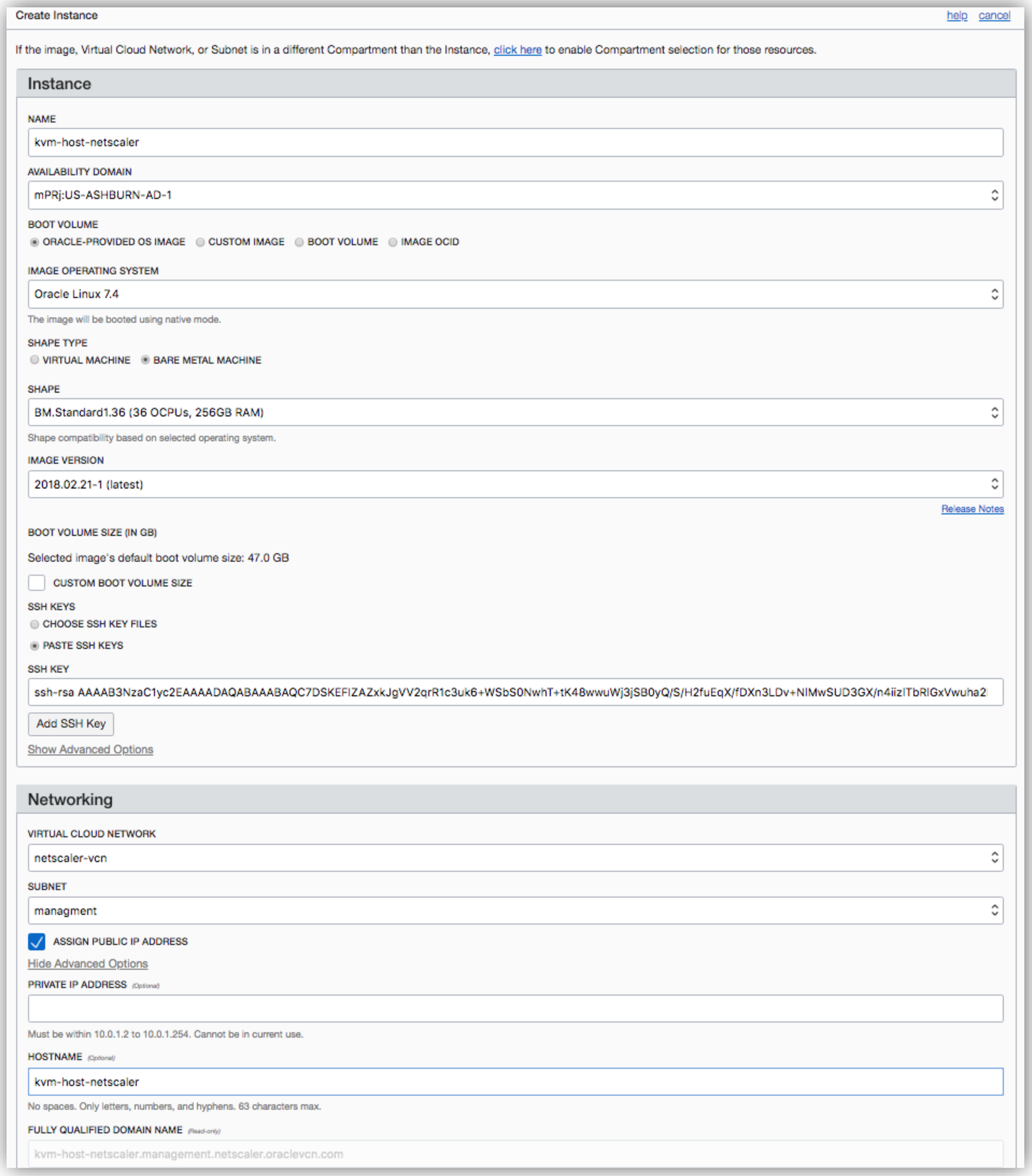
Go to the Oracle Cloud Infrastructure documentation portal.

Need Assistance?

Contact Oracle Support to get help with issues and questions.

- Enter the following to launch the OCI bare metal instance
  Name : kvm-host-netscaler
  Availability Domain: : US-ASHBURN-AD-1 (or pick up another according to the region you are deploying your solution)
  Image Source: ORACLE-PROVIDED OS IMAGE
  Image: Oracle Linux 7.x
  Shape Type: Bare Metal Machine
  Shape: BM.Standard1.36 or BM.DenseIO1.36
  Image Build: latest
  VCN: netscaler-vcn
  Subnet: management
  Assigned Public IP : Checked
  Hostname: kvm-host-netscaler
  SSH keys: Provide the public ssh keys to access the intance

The BM instance will be created within the specified VCN and subnet with an assigned Public IP



Create Instance

If the image, Virtual Cloud Network, or Subnet is in a different Compartment than the Instance, click here to enable Compartment selection for those resources.

Instance

NAME

kvm-host-netscaler

AVAILABILITY DOMAIN

mPRj:US-ASHBURN-AD-1

BOOT VOLUME

● ORACLE-PROVIDED OS IMAGE ○ CUSTOM IMAGE ○ BOOT VOLUME ○ IMAGE OCID

IMAGE OPERATING SYSTEM

Oracle Linux 7.4

The image will be booted using native mode.

SHAPE TYPE

○ VIRTUAL MACHINE ● BARE METAL MACHINE

SHAPE

BM.Standard1.36 (36 OCPUs, 256GB RAM)

Shape compatibility based on selected operating system.

IMAGE VERSION

2018.02.21-1 (latest)

BOOT VOLUME SIZE (IN GB)

Selected image's default boot volume size: 47.0 GB

□ CUSTOM BOOT VOLUME SIZE

SSH KEYS

○ CHOOSE SSH KEY FILES

● PASTE SSH KEYS

SSH KEY

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC7DSKEFIZAZxkJgVV2qrR1c3uk6+WSbS0NwhT+tK48wwuWj3jSB0yQ/S/H2fuEqX/fDXn3LDv+NIMwSUD3GX/n4iizITbRIGxVwuha2I

Networking

VIRTUAL CLOUD NETWORK

netscaler-vcn

SUBNET

managment

☑ ASSIGN PUBLIC IP ADDRESS

PRIVATE IP ADDRESS (Optional)

Must be within 10.0.1.2 to 10.0.1.254. Cannot be in current use.

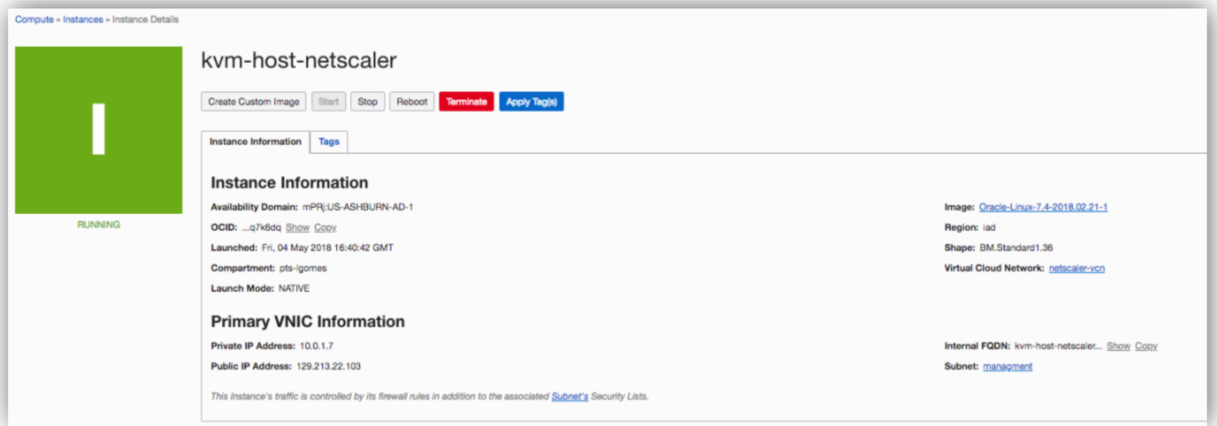
HOSTNAME (Optional)

kvm-host-netscaler

No spaces. Only letters, numbers, and hyphens. 63 characters max.

FULLY QUALIFIED DOMAIN NAME (Read-only)

kvm-host-netscaler.management.netscaler.oraclevcn.com



kvm-host-netscaler

Instance Information

Availability Domain: mPRj:US-ASHBURN-AD-1

OCID: ...q7k6dq Show Copy

Launched: Fri, 04 May 2018 16:40:42 GMT

Compartment: pts-lgomes

Launch Mode: NATIVE

Image: Oracle-Linux-7.4-2018.02.21-1

Region: iad

Shape: BM.Standard1.36

Virtual Cloud Network: netscaler-vcn

Primary VNIC Information

Private IP Address: 10.0.1.7

Public IP Address: 129.213.22.103

Internal FQDN: kvm-host-netscaler... Show Copy

Subnet: managment

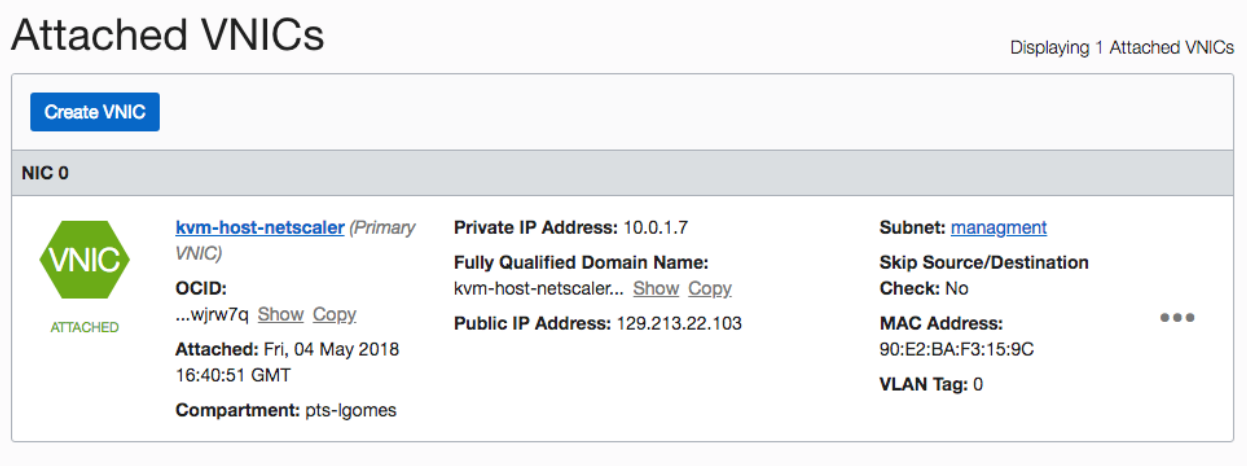
This Instance's traffic is controlled by its firewall rules in addition to the associated Subnet's Security Lists.

Attach Secondary Virtual Network Interface Cards (VNICs)

A VNIC enables an instance to connect to a VCN and determines how the instance connects with endpoints inside and outside the VCN. Each VNIC resides in a subnet in a VCN and includes these items:

- One primary private IPv4 address from the subnet the VNIC is in, chosen by either you or Oracle.
- Up to 31 optional secondary private IPv4 addresses from the same subnet the VNIC is in, chosen by either you or Oracle.
- An optional public IPv4 address for each private IP, chosen by Oracle but assigned by you at your discretion.
- An optional hostname for DNS for each private IP address (see DNS in Your Virtual Cloud Network).
- A MAC address.
- A VLAN tag assigned by Oracle and available when attachment of the VNIC to the instance is complete (relevant only for bare metal instances).
- A flag to enable or disable the source/destination check on the VNIC's network traffic (see Source/Destination Check).

Note that the default primary VNIC of the kvm-host is attached to the management subnet and is created automatically when we launched the instance:



Attached VNICs

Displaying 1 Attached VNICs

NIC 0

kvm-host-netscaler *(Primary VNIC)*

OCID: ...wjrw7q Show Copy

Attached: Fri, 04 May 2018 16:40:51 GMT

Compartment: pts-lgomes

Private IP Address: 10.0.1.7

Fully Qualified Domain Name: kvm-host-netscaler... Show Copy

Public IP Address: 129.213.22.103

Subnet: managment

Skip Source/Destination Check: No

MAC Address: 90:E2:BA:F3:15:9C

VLAN Tag: 0

Now you should create secondary VNICs and attach secondary IPs that will be attached to the Guest VM (NetScaler VPX). These vnics should connect to FrontEnd and BackEnd subnets, according to the table below:

VNIC Name	Subnet	Hostname	Secondary IP hostname
vnic2	front-end	nsip	vip-gateway
vnic3	back-end	snip	vip-lb

Create **vnic2 for NetScaler VM (Guest) that will be used as NSIP.**

- Go to **Instance Details** page
- Click on Attached VNICs link on the left hand side
- Click on **Create VNIC**
- Enter the Following for creating the VNIC:

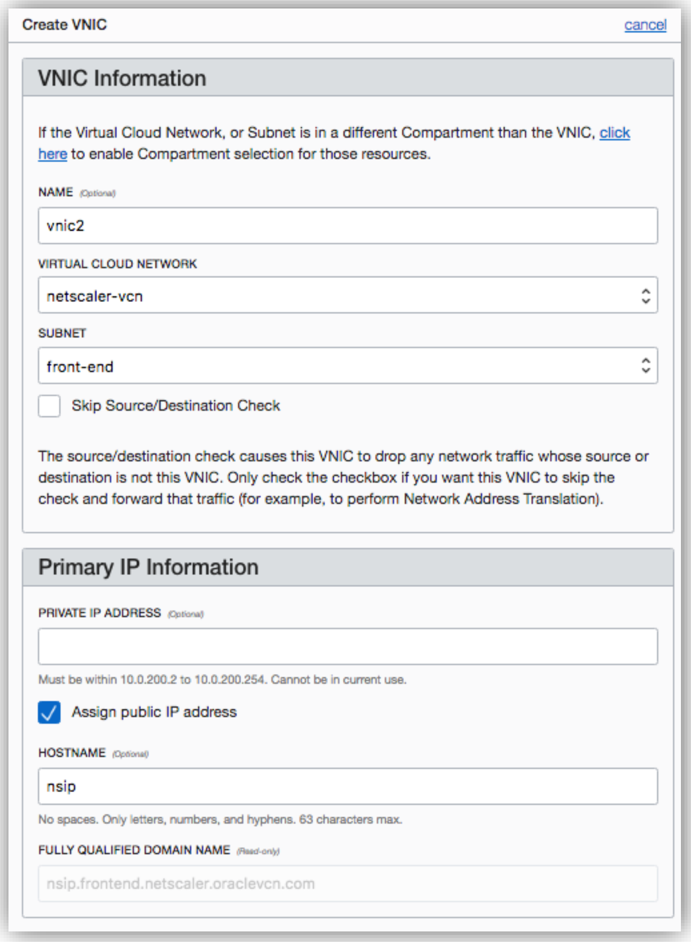
  Name: vnic2

  Virtual Cloud Network: <Select netscaler-vcn VCN>

  Subnet: <Select front-end Subnet>

  Assign public IP Address: **Checked**

  Hostname: nsip



Create VNIC

VNIC Information

If the Virtual Cloud Network, or Subnet is in a different Compartment than the VNIC, click here to enable Compartment selection for those resources.

NAME (Optional): vnic2

VIRTUAL CLOUD NETWORK: netscaler-vcn

SUBNET: front-end

□ Skip Source/Destination Check

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).

Primary IP Information

PRIVATE IP ADDRESS (Optional):

Must be within 10.0.200.2 to 10.0.200.254. Cannot be in current use.

☑ Assign public IP address

HOSTNAME (Optional): nsip

No spaces. Only letters, numbers, and hyphens. 63 characters max.

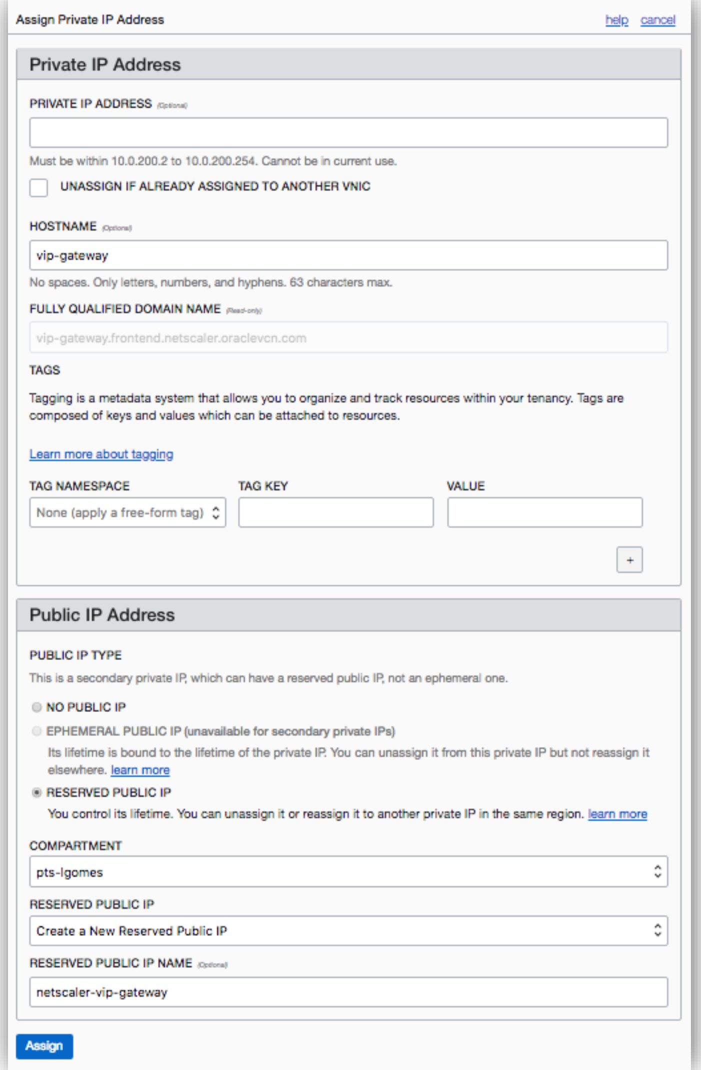
FULLY QUALIFIED DOMAIN NAME (Read-only): nsip.frontend.netscaler.oraclevcn.com

Create a Secondary IP Address for the VIP (Gateway):

- Go to **Instance Details** page
- Click on Attached VNICs link on the left hand side

- Click on **vnic2**
- Click on Assign Private IP Address
- Enter the Following for creating the Private IP Address:

  Hostname: **vip-gateway**
- Enter the Following for creating the Public IP Address:
  - Select Reserved Public IP
  - If you don't have one already in place, you can select <Create a New Reserved Public IP> selector and then enter a Reserved Public IP Name: vip-gateway



Assign Private IP Address

Private IP Address

PRIVATE IP ADDRESS (Optional)

Must be within 10.0.200.2 to 10.0.200.254. Cannot be in current use.

□ UNASSIGN IF ALREADY ASSIGNED TO ANOTHER VNIC

HOSTNAME (Optional): vip-gateway

No spaces. Only letters, numbers, and hyphens. 63 characters max.

FULLY QUALIFIED DOMAIN NAME (Read-only): vip-gateway.frontend.netscaler.oraclevcn.com

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

Public IP Address

PUBLIC IP TYPE

This is a secondary private IP, which can have a reserved public IP, not an ephemeral one.

○ NO PUBLIC IP

○ EPHEMERAL PUBLIC IP (unavailable for secondary private IPs)

Its lifetime is bound to the lifetime of the private IP. You can unassign it from this private IP but not reassign it elsewhere. learn more

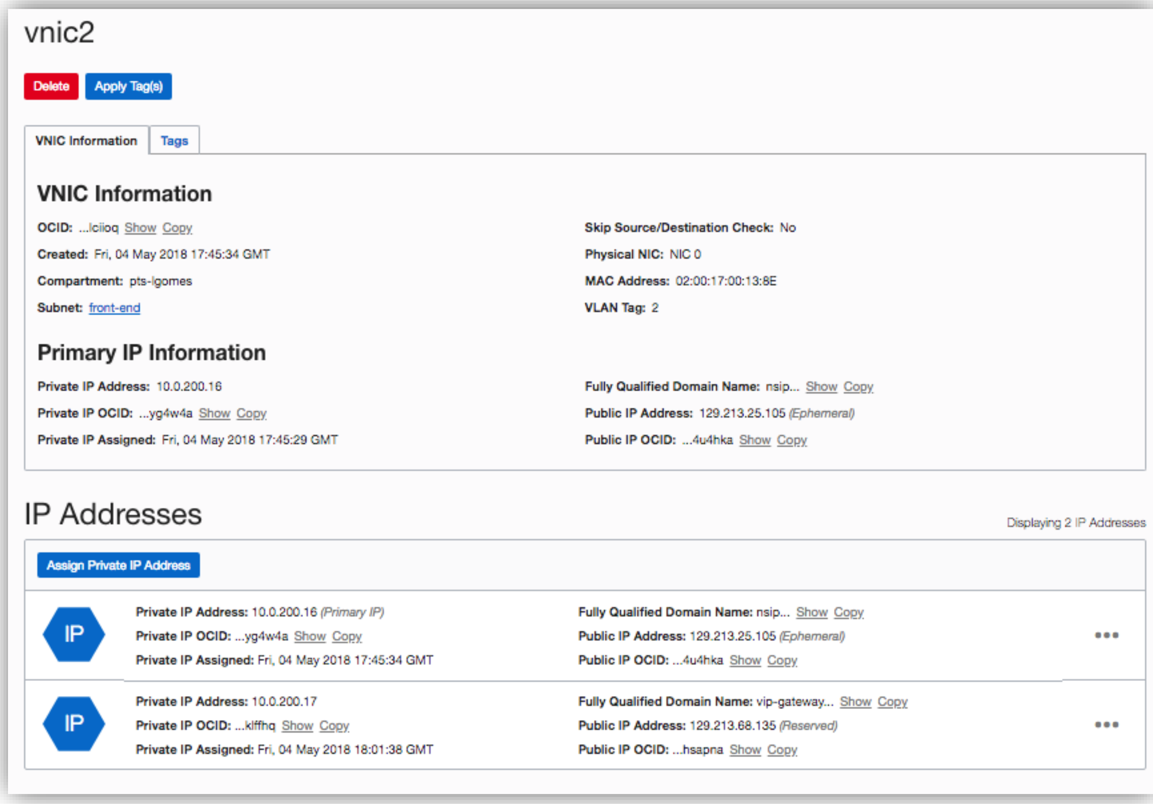
● RESERVED PUBLIC IP

You control its lifetime. You can unassign it or reassign it to another private IP in the same region. learn more

COMPARTMENT: pts-lgomes

RESERVED PUBLIC IP: Create a New Reserved Public IP

RESERVED PUBLIC IP NAME (Optional): netscaler-vip-gateway



VNIC Information

OCID: ...lciioq Show Copy
Created: Fri, 04 May 2018 17:45:34 GMT
Compartment: pts-lgomes
Subnet: front-end

Skip Source/Destination Check: No
Physical NIC: NIC 0
MAC Address: 02:00:17:00:13:8E
VLAN Tag: 2

Primary IP Information

Private IP Address: 10.0.200.16
Private IP OCID: ...yg4w4a Show Copy
Private IP Assigned: Fri, 04 May 2018 17:45:29 GMT

Fully Qualified Domain Name: nsip... Show Copy
Public IP Address: 129.213.25.105 *(Ephemeral)*
Public IP OCID: ...4u4hka Show Copy

IP Addresses

Displaying 2 IP Addresses

Private IP Address: 10.0.200.16 *(Primary IP)*
Private IP OCID: ...yg4w4a Show Copy
Private IP Assigned: Fri, 04 May 2018 17:45:34 GMT

Fully Qualified Domain Name: nsip... Show Copy
Public IP Address: 129.213.25.105 *(Ephemeral)*
Public IP OCID: ...4u4hka Show Copy

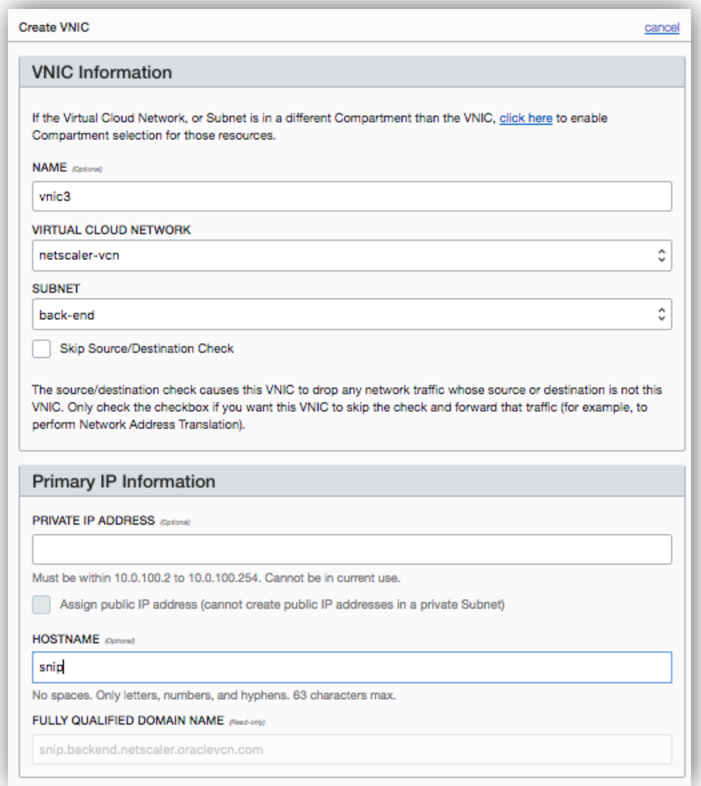
Private IP Address: 10.0.200.17
Private IP OCID: ...klffhq Show Copy
Private IP Assigned: Fri, 04 May 2018 18:01:38 GMT

Fully Qualified Domain Name: vip-gateway... Show Copy
Public IP Address: 129.213.68.135 *(Reserved)*
Public IP OCID: ...hsapna Show Copy

Create **vnic3 for NetScaler VM (Guest) that will be used as SNIP**

- Go to **Instance Details** page
- Click on Attached VNICs link on the left hand side
- Click on **Create VNIC**
- Enter the Following for creating the VNIC:

  Name: vnic3
  Virtual Cloud Network: <Select netscaler-vcn VCN>
  Subnet: <Select back-end Subnet>
  Hostname: snip



Create VNIC

VNIC Information

If the Virtual Cloud Network, or Subnet is in a different Compartment than the VNIC, click here to enable Compartment selection for those resources.

NAME (Optional)

vnic3

VIRTUAL CLOUD NETWORK

netscaler-vcn

SUBNET

back-end

□ Skip Source/Destination Check

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).

Primary IP Information

PRIVATE IP ADDRESS (Optional)

Must be within 10.0.100.2 to 10.0.100.254. Cannot be in current use.

□ Assign public IP address (cannot create public IP addresses in a private Subnet)

HOSTNAME (Optional)

snip

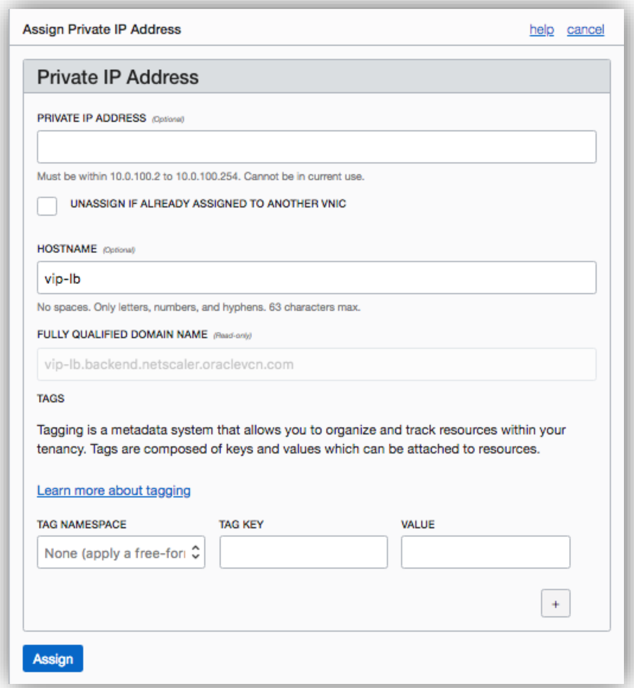
No spaces. Only letters, numbers, and hyphens. 63 characters max.

FULLY QUALIFIED DOMAIN NAME (Read-only)

snip.backend.netscaler.oraclevcn.com

Create a Secondary IP Address for the VIP (Load Balancer):

- Go to **Instance Details** page
- Click on Attached VNICs link on the left hand side
- Click on **vnic3**
- Click on Assign Private IP Address
- Enter the Following for creating the Private IP Address:
  Hostname: **vip-lb**



Assign Private IP Address

Private IP Address

PRIVATE IP ADDRESS *(Optional)*

Must be within 10.0.100.2 to 10.0.100.254. Cannot be in current use.

□ UNASSIGN IF ALREADY ASSIGNED TO ANOTHER VNIC

HOSTNAME *(Optional)*: vip-lb

No spaces. Only letters, numbers, and hyphens. 63 characters max.

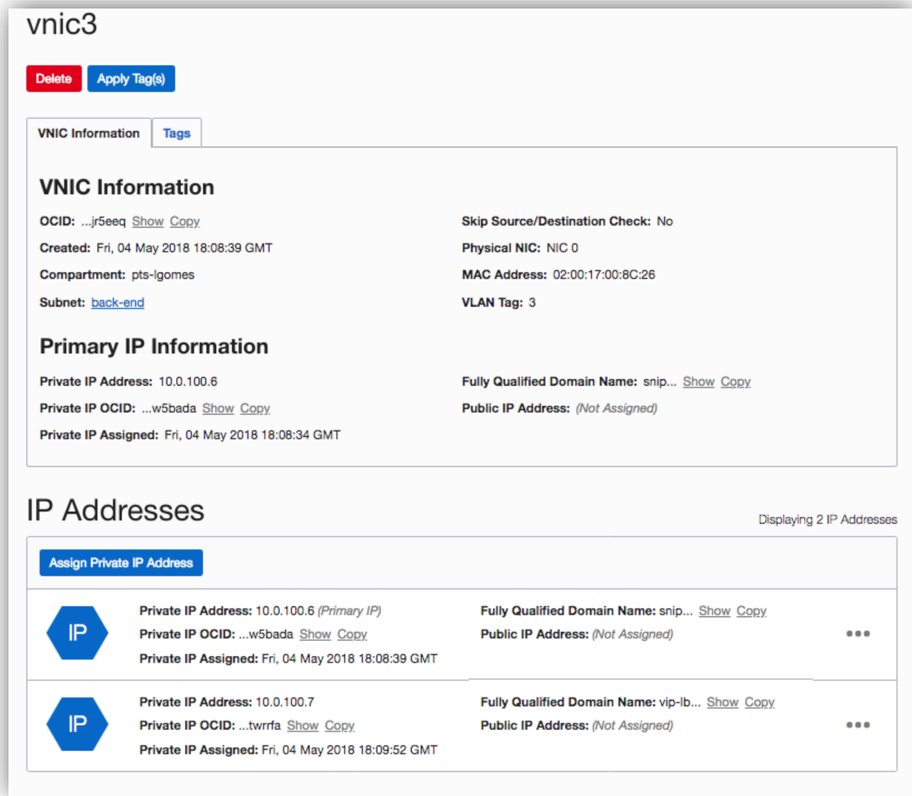
FULLY QUALIFIED DOMAIN NAME *(Read-only)*: vip-lb.backend.netscaler.oraclevcn.com

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

Learn more about tagging

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-fori		



vnic3

VNIC Information

OCID: ...jr5eeq Show Copy
Created: Fri, 04 May 2018 18:08:39 GMT
Compartment: pts-lgomes
Subnet: back-end

Skip Source/Destination Check: No
Physical NIC: NIC 0
MAC Address: 02:00:17:00:8C:26
VLAN Tag: 3

Primary IP Information

Private IP Address: 10.0.100.6
Private IP OCID: ...w5bada Show Copy
Private IP Assigned: Fri, 04 May 2018 18:08:34 GMT

Fully Qualified Domain Name: snip... Show Copy
Public IP Address: *(Not Assigned)*

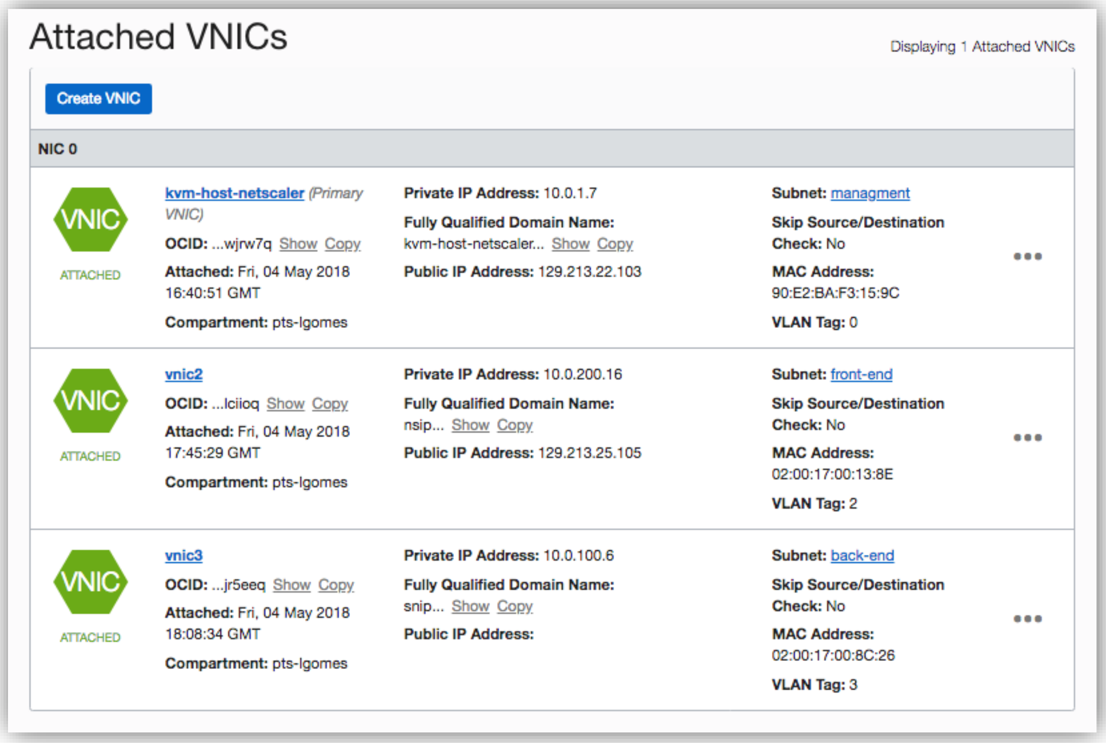
IP Addresses

Displaying 2 IP Addresses

Private IP Address: 10.0.100.6 *(Primary IP)*
Private IP OCID: ...w5bada Show Copy
Private IP Assigned: Fri, 04 May 2018 18:08:39 GMT
Fully Qualified Domain Name: snip... Show Copy
Public IP Address: *(Not Assigned)*

Private IP Address: 10.0.100.7
Private IP OCID: ...twrrfa Show Copy
Private IP Assigned: Fri, 04 May 2018 18:09:52 GMT
Fully Qualified Domain Name: vip-lb... Show Copy
Public IP Address: *(Not Assigned)*

Take notes of all VNICS Private/Public IP Address, MAC Address & VLAN Tag. This information will be used further for setting up the KVM domain.



Attached VNICs

Displaying 1 Attached VNICs

NIC 0

ATTACHED

kvm-host-netscaler *(Primary VNIC)*

OCID: ...wjrw7q Show Copy

Attached: Fri, 04 May 2018 16:40:51 GMT

Compartment: pts-lgomes

Private IP Address: 10.0.1.7

Fully Qualified Domain Name: kvm-host-netscaler... Show Copy

Public IP Address: 129.213.22.103

Subnet: managment

Skip Source/Destination Check: No

MAC Address: 90:E2:BA:F3:15:9C

VLAN Tag: 0

ATTACHED

vnic2

OCID: ...lciioq Show Copy

Attached: Fri, 04 May 2018 17:45:29 GMT

Compartment: pts-lgomes

Private IP Address: 10.0.200.16

Fully Qualified Domain Name: nsip... Show Copy

Public IP Address: 129.213.25.105

Subnet: front-end

Skip Source/Destination Check: No

MAC Address: 02:00:17:00:13:8E

VLAN Tag: 2

ATTACHED

vnic3

OCID: ...jr5eeq Show Copy

Attached: Fri, 04 May 2018 18:08:34 GMT

Compartment: pts-lgomes

Private IP Address: 10.0.100.6

Fully Qualified Domain Name: snip... Show Copy

Public IP Address:

Subnet: back-end

Skip Source/Destination Check: No

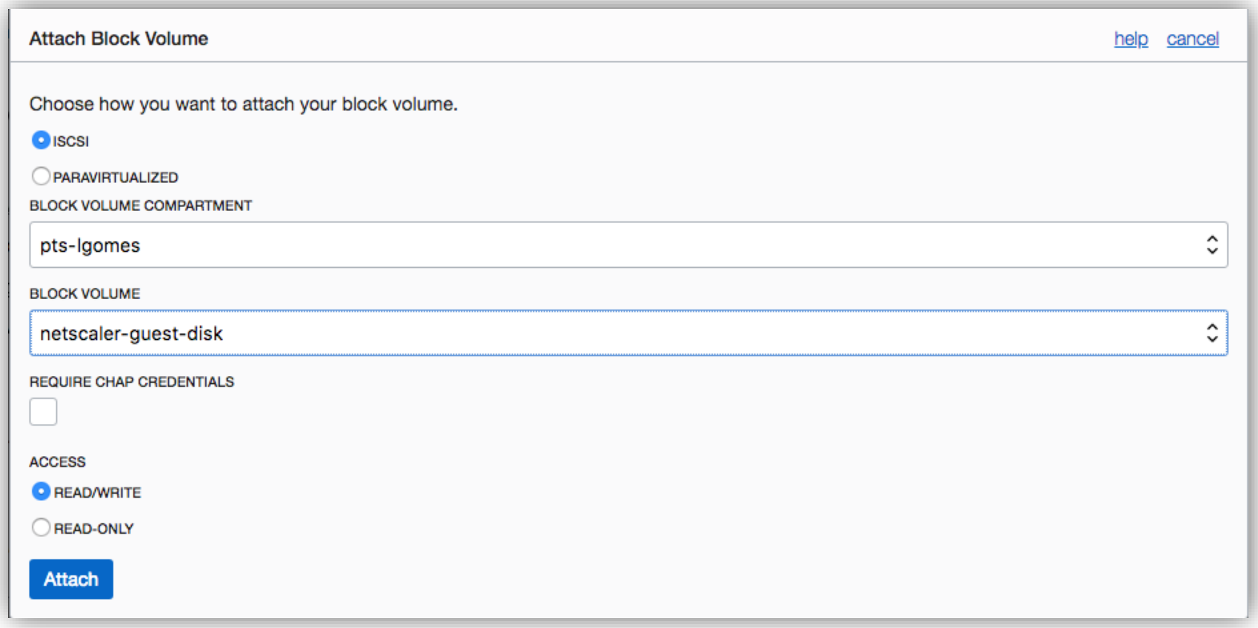
MAC Address: 02:00:17:00:8C:26

VLAN Tag: 3

Attach Block Volume to the KVM host (optional)

- Go to **Instance Details** page
- Click on Attached Block Volumes link on the left hand side
- Click on **Attach Block Volume**
- Enter the Following for attaching the Block Volume:

  Block Volume Compartment: <SelectVCN Compartment>
  Block Volume: <Select netscaler-guest-disk >



Attach Block Volume

Choose how you want to attach your block volume.

● ISCSI
○ PARAVIRTUALIZED

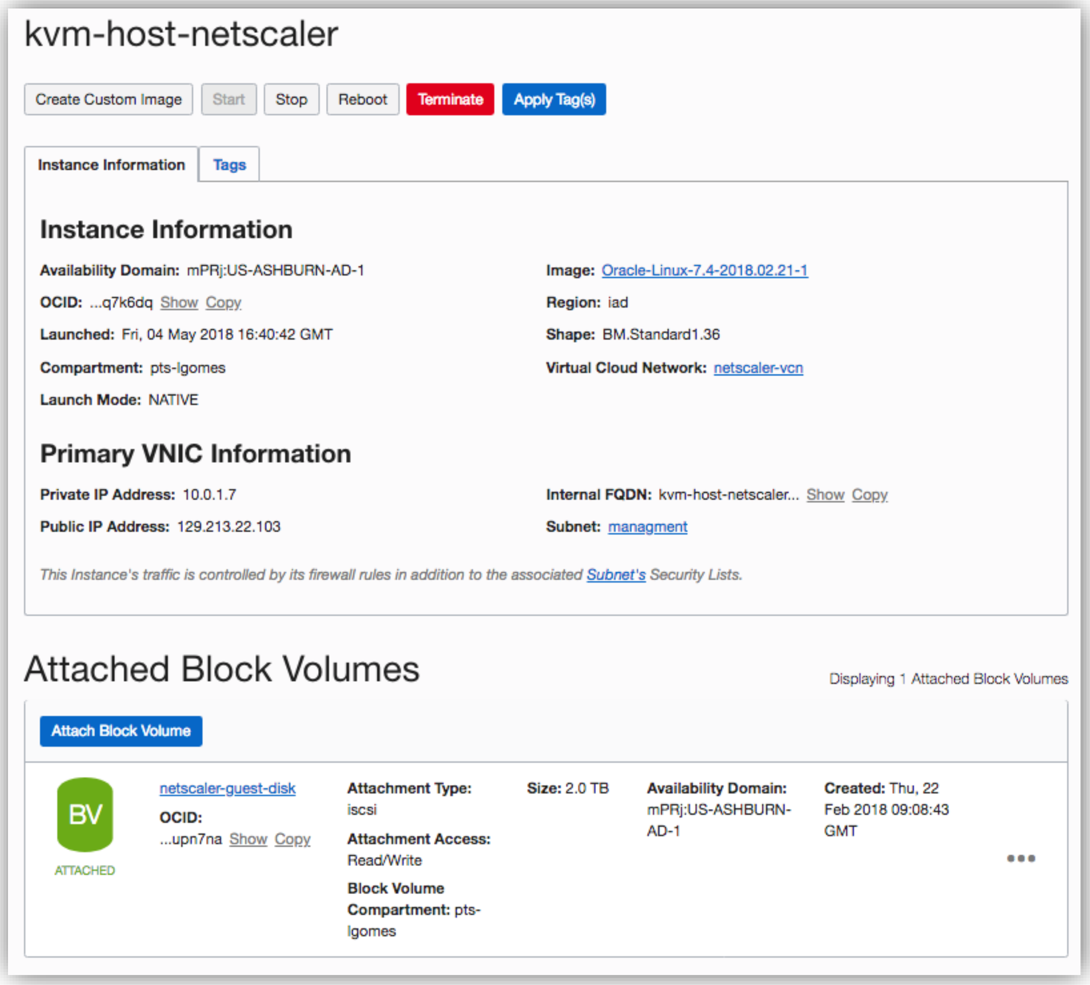
BLOCK VOLUME COMPARTMENT: pts-lgomes

BLOCK VOLUME: netscaler-guest-disk

REQUIRE CHAP CREDENTIALS □

ACCESS
● READ/WRITE
○ READ-ONLY

Disk is now attached to the kvm-host:



kvm-host-netscaler

Instance Information

Availability Domain: mPRj:US-ASHBURN-AD-1

OCID: ...q7k6dq Show Copy

Launched: Fri, 04 May 2018 16:40:42 GMT

Compartment: pts-lgomes

Launch Mode: NATIVE

Image: Oracle-Linux-7.4-2018.02.21-1

Region: iad

Shape: BM.Standard1.36

Virtual Cloud Network: netscaler-vcn

Primary VNIC Information

Private IP Address: 10.0.1.7

Public IP Address: 129.213.22.103

Internal FQDN: kvm-host-netscaler... Show Copy

Subnet: managment

This Instance's traffic is controlled by its firewall rules in addition to the associated Subnet's Security Lists.

Attached Block Volumes

Displaying 1 Attached Block Volumes

netscaler-guest-disk

OCID: ...upn7na Show Copy

Attachment Type: iscsi

Attachment Access: Read/Write

Block Volume Compartment: pts-lgomes

Size: 2.0 TB

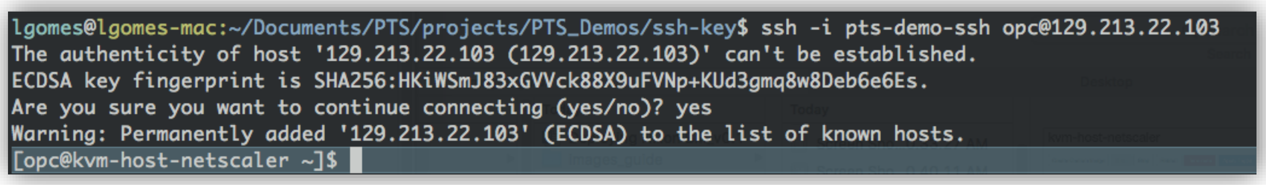
Availability Domain: mPRj:US-ASHBURN-AD-1

Created: Thu, 22 Feb 2018 09:08:43 GMT

Install KVM on Bare Metal Host & Activate VT-d in the kernel

All the next steps were created based on the official OCI Whitepaper available in the documentation: Installing and configuring KVM on Bare Metal Instances with Multi-VNIC:

- Connect to the kvm-host-vnic in the KVM host via SSH (user is opc)

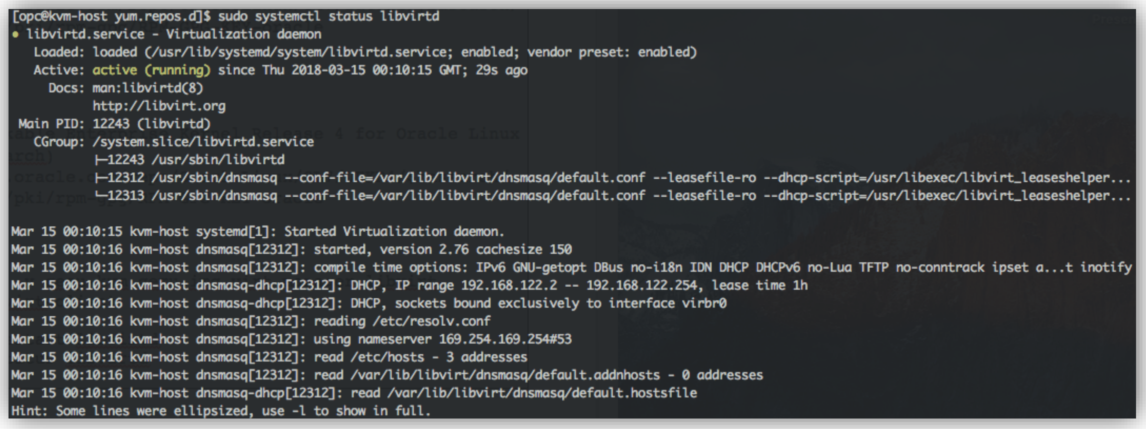


```
lgomes@lgomes-mac:~/Documents/PTS/projects/PTS_Demos/ssh-key$ ssh -i pts-demo-ssh opc@129.213.22.103
The authenticity of host '129.213.22.103 (129.213.22.103)' can't be established.
ECDSA key fingerprint is SHA256:HKiWSmJ83xGVVck88X9uFVNp+KUd3gmq8w8Deb6e6Es.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '129.213.22.103' (ECDSA) to the list of known hosts.
[opc@kvm-host-netscaler ~]$
```

- Update the system and install KVM and other softwares (run with sudo su -). Make sure you have an existing yum repo configured and enabled for UEK4. Then install all the required packages.
  Source: Configure a KVM Host with UEK4:

```
# cd /etc/yum.repos.d/
# wget http://yum.oracle.com/public-yum-ol7.repo
# vim public-yum-ol7.repo
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol7_UEKR4]
name=Latest Unbreakable Enterprise Kernel Release 4 for Oracle Linux
$releasever ($basearch)
baseurl=http://yum.oracle.com/repo/OracleLinux/OL7/UEKR4/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

# yum install -y qemu-kvm qemu-img virt-manager virt-install libvirt
libvirt-python libvirt-client lshw

# systemctl restart libvirtd
# systemctl status libvirtd
```

After restarting libvirtd daemon double check if it's active.



```
[opc@kvm-host yum.repos.d]$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2018-03-15 00:10:15 GMT; 29s ago
     Docs: man:libvirtd(8)
           http://libvirt.org
 Main PID: 12243 (libvirtd)
   CGroup: /system.slice/libvirtd.service
           ├─12243 /usr/sbin/libvirtd
           ├─12312 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper...
           └─12313 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper...

Mar 15 00:10:15 kvm-host systemd[1]: Started Virtualization daemon.
Mar 15 00:10:16 kvm-host dnsmasq[12312]: started, version 2.76 cachesize 150
Mar 15 00:10:16 kvm-host dnsmasq[12312]: compile time options: IPv6 GNU-getopt DBus no-i18n IDN DHCP DHCPv6 no-Lua TFTP no-conntrack ipset a...t inotify
Mar 15 00:10:16 kvm-host dnsmasq-dhcp[12312]: DHCP, IP range 192.168.122.2 -- 192.168.122.254, lease time 1h
Mar 15 00:10:16 kvm-host dnsmasq-dhcp[12312]: DHCP, sockets bound exclusively to interface virbr0
Mar 15 00:10:16 kvm-host dnsmasq[12312]: reading /etc/resolv.conf
Mar 15 00:10:16 kvm-host dnsmasq[12312]: using nameserver 169.254.169.254#53
Mar 15 00:10:16 kvm-host dnsmasq[12312]: read /etc/hosts - 3 addresses
Mar 15 00:10:16 kvm-host dnsmasq[12312]: read /var/lib/libvirt/dnsmasq/default.addnhosts - 0 addresses
Mar 15 00:10:16 kvm-host dnsmasq-dhcp[12312]: read /var/lib/libvirt/dnsmasq/default.hostsfile
Hint: Some lines were ellipsized, use -l to show in full.
```

- Create and run the following script to activate VT-d in KVM, which is used to configure the host for PCI Passthrough. This will add the “intel_iommu=on” line to the end of the GRUB_CMDLINE_LINUX.

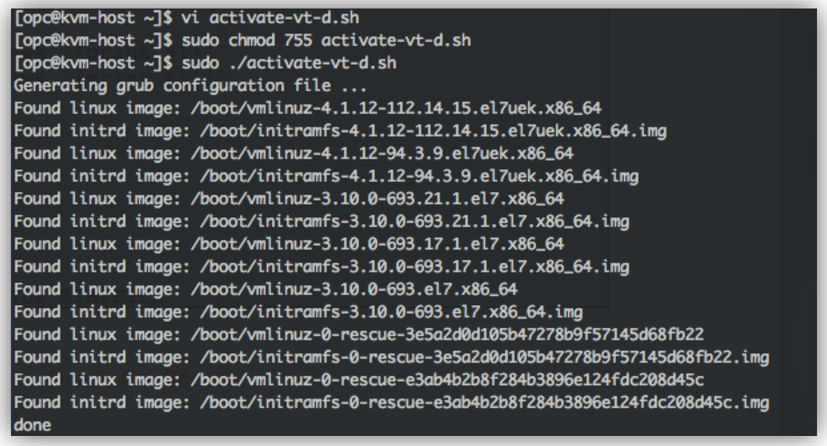
```
cd /home/opc
# vim activate-vt-d.sh
```

Copy and paste the content below into the file:

```
#!/bin/bash

#Modify grub
GRUBFILE=/etc/default/grub
TMPFILE=`mktemp`

sed -e 's/^\(GRUB_CMDLINE_LINUX=".*\)"/\1 intel_iommu=on"/' $GRUBFILE
> $TMPFILE

size=`du -b $GRUBFILE | awk '{print $1}'`
nsize=`du -b $TMPFILE | awk '{print $1}'`

if [[ $nsize -lt $size ]]
then
    echo "Error"
    exit 1
fi

chown --reference=$GRUBFILE $TMPFILE
chmod --reference=$GRUBFILE $TMPFILE

mv $TMPFILE $GRUBFILE

grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

Set the file permissions & run the script:

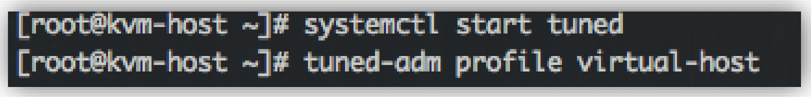
```
# chmod +x activate-vt-d.sh
# ./activate-vt-d.sh
```



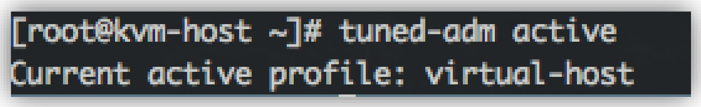
```
[opc@kvm-host ~]$ vi activate-vt-d.sh
[opc@kvm-host ~]$ sudo chmod 755 activate-vt-d.sh
[opc@kvm-host ~]$ sudo ./activate-vt-d.sh
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.1.12-112.14.15.el7uek.x86_64
Found initrd image: /boot/initramfs-4.1.12-112.14.15.el7uek.x86_64.img
Found linux image: /boot/vmlinuz-4.1.12-94.3.9.el7uek.x86_64
Found initrd image: /boot/initramfs-4.1.12-94.3.9.el7uek.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-693.21.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-693.21.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-693.17.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-693.17.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-693.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-693.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-3e5a2d0d105b47278b9f57145d68fb22
Found initrd image: /boot/initramfs-0-rescue-3e5a2d0d105b47278b9f57145d68fb22.img
Found linux image: /boot/vmlinuz-0-rescue-e3ab4b2b8f284b3896e124fdc208d45c
Found initrd image: /boot/initramfs-0-rescue-e3ab4b2b8f284b3896e124fdc208d45c.img
done
```

- Enable *tuned* and set the performance optimization for *virtual-host*

```
# systemctl enable tuned
# systemctl start tuned
# tuned-adm profile virtual-host
# tuned-adm active
```



```
[root@kvm-host ~]# systemctl start tuned
[root@kvm-host ~]# tuned-adm profile virtual-host
```



```
[root@kvm-host ~]# tuned-adm active
Current active profile: virtual-host
```

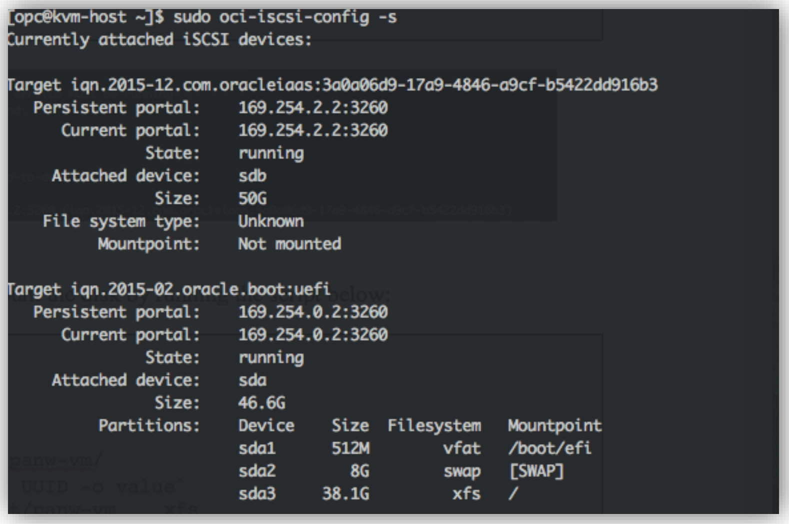
- Install ***oci-utils*** package for Oracle Linux, if not yet installed.

```
$ sudo yum install -y oci-utils
```

Attach a Block Volume to the KVM host to hold the NetScaler VM

- Attach the scsi disk (block volume) previously created.

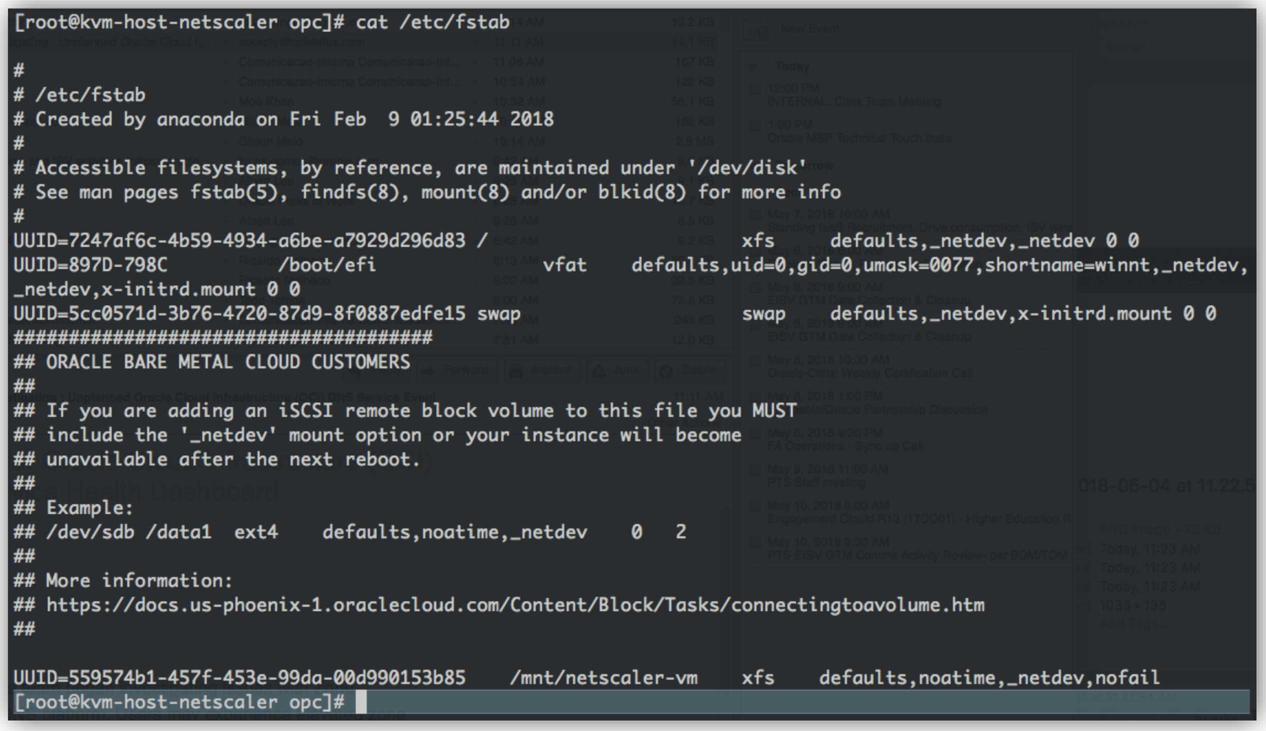
```
# oci-iscsi-config -s
```



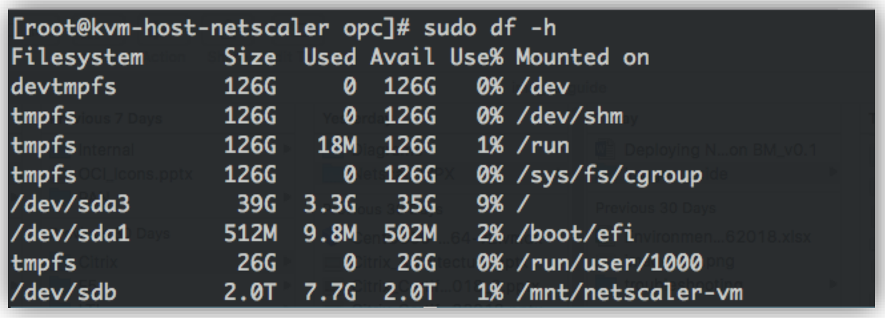
```
[opc@kvm-host ~]$ sudo oci-iscsi-config -s
Currently attached iSCSI devices:

Target iqn.2015-12.com.oracleiaas:3a0a06d9-17a9-4846-a9cf-b5422dd916b3
   Persistent portal:     169.254.2.2:3260
      Current portal:     169.254.2.2:3260
               State:     running
     Attached device:     sdb
                Size:     50G
    File system type:     Unknown
          Mountpoint:     Not mounted

Target iqn.2015-02.oracle.boot:uefi
   Persistent portal:     169.254.0.2:3260
      Current portal:     169.254.0.2:3260
               State:     running
     Attached device:     sda
                Size:     46.6G
          Partitions:     Device    Size  Filesystem   Mountpoint
                          sda1      512M        vfat   /boot/efi
                          sda2        8G        swap   [SWAP]
                          sda3     38.1G         xfs   /
```

- Create a file system and mount the disk by running the script below:

```
#!/bin/bash

mkfs.xfs /dev/sdb
mkdir /mnt/netscaler-vm
mount -t xfs /dev/sdb /mnt/netscaler-vm/
sdb_uuid=`blkid /dev/sdb -s UUID -o value`
echo "UUID=$sdb_uuid    /mnt/netscaler-vm    xfs
defaults,noatime,_netdev,nofail" >> /etc/fstab
```

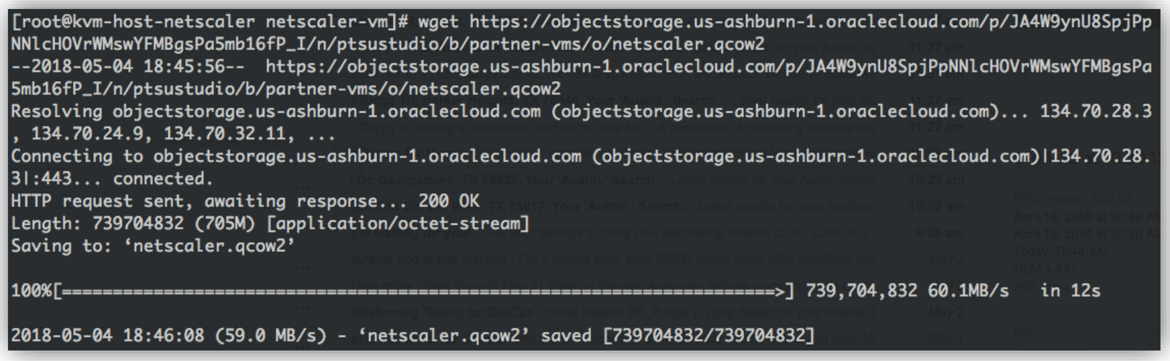


```
[root@kvm-host-netscaler opc]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Feb  9 01:25:44 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=7247af6c-4b59-4934-a6be-a7929d296d83 /                       xfs     defaults,_netdev,_netdev 0 0
UUID=897D-798C          /boot/efi               vfat    defaults,uid=0,gid=0,umask=0077,shortname=winnt,_netdev,
_netdev,x-initrd.mount 0 0
UUID=5cc0571d-3b76-4720-87d9-8f0887edfe15 swap                    swap    defaults,_netdev,x-initrd.mount 0 0
######################################
## ORACLE BARE METAL CLOUD CUSTOMERS
##
## If you are adding an iSCSI remote block volume to this file you MUST
## include the '_netdev' mount option or your instance will become
## unavailable after the next reboot.
##
## Example:
## /dev/sdb /data1  ext4    defaults,noatime,_netdev    0   2
##
## More information:
## https://docs.us-phoenix-1.oraclecloud.com/Content/Block/Tasks/connectingtoavolume.htm
##

UUID=559574b1-457f-453e-99da-00d990153b85    /mnt/netscaler-vm    xfs    defaults,noatime,_netdev,nofail
[root@kvm-host-netscaler opc]#
```



```
[root@kvm-host-netscaler opc]# sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        126G     0  126G   0% /dev
tmpfs           126G     0  126G   0% /dev/shm
tmpfs           126G   18M  126G   1% /run
tmpfs           126G     0  126G   0% /sys/fs/cgroup
/dev/sda3        39G  3.3G   35G   9% /
/dev/sda1       512M  9.8M  502M   2% /boot/efi
tmpfs            26G     0   26G   0% /run/user/1000
/dev/sdb        2.0T  7.7G  2.0T   1% /mnt/netscaler-vm
```

- Upload the NetScaler qcow2 image file to /mnt/netscaler-vm

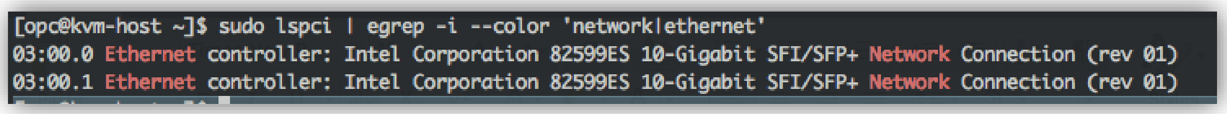


```
[root@kvm-host-netscaler netscaler-vm]# wget https://objectstorage.us-ashburn-1.oraclecloud.com/p/JA4W9ynU8SpjPp
NNlcHOVrWMswYFMBgsPa5mb16fP_I/n/ptsustudio/b/partner-vms/o/netscaler.qcow2
--2018-05-04 18:45:56--  https://objectstorage.us-ashburn-1.oraclecloud.com/p/JA4W9ynU8SpjPpNNlcHOVrWMswYFMBgsPa
5mb16fP_I/n/ptsustudio/b/partner-vms/o/netscaler.qcow2
Resolving objectstorage.us-ashburn-1.oraclecloud.com (objectstorage.us-ashburn-1.oraclecloud.com)... 134.70.28.3
, 134.70.24.9, 134.70.32.11, ...
Connecting to objectstorage.us-ashburn-1.oraclecloud.com (objectstorage.us-ashburn-1.oraclecloud.com)|134.70.28.
3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 739704832 (705M) [application/octet-stream]
Saving to: ‘netscaler.qcow2’

100%[=========================================================================>] 739,704,832 60.1MB/s   in 12s

2018-05-04 18:46:08 (59.0 MB/s) - ‘netscaler.qcow2’ saved [739704832/739704832]
```

Configure the Network on the KVM Host

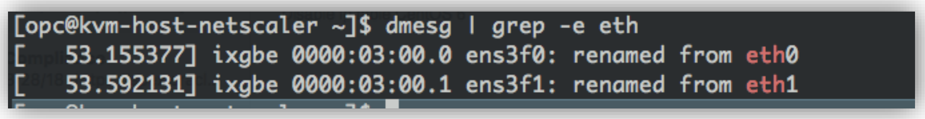
NetScaler VM Series was tested on Oracle Cloud Infrastructure BM*1* (first-generation) compute shape, which comes with only **one active** Intel 82599 based 10G NIC. We can use a combination of SR-IOV virtual functions (VFs) and the multi-VNIC feature of OCI to support the connectivity of the NetScaler to the network.

Below you can check that there are 2 network controllers:



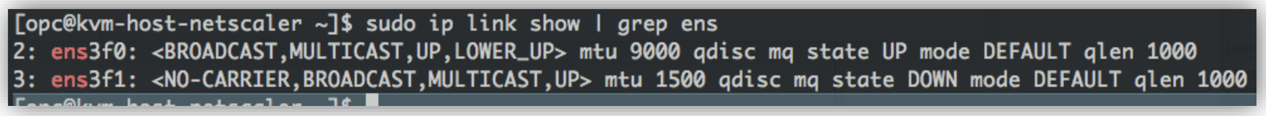
```
[opc@kvm-host ~]$ sudo lspci | egrep -i --color 'network|ethernet'
03:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
03:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
```

And how they are mapped to the network cards - note the PCI Address “0000:03.00.0 & 0000:03.00.1”.



```
[opc@kvm-host-netscaler ~]$ dmesg | grep -e eth
[   53.155377] ixgbe 0000:03:00.0 ens3f0: renamed from eth0
[   53.592131] ixgbe 0000:03:00.1 ens3f1: renamed from eth1
```

In the KVM Bare Metal host (X5 – BM.1 shapes), the NIC 0 is automatically configured with the primary VNIC’s IP configuration (the IP address, DNS hostname, and so on). The second NIC 1 is not active and should not be used. **ens3f0** is the only interface whose state is “up”.



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

All the next steps were created based on the official OCI Whitepaper available in the documentation: Installing and configuring KVM on Bare Metal Instances with Multi-VNIC.

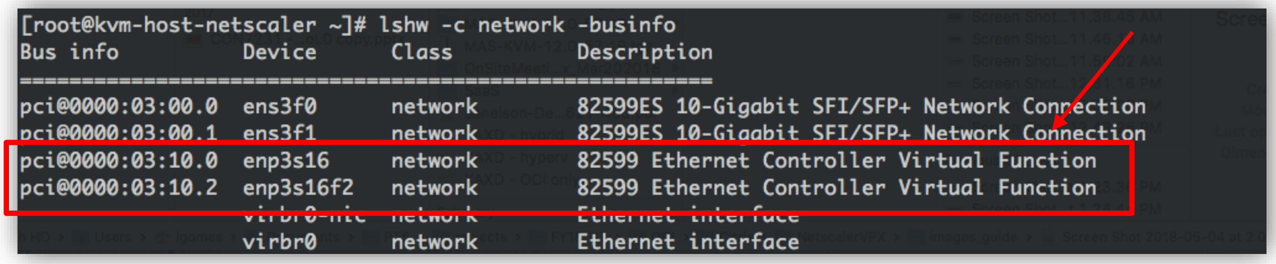
In order to avoid losing your network configuration across reboots, we will create a Linux Service and also configure the network device files attached to the KVM hypervisor in the Bare Metal Host.

- First step is to create a script and save to /usr/bin/initialize-kvm-network.sh . This script will hold the logic to enable the Virtual Functions Device and also to initialize the additional network devices. Copy the content from the snippet below into the script and make sure that this file is executable (chmod +x /usr/bin/initialize-kvm-network.sh):
  Note: You can modify the number of virtual functions through the parameter ‘number_vfs’

```
#!/bin/sh

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
```

- Next, run the script to enable Virtual Function devices
- Take notes of the Device associated with the virtual functions by running the command below:

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



```
[root@kvm-host-netscaler ~]# 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
                  virbr0-nic  network    Ethernet interface
                  virbr0      network    Ethernet interface
```

- Run “ip link” to identify the corresponding MAC address for these devices. Make sure that the MAC address of these devices matches the corresponding values of vnic2 and vnic3.

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

- Next, create a configuration file under /etc/sysconfig/network-scripts/ for each VF device, based on the template below:

Filename: ifcfg-<VF Device>

```
DEVICE=<VF Device Name>
BOOTPROTO=none
ONBOOT=yes
MACADDR="<VNIC MAC ADDRESS>"
NM_CONTROLLED=no
MTU=9000
```

- Based on our example, we should have the following files:

Config File	**Content**

/etc/sysconfig/network-scripts/ifcfg-enp3s16	DEVICE=enp3s16
	BOOTPROTO=none
	ONBOOT=yes
	MACADDR="02:00:17:02:A5:C4"
	NM_CONTROLLED=no
	MTU=9000

/etc/sysconfig/network-scripts/ifcfg-enp3s16f2	DEVICE=enp3s16f2
	BOOTPROTO=none
	ONBOOT=yes
	MACADDR="02:00:17:02:B9:4E"
	NM_CONTROLLED=no
	MTU=9000

- Next, we should create a vlan configuration file for each VF Device based on the template below. The VLAN devices will become available to NetScaler.

Filename: ifcfg-<VF Device>.vlan<vlan tag>

```
DEVICE=vlan<vlan tag>
PHYSDEV=<VF Device>
BOOTPROTO=none
ONBOOT=yes
NM_CONTROLLED=no
VLAN="yes"
IPADDR="<private IP>"
NETMASK="<Subnet MASK>"
DNS1=169.254.169.254
```

- Again, we should have the following files:

Config File	**Content**	**VLAN Link name**

/etc/sysconfig/network-scripts/ifcfg-enp3s16.vlan2	DEVICE=vlan2
	PHYSDEV=enp3s16
	BOOTPROTO=none
	ONBOOT=yes
	NM_CONTROLLED=no
	VLAN="yes"
	IPADDR="10.0.200.2"
	NETMASK="255.255.255.0"
	DNS1=169.254.169.254
		vlan2@enp3s16

/etc/sysconfig/network-scripts/ifcfg-enp3s16f2.vlan3	DEVICE=vlan3
	PHYSDEV=enp3s16f2
	BOOTPROTO=none
	ONBOOT=yes
	NM_CONTROLLED=no
	VLAN="yes"
	IPADDR="10.0.201.2"
	NETMASK="255.255.255.0"
	DNS1=169.254.169.254
		vlan3@enp3s16f2

- Then, append to /usr/bin/initialize-kvm-network.sh file the commands to start the network devices.

```
ifup enp3s16
ifup enp3s16f2
ifup vlan2
ifup vlan3
```

- After that, we should create a service file: /etc/systemd/system/kvm-network.service

```
[Unit]
Description=Enable KVM Network
Wants=network-online.target
After=cloud-init-local.service network.target network-online.target


[Service]
Type=notify
ExecStart=/usr/bin/initialize-kvm-network.sh
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure


[Install]
WantedBy=multi-user.target
```

- Finally, enable and start the service:

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

- Now, in case of reboot, the service will be automatically restarted.
- Reboot your instance

Install NetScaler VPX on KVM

Make sure you completed all the steps of the previous section before installing NetScaler. The libvirt API that is used to manage KVM includes a host of tools that allow you to create and manage virtual machines. To install NetScaler VPX on OCI running on top of KVM hypervisor, you can use any of the following methods:

1. Manually create the XML definition of the NetScaler VPX, then use virsh to import the definition. Virsh is the most powerful tool that allows for full administration of the virtual machine.
2. Use virt-install to create the definition for the NetScaler VPX and install it.

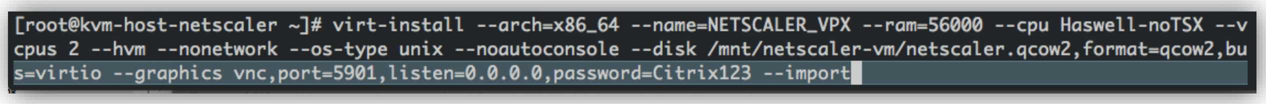
On this example we will use the approach (2), where virt-install will create the KVM domain, install the guest image and then, virsh-attach will be used to attach network interfaces..

- Create the domain to place the NetScaler VM as the guest OS.

```
virt-install --arch=x86_64 --name=NETSCALER_VPX --ram=56000 --cpu Haswell-
noTSX --vcpus 2 --hvm --nonetwork --os-type unix --noautoconsole --disk
/mnt/netscaler-vm/netscaler.qcow2,format=qcow2,bus=virtio --graphics
vnc,port=5901,listen=0.0.0.0,password=Citrix123 --import
```

Notes:

- The list of parameters may change accordingly to the NetScaler version and customer requirements. Verify NetScaler documentation for additional details.
- --nonetwork parameter was specified. This means that the network devices should be attached to the domain in the upcoming steps.
-



```
[root@kvm-host-netscaler ~]# virt-install --arch=x86_64 --name=NETSCALER_VPX --ram=56000 --cpu Haswell-noTSX --v
cpus 2 --hvm --nonetwork --os-type unix --noautoconsole --disk /mnt/netscaler-vm/netscaler.qcow2,format=qcow2,bu
s=virtio --graphics vnc,port=5901,listen=0.0.0.0,password=Citrix123 --import
```

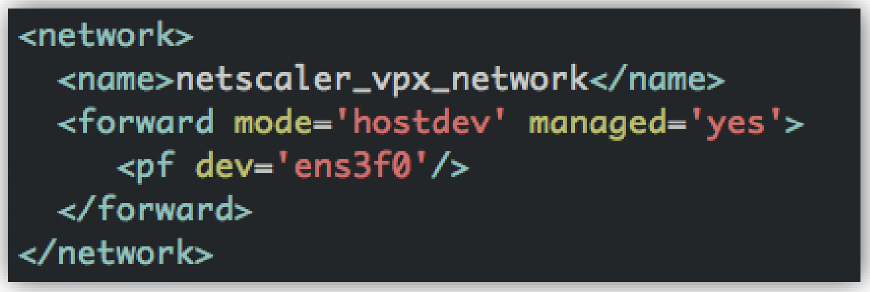
Attach the Network Devices to the Domain – SR-IOV Virtual Network Adapter Pool

Once domain creation completed, we will create a virtual network based on the NIC PCI physical function. Using this method, KVM creates a pool of network devices that can be attached to the NetScaler VM, and the size of the pool is determined by the number of VFs we created earlier.

In order to create a virtual network, we need to create a xml file based on the template below that maps to the network device which hosts all the virtual functions:

```
<network>
  <name> [network_name] </name>
  <forward mode='hostdev' managed='yes'>
   <pf dev='[device name]'/>
  </forward>
</network>
```

So, in your example, ens3f0 is the device name mapped to the VFs and as the result we will have the following xml:



```
<network>
  <name>netscaler_vpx_network</name>
  <forward mode='hostdev' managed='yes'>
     <pf dev='ens3f0'/>
  </forward>
</network>
```

Now we should load the new xml file into the KVM to create the network:

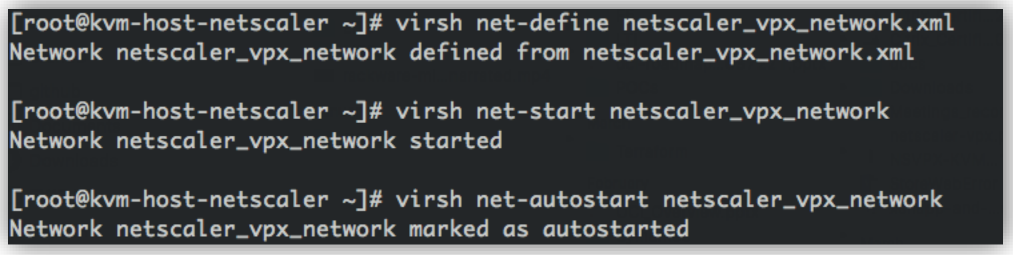
```
virsh net-define netscaler_vpx_network.xml
```

To start the virtual network, run the command: virsh net-start [network_name_in_xml]:

```
virsh net-start netscaler_vpx_network
```

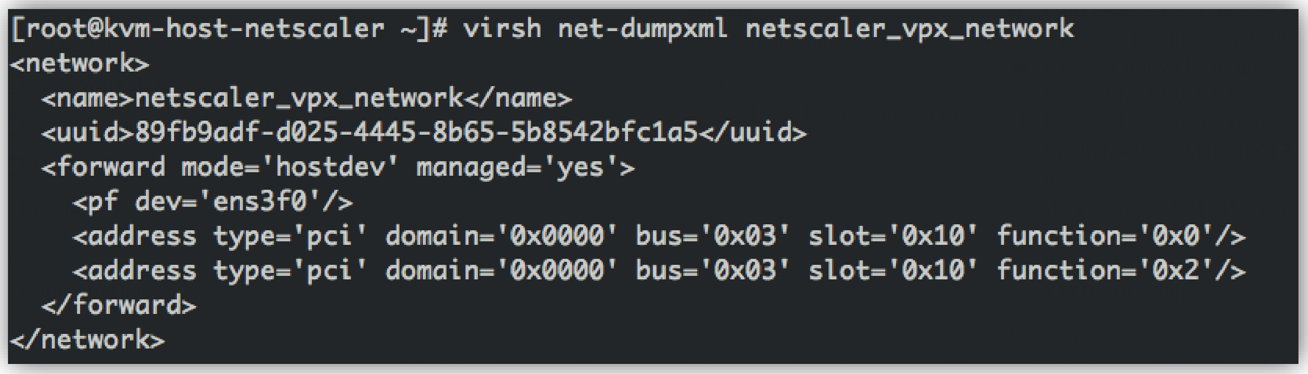
To automatically start the network when running KVM you can call autostart:

```
virsh net-autostart netscaler_vpx_network
```



```
[root@kvm-host-netscaler ~]# virsh net-define netscaler_vpx_network.xml
Network netscaler_vpx_network defined from netscaler_vpx_network.xml

[root@kvm-host-netscaler ~]# virsh net-start netscaler_vpx_network
Network netscaler_vpx_network started

[root@kvm-host-netscaler ~]# virsh net-autostart netscaler_vpx_network
Network netscaler_vpx_network marked as autostarted
```

You can verify that the network was successfully attached by running **virsh net-dumpxml netscaler_vpx_network.** The VF addresses should matches the PCI values we previously mapped.



```
[root@kvm-host-netscaler ~]# virsh net-dumpxml netscaler_vpx_network
<network>
  <name>netscaler_vpx_network</name>
  <uuid>89fb9adf-d025-4445-8b65-5b8542bfc1a5</uuid>
  <forward mode='hostdev' managed='yes'>
    <pf dev='ens3f0'/>
    <address type='pci' domain='0x0000' bus='0x03' slot='0x10' function='0x0'/>
    <address type='pci' domain='0x0000' bus='0x03' slot='0x10' function='0x2'/>
  </forward>
</network>
```

Next, you should attach all the devices that you want to expose to the NetScaler VM by creating a XML following the template below. Create one file for each interface, replace your information for the various placeholders:

attach.xml

```
<interface type='direct'>
  <source dev='vlan[vnic vlan tag]' mode='passthrough'/>
  <target dev='macvtap[vnic vlan tag]'/>
  <model type='virtio'/>
  <alias name='net[vnic vlan tag]'/>
  <mac address='[vnic mac address]'/>
</interface>
```

In order to attach the interfaces to the domain, you should run the following command (per interface/xml file):

```
virsh attach-device <your_domain_name> ./attach.xml –config
```

Below you have the list of interfaces we previously mapped:

Interface	**VNIC Name**	**Private IP**	**MAC**	**VLAN tag**	**VF**
nsip	vnic2	10.0.200.16	02:00:17:00:13:8E	2	0

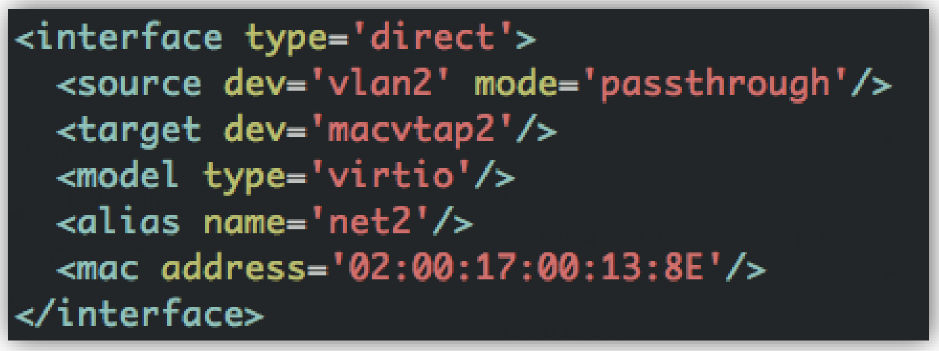
snip	vnic3	10.0.100.6	02:00:17:00:8C:26	3	1

- Create **nsip.xml**

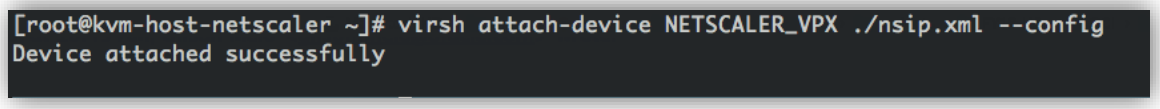
```
<interface type='direct'>
  <source dev='vlan2' mode='passthrough'/>
  <target dev='macvtap2'/>
  <model type='virtio'/>
  <alias name='net2'/>
  <mac address='02:00:17:00:13:8E'/>
</interface>
```

- Attach the **nsip/vnic2** device to the domain:

```
# virsh attach-device NETSCALER_VPX ./nsip.xml --config
```



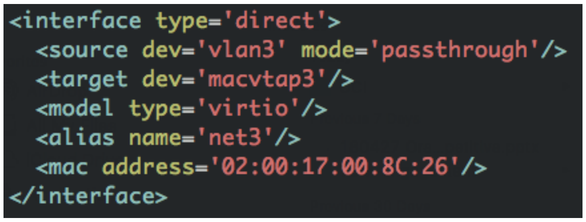
```
<interface type='direct'>
  <source dev='vlan2' mode='passthrough'/>
  <target dev='macvtap2'/>
  <model type='virtio'/>
  <alias name='net2'/>
  <mac address='02:00:17:00:13:8E'/>
</interface>
```



```
[root@kvm-host-netscaler ~]# virsh attach-device NETSCALER_VPX ./nsip.xml --config
Device attached successfully
```

- Create **snip.xml**

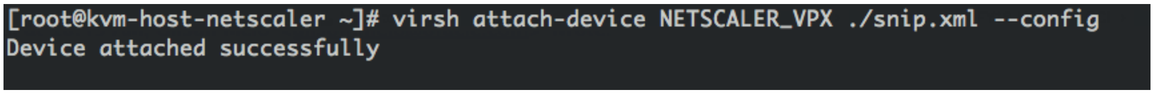
```
<interface type='direct'>
  <source dev='vlan3' mode='passthrough'/>
  <target dev='macvtap3'/>
  <model type='virtio'/>
  <alias name='net3'/>
  <mac address='02:00:17:00:8C:26'/>
</interface>
```



```
<interface type='direct'>
  <source dev='vlan3' mode='passthrough'/>
  <target dev='macvtap3'/>
  <model type='virtio'/>
  <alias name='net3'/>
  <mac address='02:00:17:00:8C:26'/>
</interface>
```

- Attach the **snip/vnic3** device to the domain:

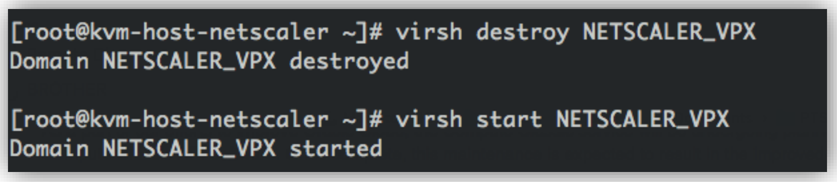
```
 virsh attach-device NETSCALER_VPX ./snip.xml --config
```



```
[root@kvm-host-netscaler ~]# virsh attach-device NETSCALER_VPX ./snip.xml --config
Device attached successfully
```

- Force restart the domain by running the commands below:

```
 # virsh destroy NETSCALER_VPX
 # virsh start NETSCALER_VPX
```



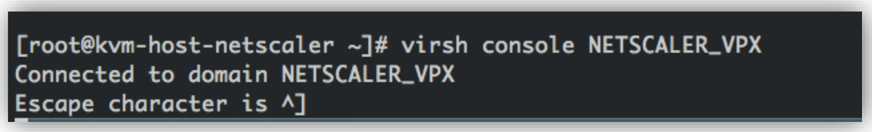
```
[root@kvm-host-netscaler ~]# virsh destroy NETSCALER_VPX
Domain NETSCALER_VPX destroyed

[root@kvm-host-netscaler ~]# virsh start NETSCALER_VPX
Domain NETSCALER_VPX started
```

- You can verify all network devices are attached by running virsh dumpxml NETSCALER_VPX

Connect to the NetScaler Console

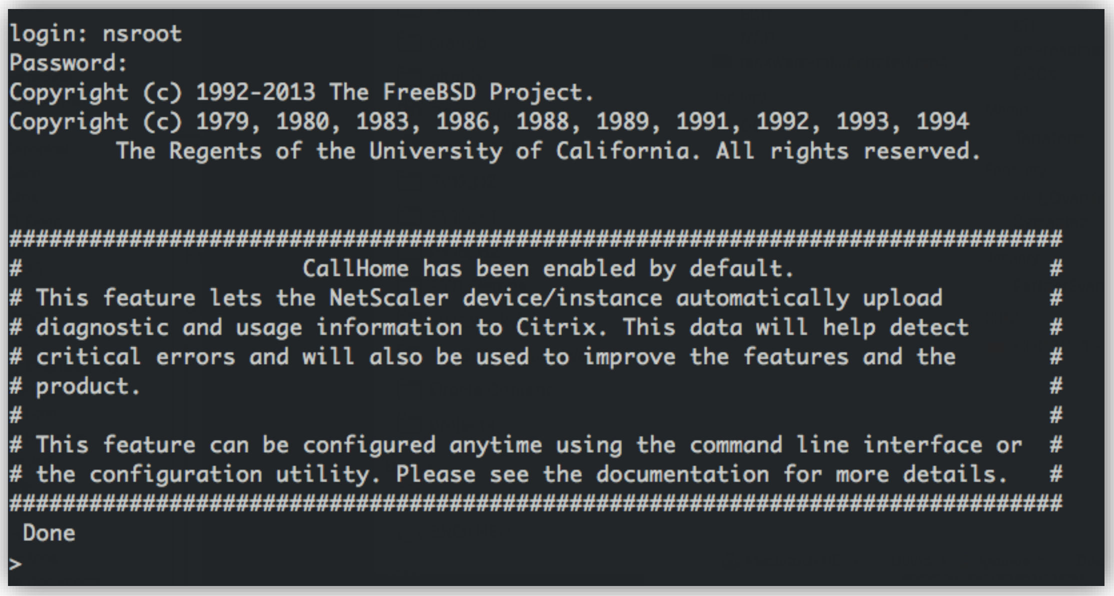
- Once your domain is running, you can connect to the NetScaler console using virsh:

```
# virsh console NETSCALER_VPX
```



```
[root@kvm-host-netscaler ~]# virsh console NETSCALER_VPX
Connected to domain NETSCALER_VPX
Escape character is ^]
```

- Press Enter to get access to the login page. It may take some time for NetScaler VPX to complete boot process. After that, login to NetScaler with username/password: nsroot/nsroot



```
login: nsroot
Password:
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.


###############################################################################
#                     CallHome has been enabled by default.                   #
# This feature lets the NetScaler device/instance automatically upload        #
# diagnostic and usage information to Citrix. This data will help detect      #
# critical errors and will also be used to improve the features and the       #
# product.                                                                    #
#                                                                             #
# This feature can be configured anytime using the command line interface or  #
# the configuration utility. Please see the documentation for more details.   #
###############################################################################
 Done
>
```

- Configure NetScaler, e.g. set up NSIP, SNIP

nsip	vnic2	10.0.200.16	02:00:17:00:13:8E	2	0
snip	vnic3	10.0.100.6	02:00:17:00:8C:26	3	1

- NSIP setup:

```
set ns config -IPAddress <nsip/vnic2 ip_addr> -netmask <netmask>
show ns config

add route 0 0 <FrontEnd subnet gateway>
show route
save config
```

```
set ns config -IPAddress 10.0.200.16 -netmask 255.255.255.0
show ns config


add route 0 0 10.0.200.1
show route
save config
```

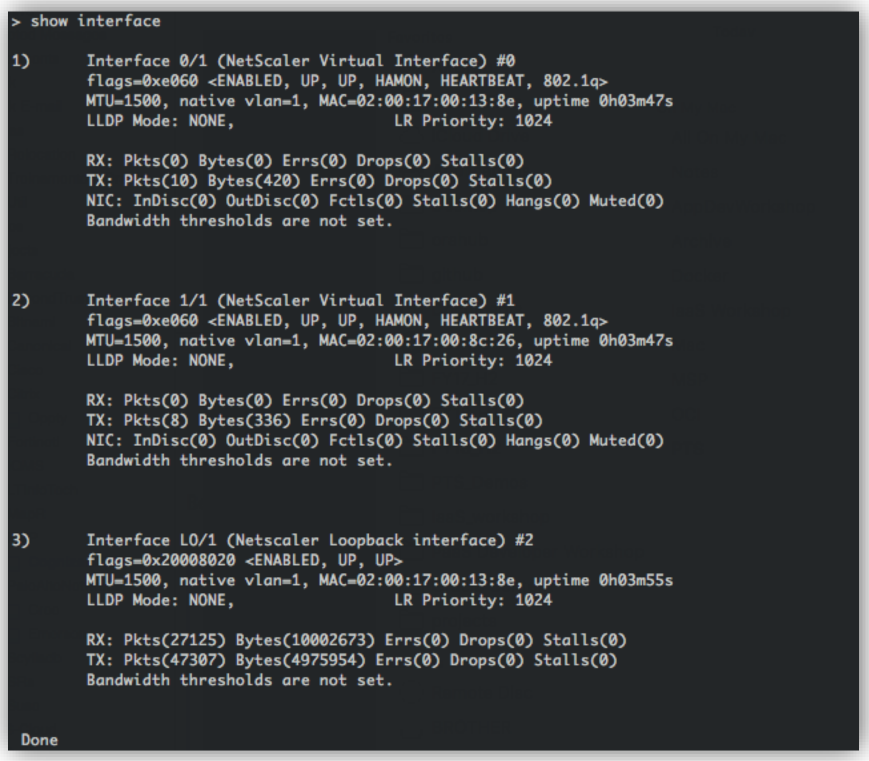
- SNIP Setup:

```
add ns ip <snip ip> <netmask> -type SNIP
show ns ip <snip ip>
```

```
add ns ip 10.0.100.6 255.255.255.0 -type SNIP
show ns ip 10.0.100.6
```

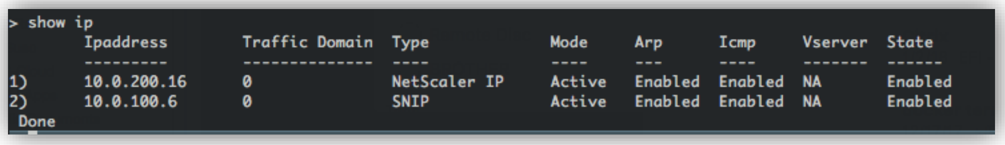
- Verify the Interfaces attached to your NetScaler (MAC address should match the values were attached to the KVM domain):

```
show interface
```



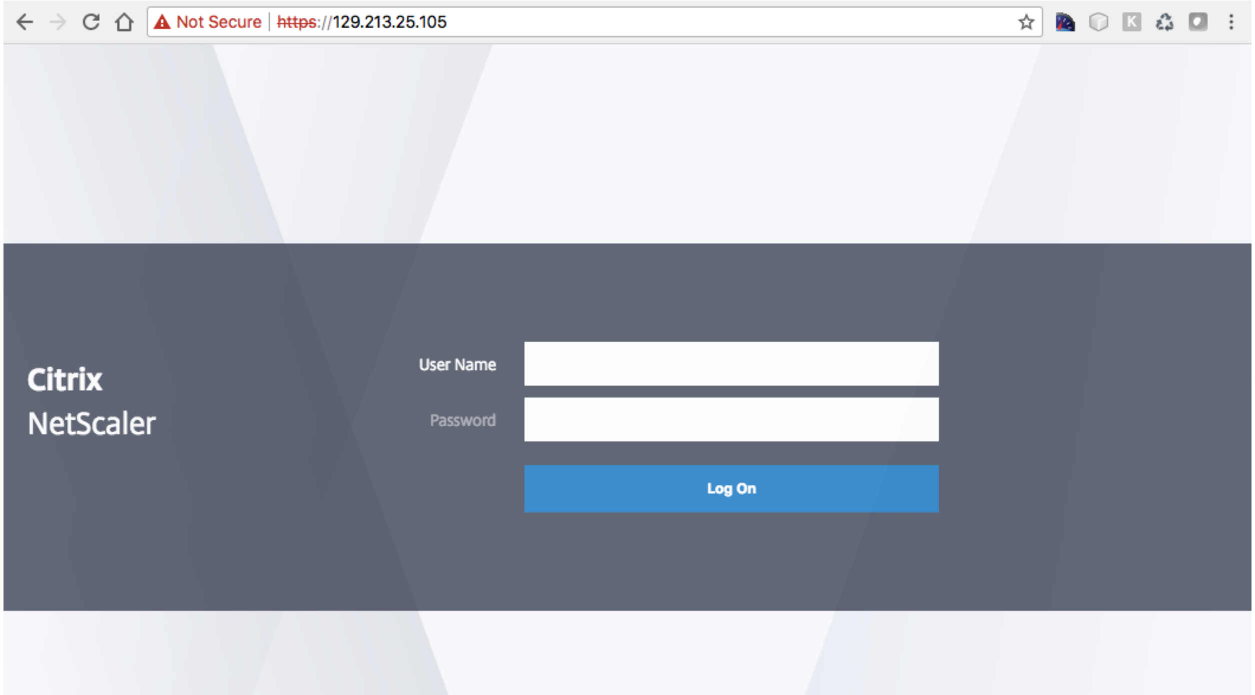
```
> show interface

1)      Interface 0/1 (NetScaler Virtual Interface) #0
        flags=0xe060 <ENABLED, UP, UP, HAMON, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=1, MAC=02:00:17:00:13:8e, uptime 0h03m47s
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(10) Bytes(420) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.


2)      Interface 1/1 (NetScaler Virtual Interface) #1
        flags=0xe060 <ENABLED, UP, UP, HAMON, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=1, MAC=02:00:17:00:8c:26, uptime 0h03m47s
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(8) Bytes(336) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.


3)      Interface LO/1 (Netscaler Loopback interface) #2
        flags=0x20008020 <ENABLED, UP, UP>
        MTU=1500, native vlan=1, MAC=02:00:17:00:13:8e, uptime 0h03m55s
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(27125) Bytes(10002673) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(47307) Bytes(4975954) Errs(0) Drops(0) Stalls(0)
        Bandwidth thresholds are not set.


 Done
```

- Check the IP addresses:

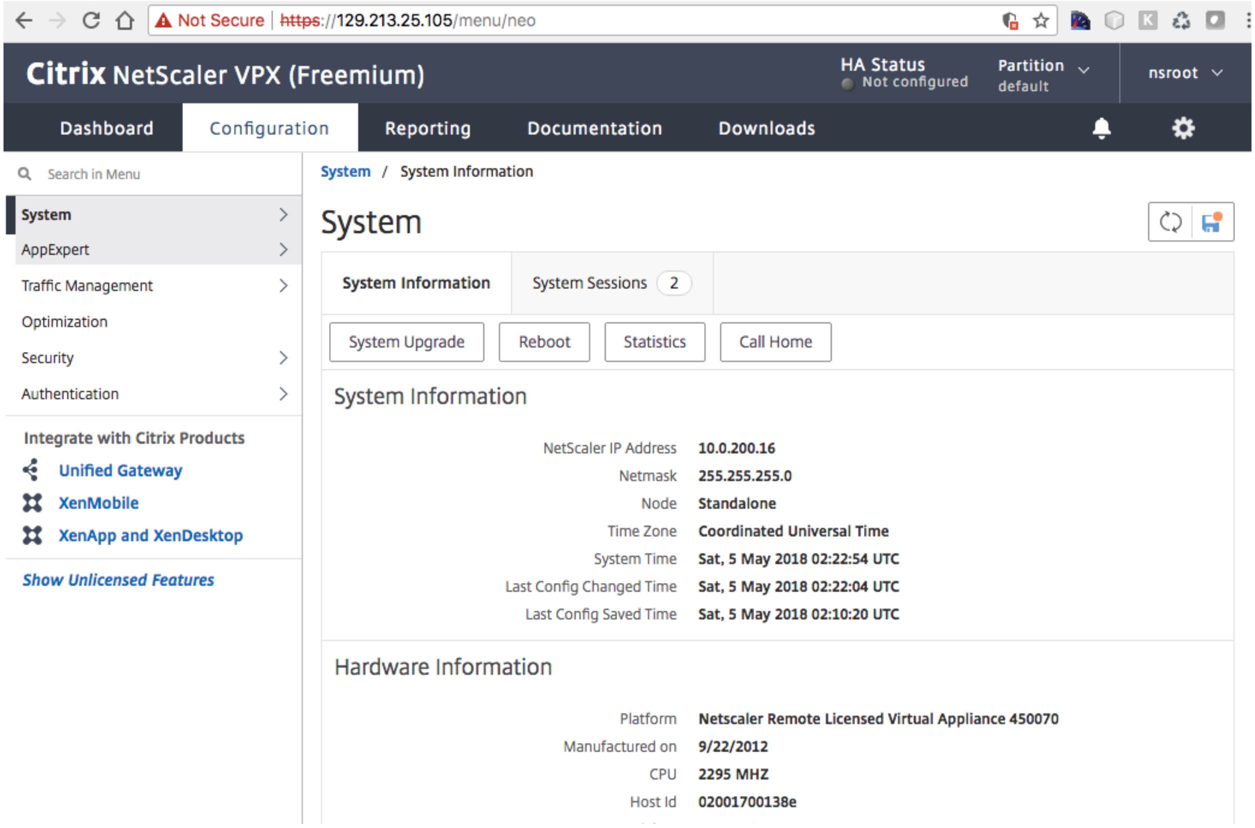


```
> show ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.0.200.16      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.0.100.6       0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
```

- Access the web interface (public IP associated with NSIP) of the NetScaler VPX.



Not Secure | ~~https~~://129.213.25.105



System Information

NetScaler IP Address: 10.0.200.16
Netmask: 255.255.255.0
Node: Standalone
Time Zone: Coordinated Universal Time
System Time: Sat, 5 May 2018 02:22:54 UTC
Last Config Changed Time: Sat, 5 May 2018 02:22:04 UTC
Last Config Saved Time: Sat, 5 May 2018 02:10:20 UTC

Hardware Information

Platform: Netscaler Remote Licensed Virtual Appliance 450070
Manufactured on: 9/22/2012
CPU: 2295 MHZ
Host Id: 02001700138e