BreakingPoint VE

Installation Guide

Version 9.10 update 2

Document version: 202101261332-06:00

Notices

Copyright Notice

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data. 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contacting Us

Keysight headquarters

1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1738
www.ixiacom.com/contact/info

Support

Global Support	+1 818 595 2599	support@ixiacom.com
Regional and local support contacts:		
APAC Support	+91 80 4939 6410	support@ixiacom.com
Australia	+61-742434942	support@ixiacom.com
EMEA Support	+40 21 301 5699	support-emea@ixiacom.com
Greater China Region	+400 898 0598	support-china@ixiacom.com
Hong Kong	+852-30084465	support@ixiacom.com
India Office	+91 80 4939 6410	support-india@ixiacom.com
Japan Head Office	+81 3 5326 1980	support-japan@ixiacom.com
Korea Office	+82 2 3461 0095	support-korea@ixiacom.com
Singapore Office	+65-6215-7700	support@ixiacom.com
Taiwan (local toll-free number)	00801856991	support@ixiacom.com

Documentation conventions

The following documentation conventions are used in this guide:

Describing interactions with the UI

You can interact with products by using different input methods: keyboard, mouse, touch, and more. So in most parts of the user documentation, generic verbs have been used that work with any input method. In cases where input-neutral verbs do not work, mouse-specific verbs are used as the first choice, followed by touch-specific verbs as the second choice.

See the following table for examples on how you can interpret the different input methods.

Input-neutral	Mouse	Touch
Select **Modify**.	Click **Modify**.	Tap **Modify**.
Select **Accounts** > **Other accounts** > **Add an account**.	Click **Accounts** > **Other accounts** > **Add an account**.	Tap **Accounts** > **Other accounts** > **Add an account**.
To open the document in Outline view, select **View** > **Outline**.	To open the document in Outline view, click **View** > **Outline**.	To open the document in Outline view, tap **View** > **Outline**.
Select **Protocols**.	Click the **Protocols** tab.	Tap **Protocols**.
-NA-	Double-click the **Client** wizard.	Double-tap the **Client** wizard.
Open the **Packages** context menu.	Right-click **Packages** to open the shortcut menu.	Long tap **Packages** to open the shortcut menu.

Deprecated words

The following words have been replaced with new words, considering the audience profile, our modern approach to voice and style, and our emphasis to use input-neutral terms that support all input methods.

Old usage...	New usage...
shortcut menu, right-click menu	context menu
click, right-click	select
drag and drop	drag

CONTENTS

Related Documentation

The latest documentation for each release can be found on the Ixia Support website.

Related Documentation

Documentation	Description
BreakingPoint User Guide	Provides information on how to use the Control Center to set up, customize, and run traffic through devices under test.
BreakingPoint Release Notes	Provides information about new features, resolved customer issues, known defects and workarounds (if available).
BreakingPoint Online Help	Online documentation for all BreakingPoint products. Proper viewing will require a supported HTML browser.

BreakingPoint Virtual Edition Feature Support

The tables in this section describe the feature support for the BreakingPoint Virtual Edition.

Network Neighborhood	BPS VE	BPS on AWS	BPS on MS Azure	BPS on GCP
IPv4 Static Hosts	✓	✓	✓	✓
IPv6 Static Hosts	✓	✓	✓	NS
IPv4 External Hosts	✓	✓	✓	✓
IPv6 External Hosts	✓	✓	✓	NS
NAT	✓	NS	NS	NS
VLAN	✓	NS	NS	NS
IPv4 Router	✓	✓	NS	✓
IPv6 Router	✓	✓	NS	NS
DHCPv4 (client/server)	✓	NS	NS	NS
DHCPv6 (client/server)	NS	NS	NS	NS
IPv4 DNS	✓	✓	✓	✓
IPv6 DNS	✓	✓	✓	NS
IPsec IKEv1/IKEv2	✓ *1	NS	NS	NS
LTE(IPv4)	✓	NS	NS	NS
LTE(IPv6)	NS	NS	NS	NS
3G	NS	NS	NS	NS
6RD	NS	NS	NS	NS
DSLite	NS	NS	NS	NS
IPv6 SLAAC	NS	NS	NS	NS

*1-tested only for VMware hypervisor

Test Components	BPS VE	BPS on AWS	BPS on MS Azure	BPS on GCP
Live Application Simulator	✓	NS	NS	✓
Application Simulator	✓	✓	✓	✓
Client Simulation	✓	✓	✓	✓
Security	✓	✓ *1	✓ *1	✓
Malware	✓	✓ *1	✓ *1	✓
Session Sender	✓	✓	✓	✓
Stack Scrambler	✓	✓ *2	✓ *2	✓
SSL/TLS	✓	✓	✓	✓
Packet Capture	✓	✓	✓	✓
Impairment	NS	NS	NS	NS
Bit Blaster	✓	NS	NS	✓
Routing Robot	✓	✓	✓	✓
Recreate	✓	✓ *3	✓ *3	✓
SCTP	✓	✓	✓	✓

*1- Some attacks may get blocked by AWS.

*2 - Some invalid IP packet patterns are not compatible with AWS (traffic might get dropped by AWS).

*3 - Limited support. This is because Replay Capture File Without Modification mode replays libpcap formatted capture files without modifying Layer 2 through Layer 7 and AWS requires BPS to use the MAC address that corresponds to the interface that is sending the packets.

Google Cloud Platform limitations:

- Some attacks may get blocked by GCP
- MTU is limited to 1460 bytes
- Some bad TCP packets are blocked by GCP (bad L4 checksum, bad TCP header length)

BreakingPoint Labs	BPS VE	BPS on AWS	BPS on MS Azure	BPS on GCP
Session Sender Lab	✓	NS	NS	NS
RFC 2544 Lab	✓	NS	NS	NS
Multicast Lab	✓	NS	NS	NS
Lawful Intercept Lab	✓	NS	NS	NS
Device Validation Lab	NS	NS	NS	NS
Multibox Testing	NS	NS	NS	NS
Resiliency Score	NS	NS	NS	NS
Data Center Resiliency	NS	NS	NS	NS
DDoS Lab	✓	NS	NS	NS

CHAPTER 1 BPS VE Install on Hypervisor

This section of the guide describes how to install BreakingPoint Virtual Edition on a VMware or KVM hypervisor.

BPS VE installation on Microsoft's Hyper-V (using .vhd files) is covered in BPS VE Install on Hypervisor above .

Overview

BreakingPoint Virtual Edition is a software-based test platform that enables you to run a BreakingPoint vController and traffic generation blades on a virtual chassis.

BreakingPoint Virtual Edition offers the following benefits:

- **Low Hardware Cost**: You can use low-cost servers or dedicated virtualization servers to generate the traffic.
- **More Efficient use of Hardware**: The same servers used to generate Ixia traffic can also be used for other non-Ixia applications; or the virtual Ixia ports can be hosted on a virtualization server used to host other applications.
- **Ease of Use**: The BreakingPoint Virtual Edition user interface is nearly identical to the standard hardware versions which reduces the learning time.
- **Reduced System Administration**: The BreakingPoint Virtual Edition chassis does not need to be maintained or monitored in a lab because it is virtual in nature.
- **Rapid and Easy Deployment**: Virtual Ixia ports can be instantiated as necessary, used to generate traffic, and then destroyed when no longer needed.
- **Pre-configured Templates**: The BreakingPoint Virtual Edition is delivered as a pre-configured `.ova` template for VMware and as qcow2 image for KVM.

Basic Elements

The basic elements involved in the BreakingPoint Virtual Edition

- A simple installer based on a single OVA image, qcow2 image or installation script.
- Deployment and discovery tools for easy provisioning of Virtual Blades (vBlades).
- Standalone vBlade installation options.
- A license server that also runs on the BreakingPoint vController.

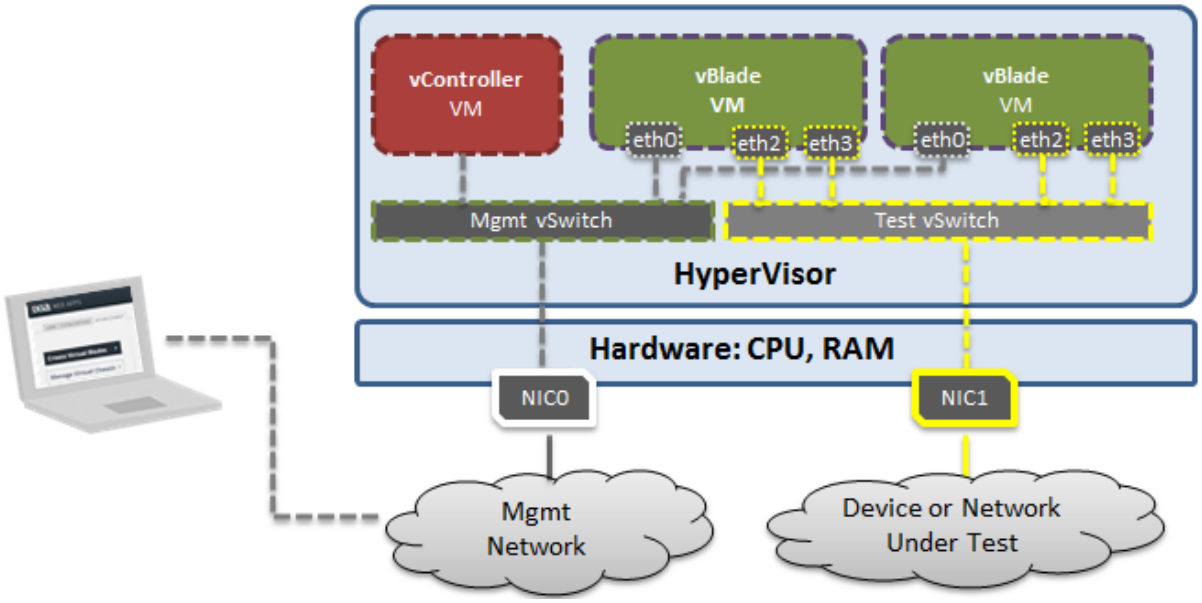
Components of the BreakingPoint Virtual Edition

The components of BPS VE are:

- vBlades for virtualization of load modules:
  - A single management interface
  - From two to eight virtual test ports

  See the Hardware Requirements for minimum vBlade specifications.
- vController for virtualization of the System Controller:
  - Controls up to 12 vBlades and up to 96 vPorts
  - Controls vBlades spanning across different physical servers

The following image depicts the components of the BreakingPoint Virtual Edition.



System Requirements

Before you deploy a BreakingPoint Virtual Chassis in a Virtual Environment, it is important to be aware of the following requirements and features.

- Hardware Requirements below
- Supported Platforms
- Software Requirements on the facing page
- BPS VE Adaptability to Low Resource Footprint on page 7
- Open Port Requirements for BPS VE on page 180

Hardware Requirements

The recommended minimum hardware requirements to install BreakingPoint in a Virtual Environment are as follows:

Note: Starting with release 8.10, BPS VE support is only available on DPDK enabled hardware. This functionality is currently supported with the Amazon ENA (Elastic Network Adapter) driver.

- Physical server based on Intel x86-64 architecture
- BreakingPoint vController Hardware Requirements - 8 GB RAM, 8 vCPU, 100 GB available hard disk space
- BreakingPoint vBlade Hardware Requirements - 8 GB RAM, 4 vCPU, 10 GB available hard disk space

Note: A BreakingPoint Virtual Chassis includes a vController and up to 12 vBlades.

Software Requirements

VMware ESX/ESXi Installation:

- Firmware ESXi 5.5.0 or ESXi 6.0 / 6.5 / 6.7 (Firmware vSphere Hypervisor)
- Firmware vSphere Client 5.5.0 or 6.0
- BreakingPoint installation OVA files for VMware

KVM Installation:

CentOS

Note: Testing was executed only with virtIO disk bus types. Sometimes due to the slowness of other disk bus types (for example, IDE) you may experience slowness and instability of the Virtual Machines. No testing was executed with remote storage in-house. Sometimes due to the slowness of the remote storage you may experience slowness and instability of the Virtual Machines.

- CentOS 7.x
  - QEMU emulator version 1.5.3 (qemu-kvm-1.5.3-126.el7_3.3)
  - QEMU emulator version 2.6.0 (qemu-kvm-ev-2.6.0-28.el7_3.6.1)
  - QEMU emulator version 2.9.0 (qemu-kvm-ev-2.9.0-16.el7_4.13.1)
- Virsh / libvirt versions
  - 2.0.0
  - 3.2.0
  - 3.9.0
  - 4.5.0

Ubuntu

- Ubuntu 16.04, Ubuntu 18.04
  - QEMU emulator version 2.0.0 (Debian 2.0.0+dfsg-2ubuntu1.19)
  - QEMU emulator version 2.5.0 (Debian 1:2.5+dfsg-5ubuntu10.16)

- Virsh / libvirt versions
  - 1.2.2
  - 1.3.1
  - 4.0.0

BPS VE Adaptability to Low Resource Footprint

BPS VE has resource adaptive features that allows the system to adapt and perform in a low resource footprint.

In a low resource environment, the minimum requirements for a BPS VE vBlade are:

- 1 GB RAM
- 1 vCPU
- 1 vNIC

BreakingPoint VE can also operate with a different amount of compute resources allocated to the Virtual Blade. This impacts the performance (determined as number of packets per second), scalability (determined as number of concurrent sessions), and maximum number of Test Components supported.

	SYSTEM CONTROLLER	VIRTUAL BLADE
Performance = Low Test Components (DPDK On) = 1 Test Components (DPDK Off) = 2	8 vCPUs 8 GB RAM	1 vCPUs 2 GB RAM
Performance = Medium Test Components (DPDK On) = 2 Test Components (DPDK Off) = 4	8 vCPUs 8 GB RAM	2 vCPUs 4 GB RAM
Performance = High Test Components (DPDK On) = 4 Test Components (DPDK Off) = 8	8 vCPUs 8 GB RAM	4 vCPUs 8 GB RAM
Performance = Very High Test Components (DPDK On) = 8 Test Components (DPDK Off) = 16	8 vCPUs 8 GB RAM	8 vCPUs 16 GB RAM

Super Flow and Throughput Objectives:

- BPS VE will try to achieve 125,000 super flow per second per component.
- BPS VE will try to achieve 10,000 Mbps per component.

Note: Capture is only supported when there is more than 2.5 GB of RAM available.

Note: The vBlade and vController Memory Errors that can occur are described in the Troubleshooting section.

Performance Acceleration

BPS VE supports a performance acceleration mode based on DPDK support. This functionality is currently supported with the Amazon ENA (Elastic Network Adapter) driver.

Note: A maximum of four components per vBlade can be run in performance acceleration mode. To run a maximum of eight components per vBlade, the "Enable Performance Acceleration" option needs to be unchecked.

Note: When using the DPDK Large Receive offload (LRO) feature, the LRO maximum length on ESX must be set to a value lower or equal to 9146 (because this is the maximum MSS value supported in BPS). If you are using the vmxnet3 driver, the parameter name is "Net.VmxnetLROMaxLength" and has the default value set to 32000.

Prerequisites for Performance Acceleration:

1. vBlade processor should have SIMD extensions SSSE3 or above enabled.
2. At least 8GB of RAM per vBlade.
3. Ixia recommends using VMware ESXi 6.0 with build number 3029758 or above.
4. Ixia recommends using the default settings of **Hypervisor**>**Configuration**>**Software**>**Advance Settings**>**Net**.

To enable Performance Acceleration:

Each vBlade on the Device Status page of the GUI displays a slot configuration button at the top-right corner.

1. Select the slot configuration button.
2. Select the **Enable Performance Acceleration** option.
3. Select the **Apply** button.

Getting Started

In a Virtual Environment, a virtual chassis consists of one virtual system controller (BreakingPoint vController) and up to 12 virtual blades (vBlades). Each vBlade allows you to provision from two to eight vPorts. The vBlades that send/receive traffic are also the traffic generation modules of BreakingPoint Virtual Edition.

The BreakingPoint vController runs the BreakingPoint Virtual Edition firmware and provides access to the HTML browser based BreakingPoint user interface.

Deployment Scenarios

You can deploy a vController and vBlades on the physical hosts in two scenarios:

- Single host setup
- Multi host setup

Single Host Setup

In a Single Host Setup, the vController and vBlades are on the same physical host supporting up to 12 vBlades per vController. The vController acts as a Virtual Machine (VM) and vBlades are the Linux VMs.

Multi Host Setup

In a Multi Host Setup, the vController is present on a single host, with or without vBlades. In all cases, a vController can support up to 12 vBlades. The other physical hosts are for vBlades only whereas multiple Linux VMs act as vBlades.

Network Topology Diagram

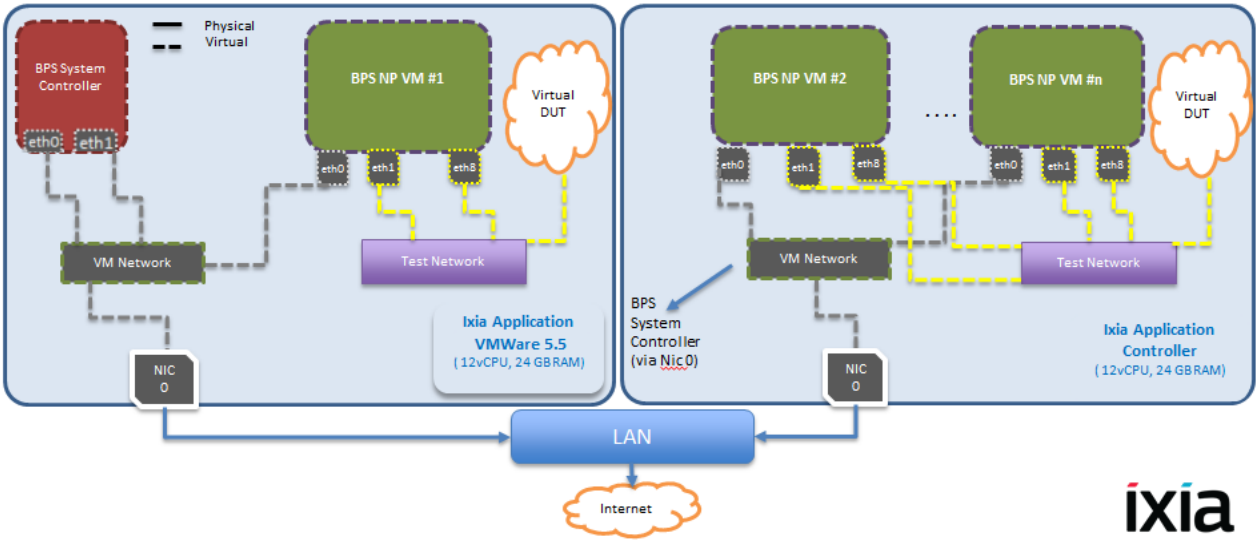
The test scenario shown in the image below has a minimum of two networks, a Virtual Machine Network (VM Network) and a Test Network.

- **Management Network** (control plane) - A Management Network is required to access the vController from a HTML browser (BPS user interface) as well as to communicate between the vController and vBlades. In this scenario, the vController and vBlades are split across several hypervisors. The Management Network (VM Network in the diagram below) in each hypervisor provides the Management-to vController-to-vBlade communications. To configure this topology, assign eth0 and eth1 of the vController (BPS System Controller) and eth0 of the vBlades (BPS NP VM #) to the Management network (VM Network). The vController can receive an IP address from a DHCP server via NIC0 in its hypervisor or the IP address can be manually configured. A vBlade can also optionally receive an IP address from a DHCP server. The NIC0 cards in both hypervisors are connected to the LAN Network.
- **Test Network** (data plane**)** - A Test Network is required to communicate within vPorts (port-to-Port test) or communicate to the virtual DUT (port-to-DUT test). Therefore, assign the Eth# ports

in the vBlades (except eth0, which is used for internal management) to the Test Network. You should also assign the NIC of the Virtual DUT to the same Test Network.

Note: In this scenario, all DUTs are present within the hypervisor. But a DUT may be present outside the hypervisor. In that scenario, assign the physical NICs except NIC0 (NIC0 in the hypervisor is already assigned to the management network) to the test network.

Note: By default, both vController interfaces are mapped to the VM Network (vSwitch0).

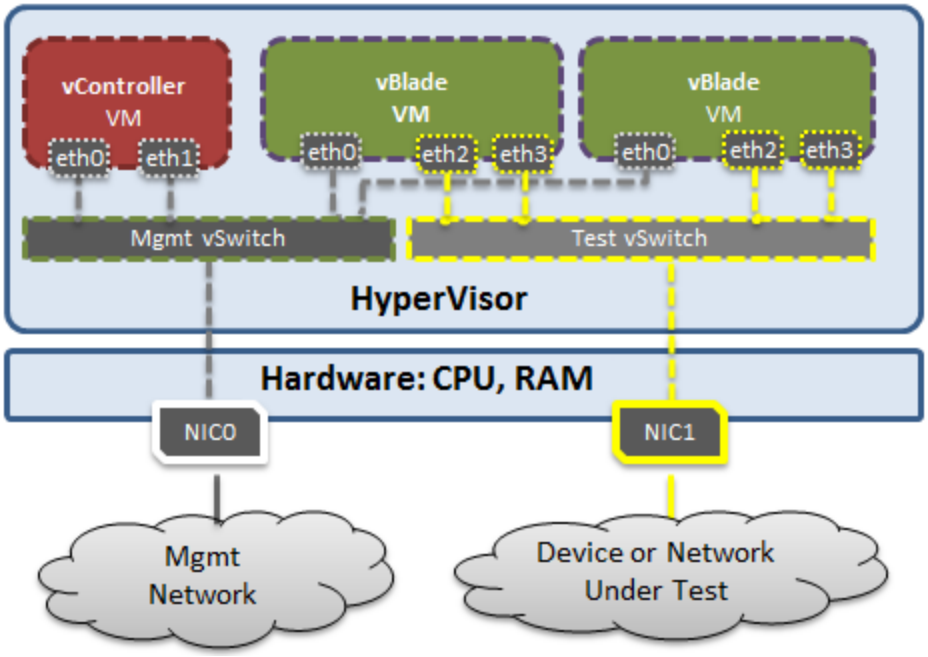


Note: A BP Virtual Chassis is resource sensitive. Not having the necessary resources may lead to instabilities in vBlade performance. It is essential that you utilize only the required number of vBlades/ports on a hypervisor. See the Hardware Requirements to calculate the resources that are required to support the vController/vBlades that will be used for your testing.

vController Management Interfaces

A vController has two management interfaces:

- External Management - Used to access the vController through web (BPS VE User Interface).
- Internal Management - Used for the internal communication between the vController and vBlades.



By default, both management interfaces are mapped to the vSwitch0 containing Management Network (Hypervisor IP address) and VM Network.

Alternatively, a dedicated internal management network can be created to connect the corresponding internal management interfaces of the vController and vBlades.

vBlades have one management interface:

- Used for the internal communication between vController and vBlades
- Must be in the same IP subnet with the vController internal management IP

Install BPS VE

This section provides detailed instructions for installing BreakingPoint Virtual Edition. Please ensure that you review the System Requirements before you begin.

There are 2 options for BPS VE hypervisor installation.

- VMware Installation
- KVM Installation

VMware Installation

This section describes the network configuration required for VMware and the vController VMware installation procedures.

Configure VMware vSwitch and Network

This section explains the vSwitch and network configuration required in VMWare before deploying BreakingPoint Virtual Edition.

It is recommended that you configure the following settings in all vSwitches across the hypervisors. If these settings are not configured, all of the network traffic may be available to all of the virtual machines, resulting in a non-functioning VLAN.

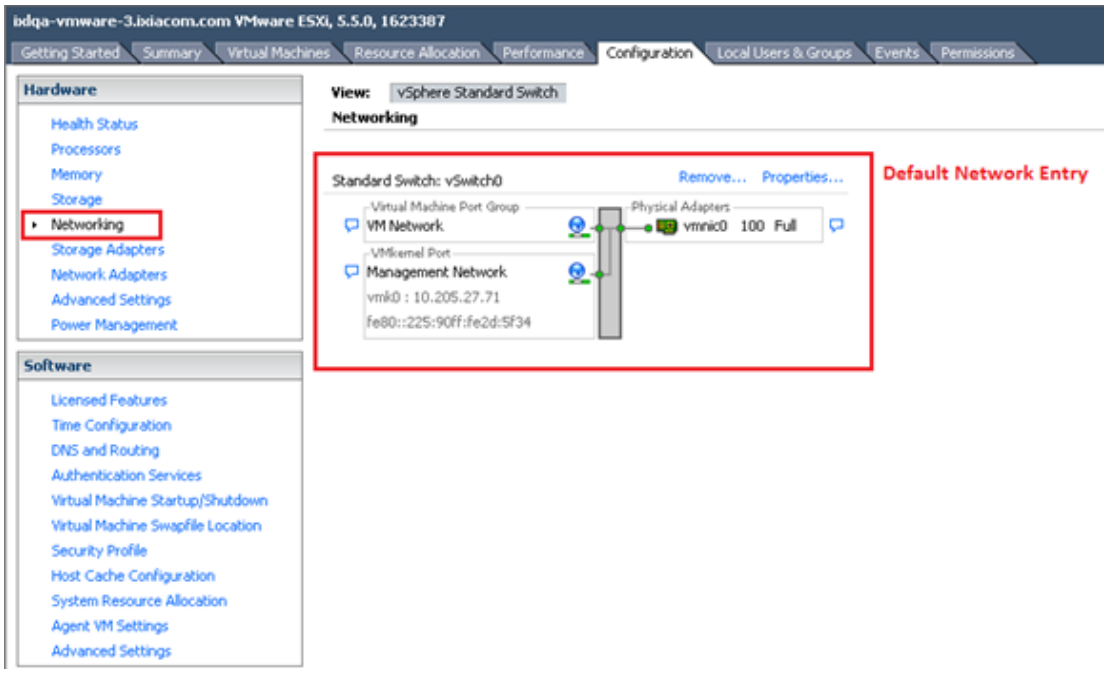
ESX server settings:

- vSwitch Traffic Shaping set as Disabled
- **vSwitch Security tab > Promiscuous Mode** set as **Accept** or **Reject**
  - **Note**: See Promiscuous Mode Recommendations on page 17 before configuring this setting
- vSwitch Properties, set the VLAN ID (Optional) from None (0) to All (4095)

To perform vSwitch and Network configuration perform the following tasks:

1. Log on to the hypervisor using the firmware vSphere Client as depicted in the following image.

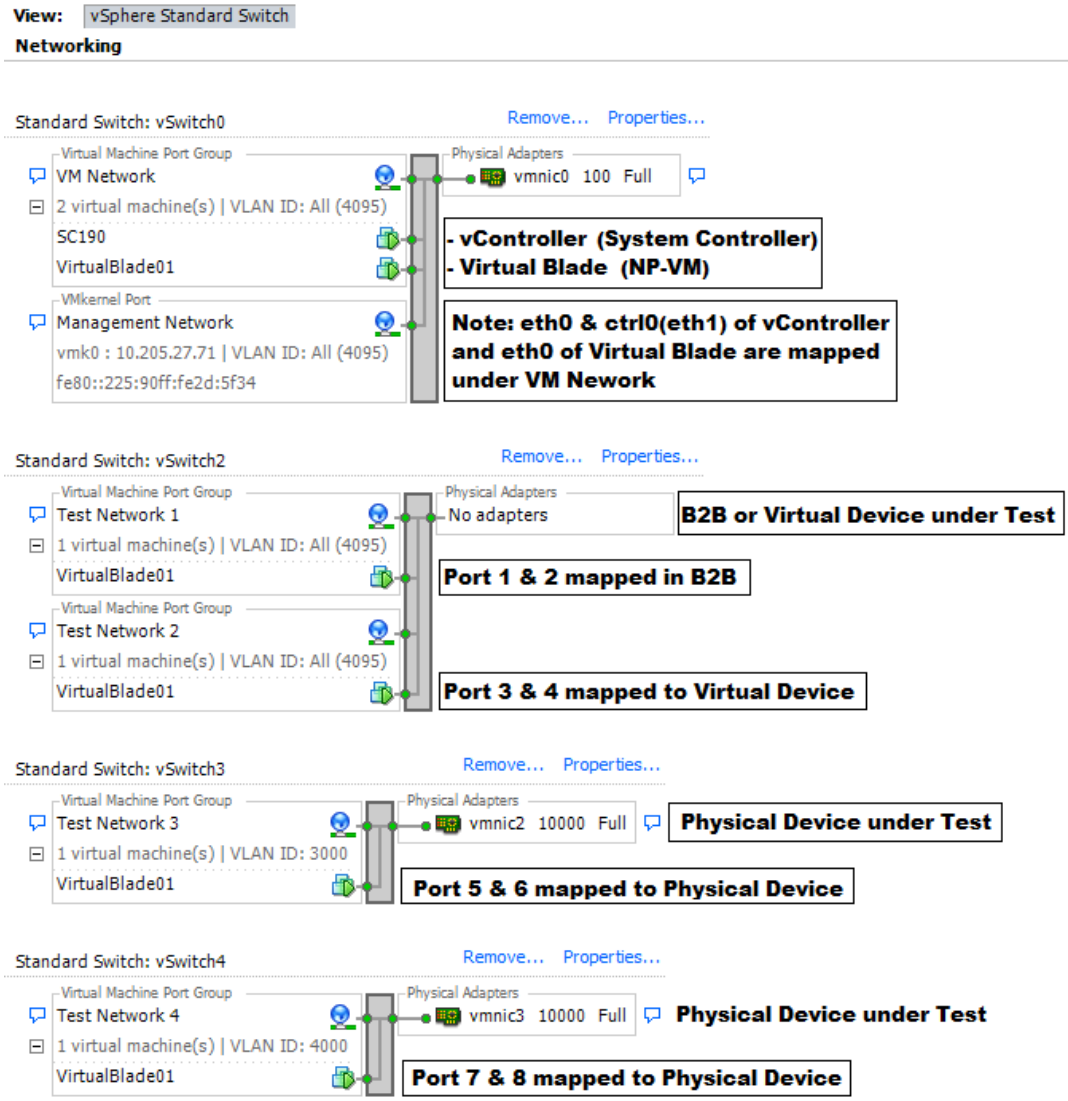
2. Select **Configuration** > **Networking**.



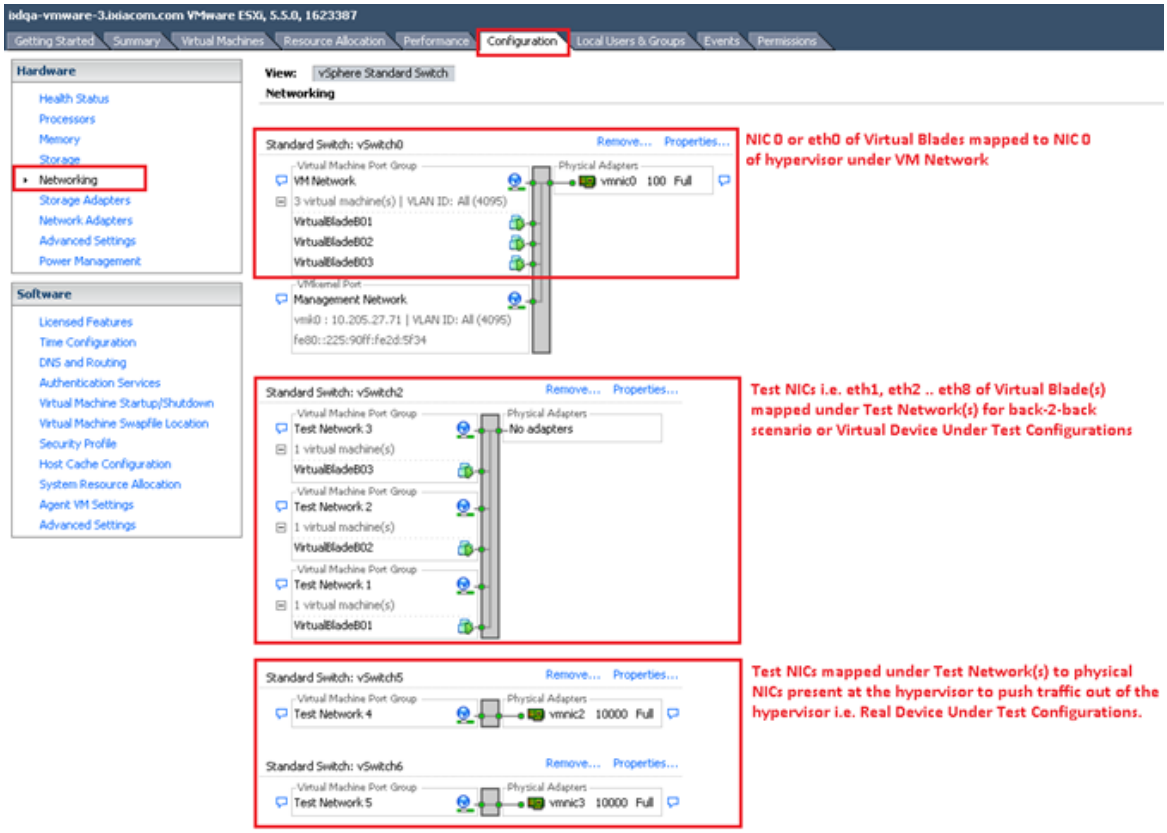
3. Add test networks to support a back-to-back/virtual Device Under Test (DUT) or a real DUT.

> **Note:** A Virtual DUT is not mapped to a physical Network Interface Card (NIC) of the hypervisor whereas a real DUT is mapped to a physical NIC.

Hypervisor Deployed with vController and vBlades



Hypervisor Deployed with vBlades Only



Promiscuous Mode Recommendations

Promiscuous Mode is an ESX server security policy setting that has two options, **Accept** and **Reject**. Enabling the **Accept** option allows a virtual machine to see all of the network traffic traversing a virtual switch. Enabling the Reject option allows a virtual machine to only see the packets that are destined for it. An example use case for enabling the Accept option is when testing an IDS or packet sniffer that needs to analyze all of the traffic on a network segment. The table below describes how the virtual machine Promiscuous Mode/BPS Network Neighborhood (NN) settings should be configured for packets to flow as expected.

vNIC Promiscuous Mode Setting	NN "Use vNIC MAC Address" Setting
Accept	Disabled or Enabled (because when the vNIC Promiscuous Mode is set to "Accept", all packets are passed regardless of this setting).
Reject	Enabled

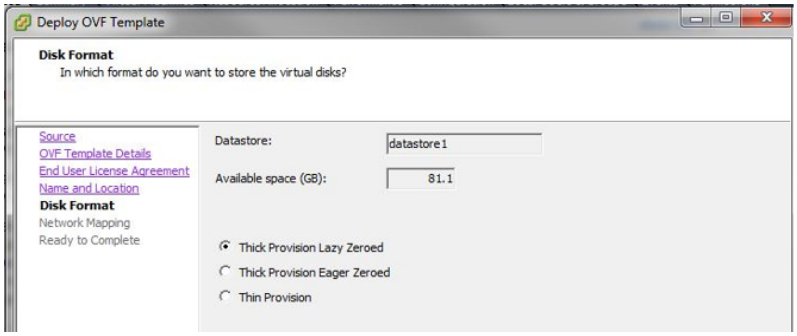
Note: In a 2-arm test configuration, packet traffic will flow regardless of the configuration settings described in the table above. A 2-arm test uses one Ixia test component (Session Sender, AppSim, etc.) to simulate both client and server in a scenario where traffic flows between Ixia ports (Ixia <-> Ixia).

Install BPS VE Controller on VMware

1. Get the BreakingPoint vController file from the Ixia website or Installation CD.
2. Log on to the hypervisor.
3. Select **File** > **Deploy OVF Template**.

   The **Deploy OVF Template** dialog box appears.
4. In the **Deploy OVF Template** dialog box, select **Browse** to locate the OVA file that has been saved to your computer. Alternatively, provide a URL address to install the OVF package from the Internet. Select **Next**.
5. Verify the **OVF Template Details** and select **Next**.
6. Accept the License Agreement. Select **Next**.
7. Specify a **Name** for the deployed template. Select **Next**.
8. Select the following **Disk Format**.
   - **Thick Provision Lazy Zeroed**

   Note: You can select the **Thin Provision** option if you need to save disk space.

   Select **Next**.
9. In the **Network Mapping** section, correctly map the **Source Networks** with the **Destination Networks**. Select **Next**.

   Note: A single interface will be selected by default.
10. In the **Ready to Complete** section, verify the **Deployment settings**.

    Select the **Power on after deployment** check box, if you want to automatically power on the virtual machines. If this box is not checked, you will have to manually power on the virtual machines post deployment. By default, this box is unchecked.

    Select **Finish** to start the OVA image file deployment.

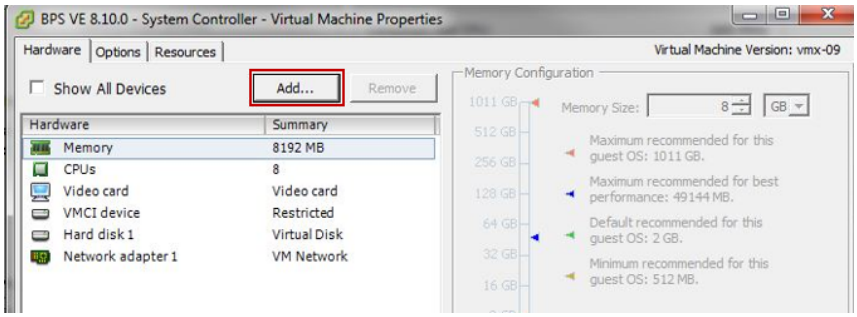
> **Note:** By default, the interface will request network configuration information (IP address, gateway, etc.) from a DHCP server. Alternatively, you can manually configure a static IP address as described in the section: Manually Set a Static IP for the Management Port on page 28.

11. Select **Finish**. The system starts the deployment of the BPS Controller in the hypervisor.
12. To add an additional interface to the vController perform the following steps:

> **Note:** Adding an additional interface will allow you to deploy the BPS VE controller in environments where the external/public network used to access the web interface is separated from the internal/private network used for chassis backplane communication.

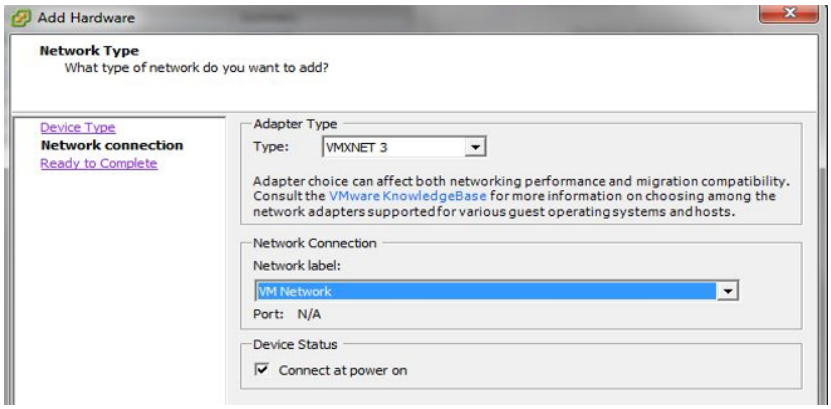
a. Power OFF the vController.

b. Edit the Virtual Machine options.



c. Select **Add**.

d. Select **Ethernet Adapter** as the Device Type. Select **Next**.



e. Select **VMXNET 3** as the Network Type. Select **Next**.

f. Select **Finish**.

g. Power ON the vController.

The vController will now operate with two interfaces.

13. Upon completion, you can Deploy and Assign vBlades.

KVM Installation

This section describes how to install BPS VE on KVM over CentOS or Ubuntu.

Install on KVM

This section describes how install BPS VE on KVM.

Note: This same procedure can be used to install the BPS vController on KVM and to perform the manual install of a BPS vBlade on KVM.

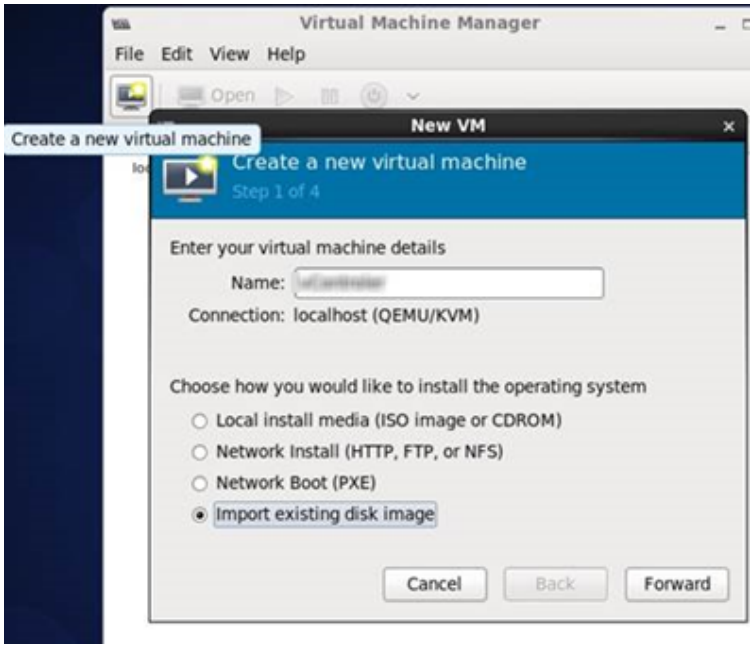
Note: To deploy a vBlade with SR-IOV Virtual Functions see, Deploy a vBlade with SR-IOV Virtual Functions.

Note: To install the **vController**, use the following file: Ixia_BreakingPoint_**Virtual_Controller**_x.x.x_EA_KVM.qcow2.
To manually install a **vBlade**, use the following file: Ixia_BreakingPoint_**Virtual_Blade**_x.x.x_EA_KVM.qcow2.

Note: Whenever you deploy a new vController or vBlade on a system, do not use the same image that was used during an earlier deployment on the system. Make a copy of the original qcow2 image and use the copied image for deployment. Using the same qcow2 image for multiple deployments may corrupt the image. Attempts to use the same image for multiple deployments will result in the message shown below. If you receive this message, reply **No,** and follow the procedure described earlier in this note.

To Deploy a BPS vController or vBlade:

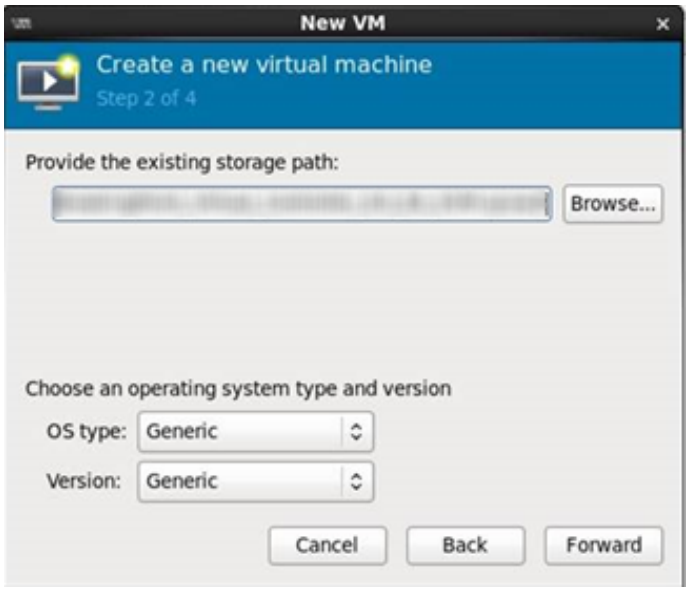
1. Download the required qcow2 image described above from the Ixia Downloads & Updates web page or from the installation CD.
2. Copy the qcow2 image to the KVM system.
3. Open the system's Virtual Machine Manager.
4. Select **Create a new virtual machine**. The window for configuring Step 1 displays.



a. Enter a name in the **Name** field. For example, if you are installing a vController, the Name could be "vController1", for a vBlade the name could be "vBlade1", etc.

b. Select **Import existing disk image**.

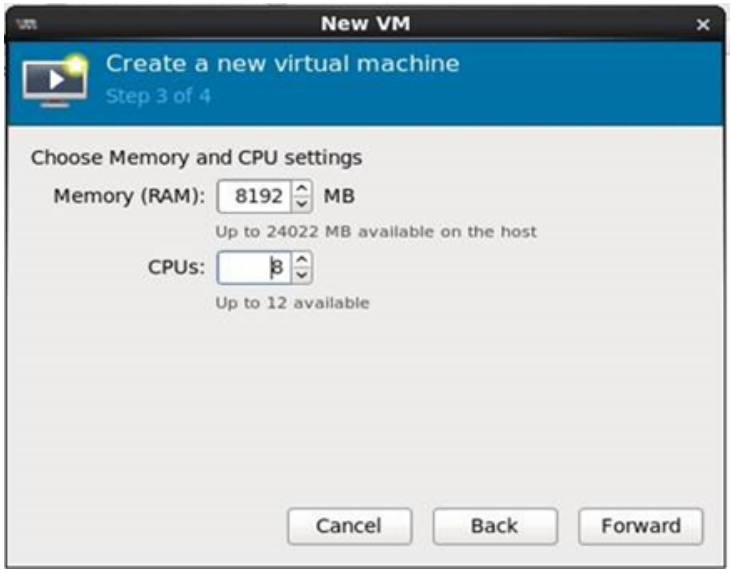
c. Select **Forward**. The window for configuring Step 2 displays.



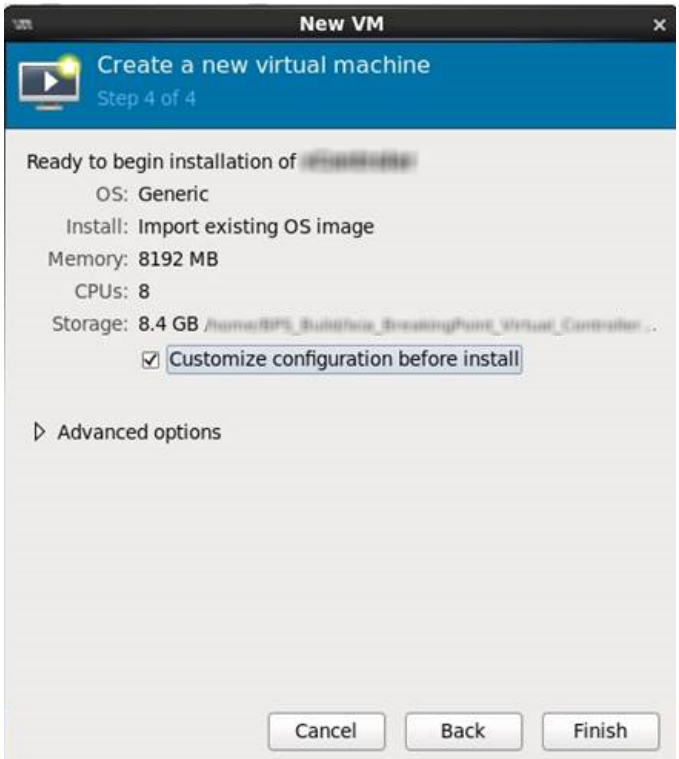
d. Configure the **Provide the existing storage path** field by selecting **Browse** and selecting the Ixia_BreakingPoint_**Virtual_Controller**_x.x.x_EA_KVM.qcow2 image.

e. Select **Forward**. The window for configuring Step 3 of 4 displays.

5. Choose **Memory** and **CPU** settings. For example, 8GB/8CPUs for a vController or 8GB/4CPUs for a vBlade. You can also reference Hardware Requirements on page 5 for more information.

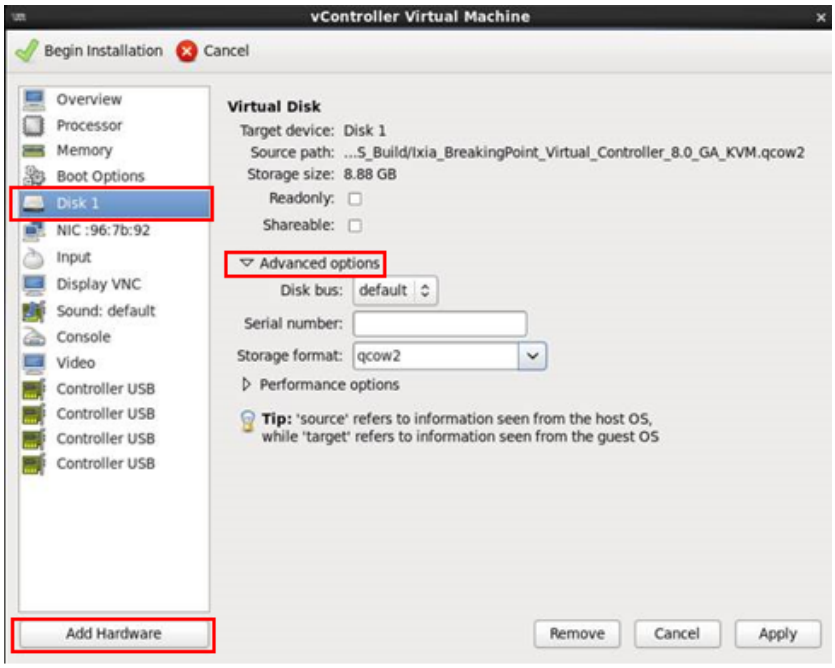


a. Configure Memory (RAM).
b. Configure number of CPUs.
c. Select **Forward**. The window for configuring Step 4 displays.
6. Select **Customize configuration before install**.



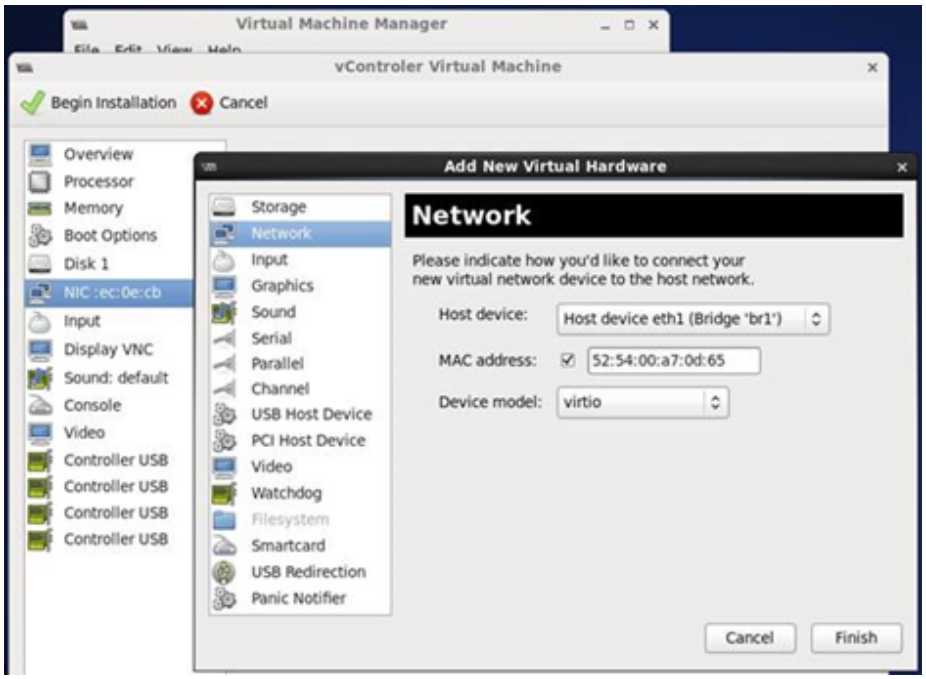
a. Select **Finish**. You will be returned to the vController Virtual Machine window.

7. Select **Disk 1**.



a. Expand the **Advanced Options** section and configure the **Storage format** as “qcow2”.

b. Select **Apply**.



8. Add the NICs that are required for testing.

a. Configure the NIC driver as "virtio".
b. Select **Finish**. You will be returned to the vController Virtual Machine window.
9. Select **Begin Installation**. Wait for the vController or vBlade to load.

vBlades must be assigned before they can be used for testing.

Deploy and Assign vBlades

vBlades can be deployed on various hypervisors using the BPS VE UI or with a BPS VE vBlade installation file and your own automation/management tools.

There are 3 vBlade deployment options:

Automatic vBlade Deployment for VMware or KVM

(Using vController VM Deployment Wizard)

Manual vBlade Deployment for VMware

Manual vBlade Deployment for KVM

After vBlades are successfully deployed, see the Manage vBlades section to learn how to discover, delete and unassign vBlades.

Automatic vBlade Deployment

Note: This procedure applies to both VMware ESXi and KVM hypervisor deployments. It does not require any additional vBlade installation images for either hypervisor.

Log on to the BPS VE UI:

1. Find the IP address of the vController.
2. Enter the vController IP address into the URL field of your HTML browser.
3. Enter a **Username** and **Password**. The default username is "`admin`". The default password is "admin".

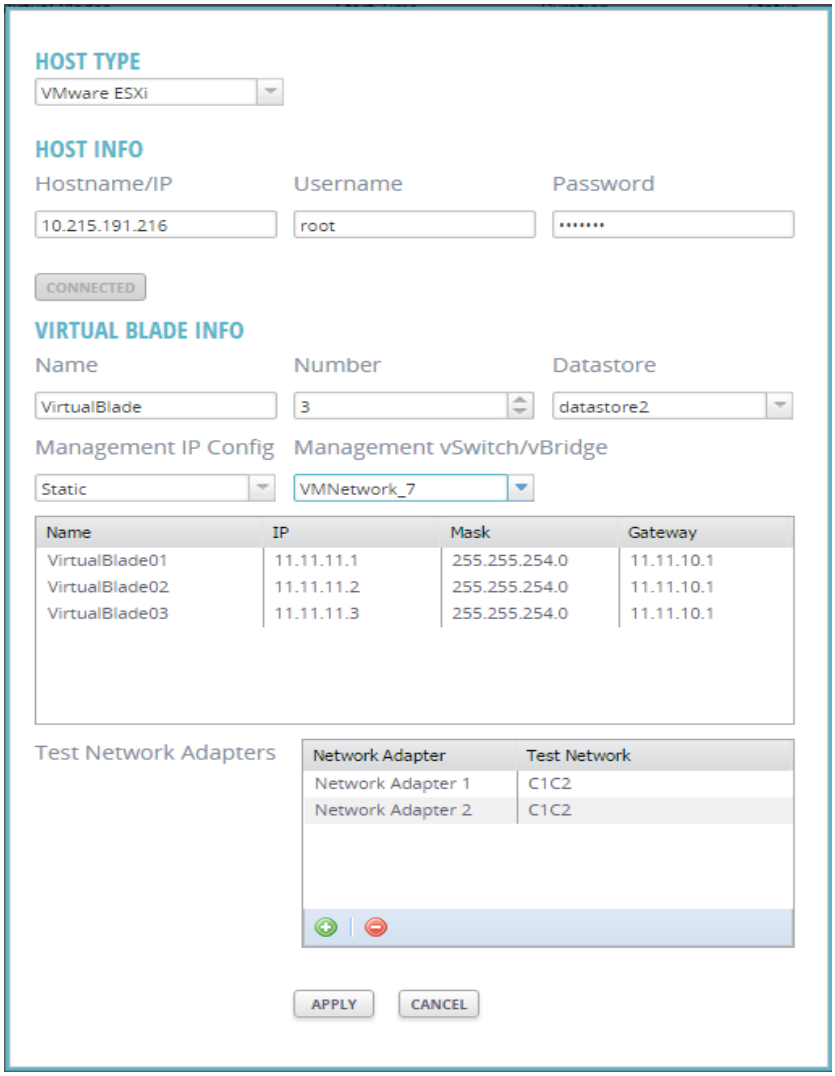
Create a Virtual Blade (vBlade)

1. After logging on to the BPS VE UI, select the **Administration** link in the upper right corner of the window.
2. Select **VM Deployment** > **Create Virtual Blades** > **Configure Virtual Blade**.

   Note: For VMware: To access the hypervisor, make sure to enable the ssh service in all target hypervisors (which is configured in **vSphere** > **Security Profile** > **SSH**).

   A dialog box displays the vBlade settings as shown in the image below. For setting descriptions, refer to the Virtual Blade Configuration Parameters on page 26 table.
3. Select the **Host Type** from the drop-down list.

4. In the **HOST INFO** section, enter the **Hostname/IP** of the hypervisor where you want to deploy the VM.
5. Enter the correct **Username**/**Password** of the target server where the vBlade will reside and select **Connect**.



6. Enter the name for the vBlades in the **Name** field.
7. Enter the number of vBlades required in the **Name** field.
8. Select Static or DHCP from the **Management IP Configuration** drop-down list.

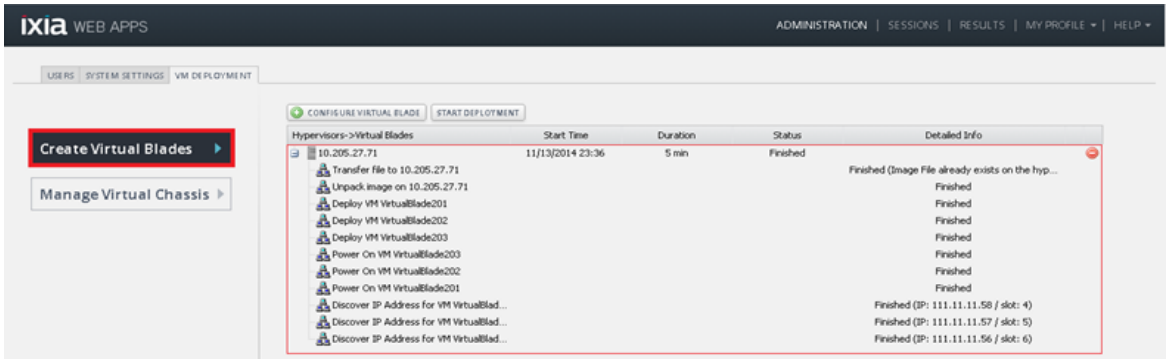
> **Note:** If you select the DHCP **IP Configuration** option, a DHCP server will be required in order to provide IP addresses to the BPS VE vController and vBlade interfaces.

> **Note:** If you select the Static **IP Configuration** option, default IP addresses are assigned to the vBlades in ascending order based on the network address of the vController (as shown in the image above). You can edit the vBlade IP addresses by double-clicking the IP Address field.

9. Select the **Datastore**. The network topology present in the hypervisor along with the **Datastore** (HDD) details are available in the Virtual Load Module Info section.
10. Select the required **Management Network** for the vBlades.
11. In the **Test Network** list, select the **Network Adapter** and map them to the relevant **Test Network**.

    vBlades can support two to eight vPorts. vPorts are directly mapped with a Network Adapter. vPort-1 refers to Network Adapter 1, vPort-2 refers to Network Adapter 2 and so on. Assign a Test Network (created in the vSwitch and Network Configuration section) to the respective vPort.
12. Select **Apply**.

    The status of the deployment is displayed (as shown in the image below). If errors occur, an error message will display in a pop-up. After successful validation, a new vBlade entry is created.



Virtual Blade Configuration Parameters

Parameter	Description
Host Type	Select the type of host you will be installing a vBlade on.
HOST INFO	
Hostname/IP	Enter the host name or IP of the hypervisor.
Username	Enter the valid user name to log on to the hypervisor.
Password	Enter the valid password to log on to the hypervisor.
VIRTUAL LOAD MODULE INFO	
Name	Enter a name for the vBlade.

Parameter	Description
Number	Enter the number of vBlades (virtual machines) to be deployed.
Management IP Configuration	Select a DHCP or Static IP configuration.
Datastore	Datastores are logical containers, analogous to file systems, that hide specifics of each storage device and provide a uniform model for storing virtual machine files. Datastores can also be used for storing ISO images, virtual machine templates, and floppy images.
Management vSwitch/vBridge	The **Management vSwitch/vBridge** is used for the internal communication between vController and vBlades. It must be in the same IP subnet with the vController internal management IP. Select at least two **Network Adapters** and map the **Test Network** to these adapters. The Test Network is used send and receives BPS VE test traffic.

Manually Set a Static IP for the Management Port

The management port IP address can be configured using the **setip** console command as shown in the image below. The command allows you to set the static IP address for the management interface of a vController or vBlade.

The following login is required:

user: netadmin

password: netadmin

```
metadmin:~$ setip -h
usage: setip [-h] -iface IFACE [-dhcp] [-ip IP] [-mask MASK] [-gw GW]

Sets the IPv4 address for the specified interface.

optional arguments:
  -h, --help    show this help message and exit
  -iface IFACE  Interface, e.g. eth0/ctrl0.
  -dhcp         DHCP/Static, if dhcp, following parameters are ignored.
  -ip IP        IP Address, e.g. 192.168.10.15
  -mask MASK    Netmask, e.g. 24; within range 1 to 31
  -gw GW        [Optional] Gateway, e.g. 192.168.10.1
metadmin:~$
metadmin:~$ setip -iface ctrl0 -ip 192.168.10.15 -mask 24 -gw 192.168.10.1_
```

Note: The interface names that can be used with this command are:

- For vController with a single management interface: ctrl0.
- For vController with 2 management interfaces: eth0 (for external management) and ctrl0 (for internal management).
- For vBlade: eth0.

Find the BPS VE vController IP Address

The BPS VE vController IP Address can be used to access the BPS VE UI. To access the BPS VE UI, enter the controller IP address into the URL field of your HTML browser and proceed to Log on to the BPS VE User Interface on the next page.

To find the System Controller IP address:

- Access the Console on the vController (System Controller) Virtual Machine (VM)
- Run the networkInfo command

Access the Console on VMware

1. Start the Console from vSphere to System Controller Virtual Machine (VM).
2. Log on using the proper credentials. For example:
   User ID - admin
   Password - admin
   The system displays the BPS prompt.
3. Run the networkInfo command to display the vController (System Controller) IP Address.

Access the Console on KVM

1. Connect to the Console on the vController Virtual Machine (VM).

   Note: ttyS0 will need to be enabled within the VM if it is not currently enabled.

2. Log on to the system using the proper credentials. For example:
   User ID - admin
   Password - admin
3. Run the networkInfo command to display the vController (System Controller) IP Address.

Run the networkInfo Command

1. Type the following command at the prompt.
   ```
   BPS> networkInfo
   ```
   The system displays following information.
   ```
   dhcp="true"
   hostname="localhost.localdomain.bpointsys.int"
   ip="10.200.225.38" <==== IP of System Controller
   netmask="22"
   gw=""
   currip="10.200.225.38"
   ........................
   ```

Log on to the BPS VE User Interface

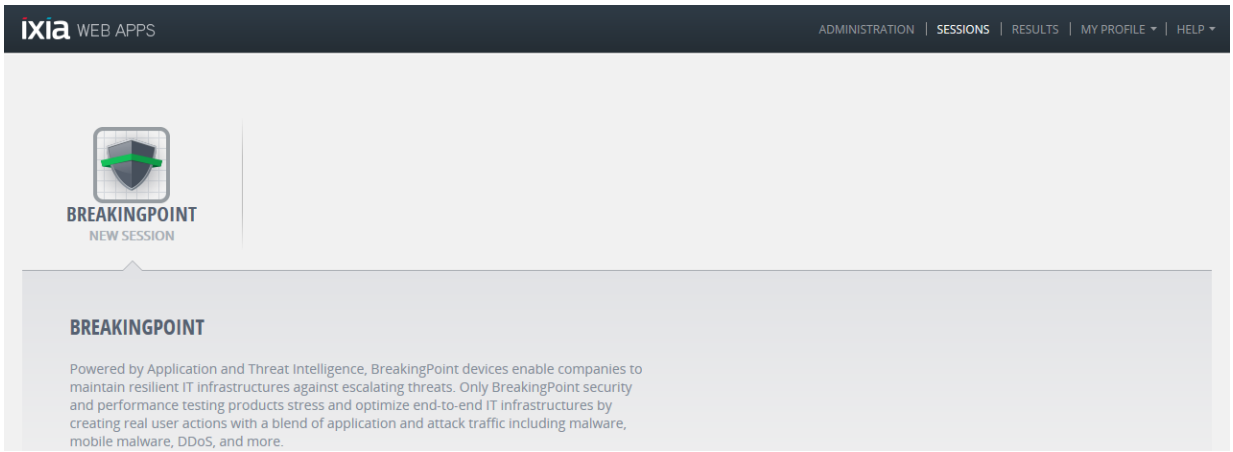
The BPS VE is used to manage BPS VE and Deploy vBlades.

To log on to the BPS VE user interface (also known as Ixia Web Apps), perform the following tasks:

1. Open a web browser, type the vController IP address in the URL field, and press Enter.
   The log on window appears.
2. In the **Username** field, type your user ID. The default username is "admin".
3. In the **Password** field, type your password.
   The default password is "admin".

4. If you want the browser to remember the log on credentials, select the **Remember me** check box.
5. Select **Login**.

   The **Ixia WEB APPS** window opens as shown in the figure below.



The Web Administration page consists of links as listed and described in the following table.

Links	Description
Administration	Perform administration tasks. For example, creating/managing user accounts, manage the Ixia Web Application and manage BreakingPoint in the Virtual Environment (VE).
Sessions	Open the BreakingPoint Control Center to manage the BreakingPoint sessions (Individual or multiple instances of running tests).
Results	View the list of completed and currently running tests.
My Profile	View and edit the properties of your account. For example, your user name and password can be modified.
Help	View the product user guides, download the latest software, and perform system diagnostics.

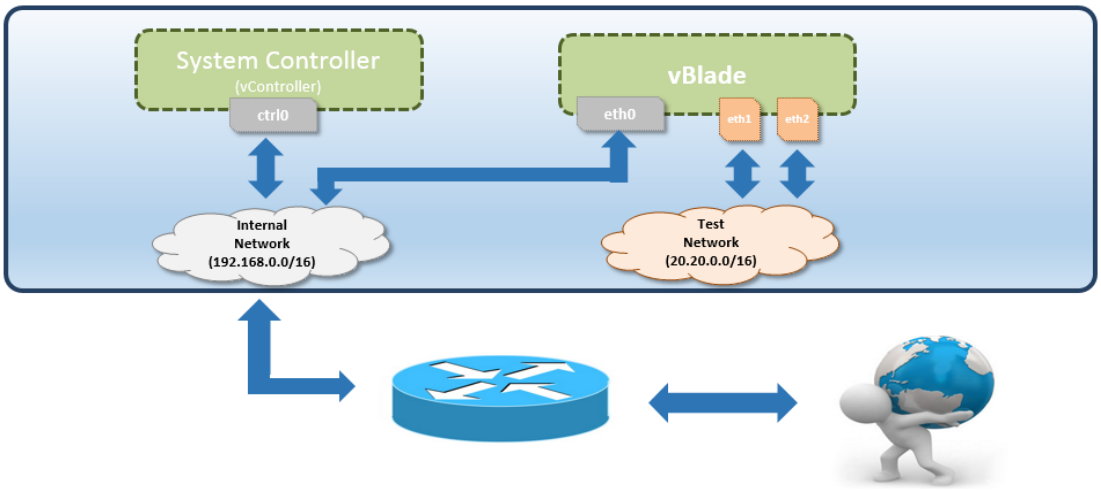
Install BPS VE using OpenStack

OpenStack is a free and open-source software platform for cloud computing. This section provides a detailed graphical example of BPS VE installation and setup using OpenStack.

> **Note:** Testing was executed only with virtIO disk bus types. Sometimes due to the slowness of other disk bus types (e.g. IDE) you may experience slowness and instability of the Virtual Machines. No testing was executed with remote storage in-house. Sometimes due to the slowness of the remote storage you may experience slowness and instability of the Virtual Machines.

Network Topology

The topology shown in the image below will be used for the example OpenStack BPS VE Installation.

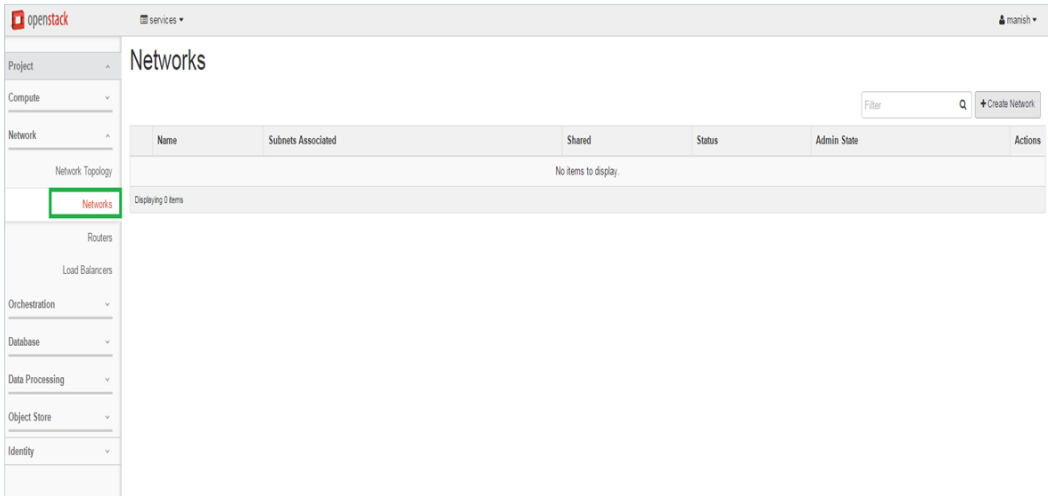


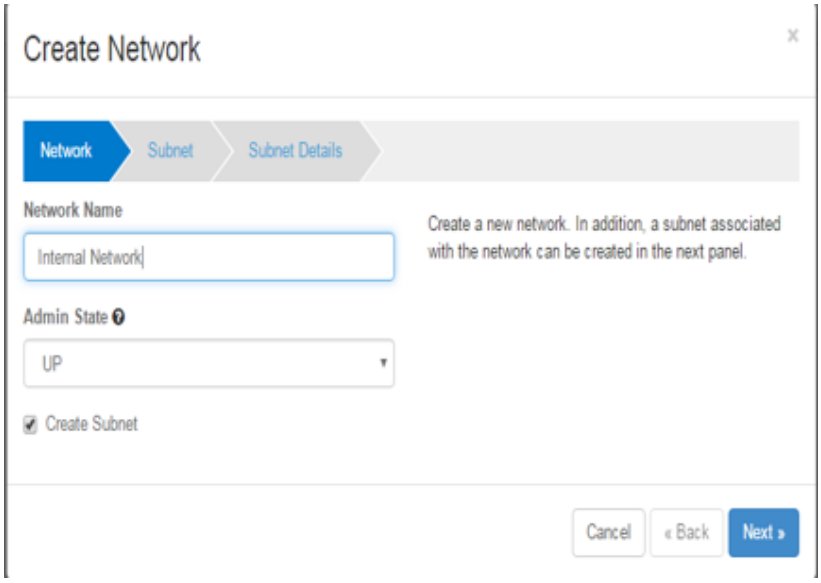
OpenStack Login

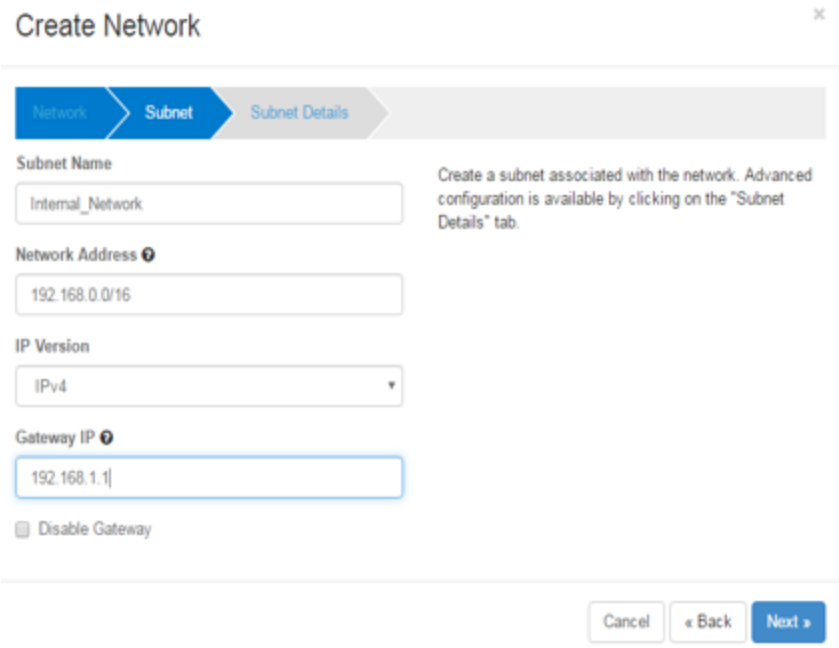
Log in to your OpenStack dashboard.

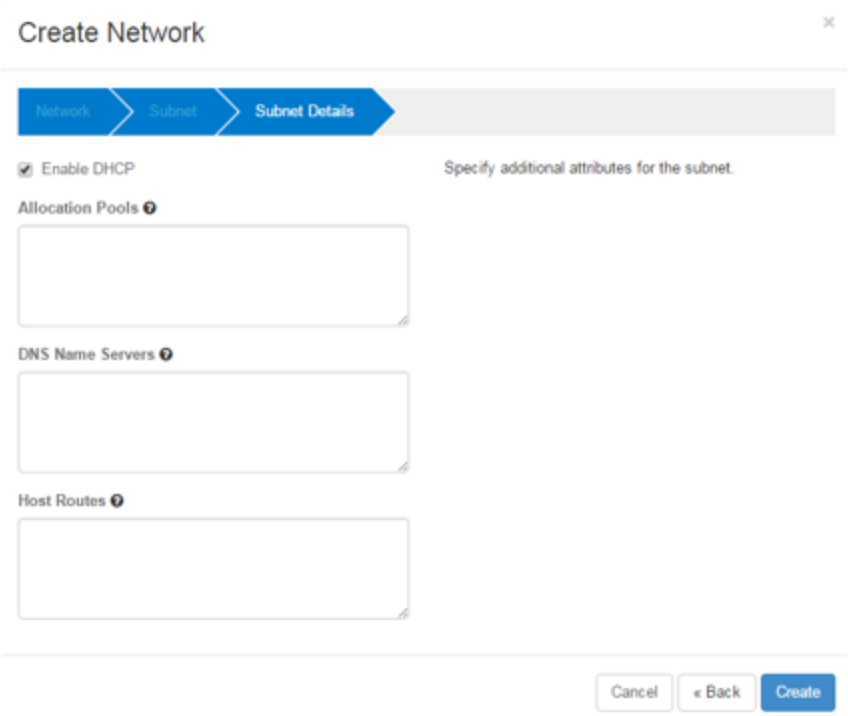
Create Networks

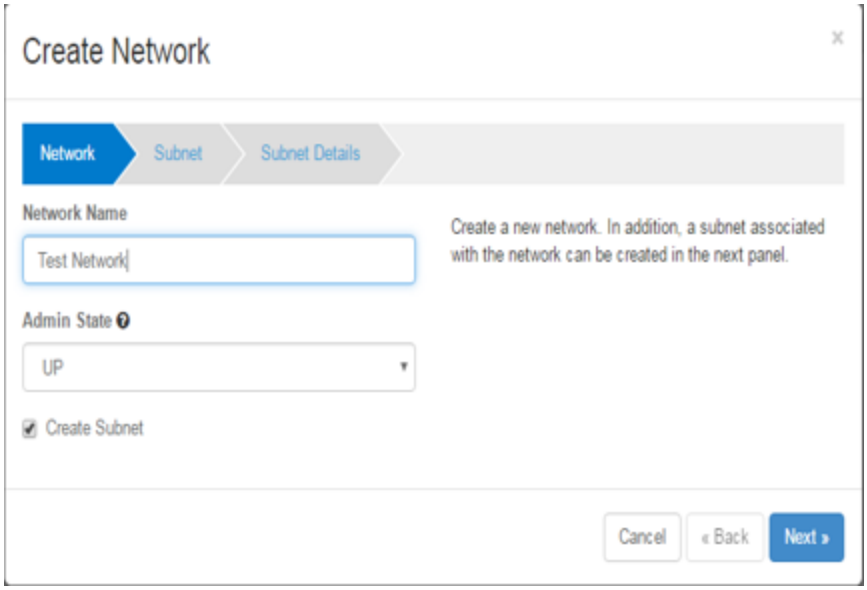
Create the required networks based on the Network Topology.

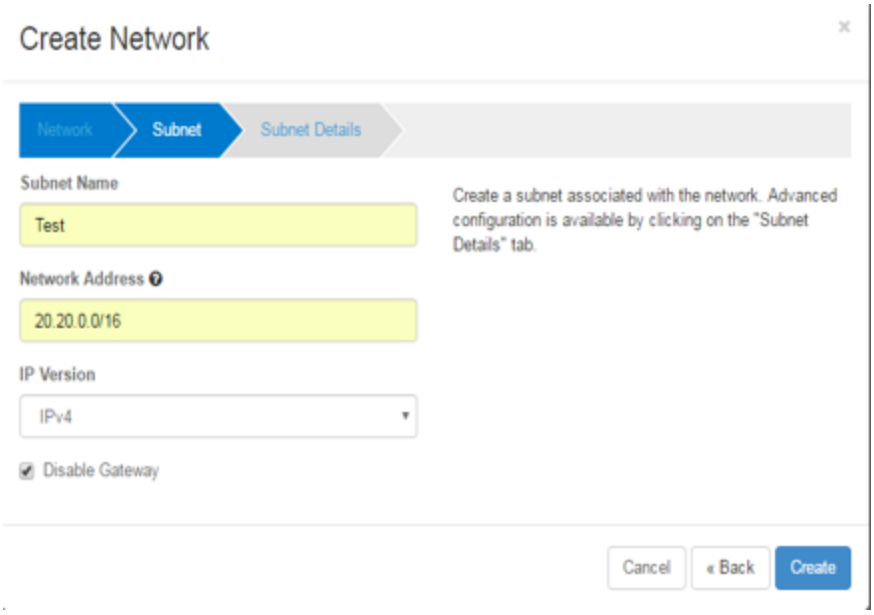


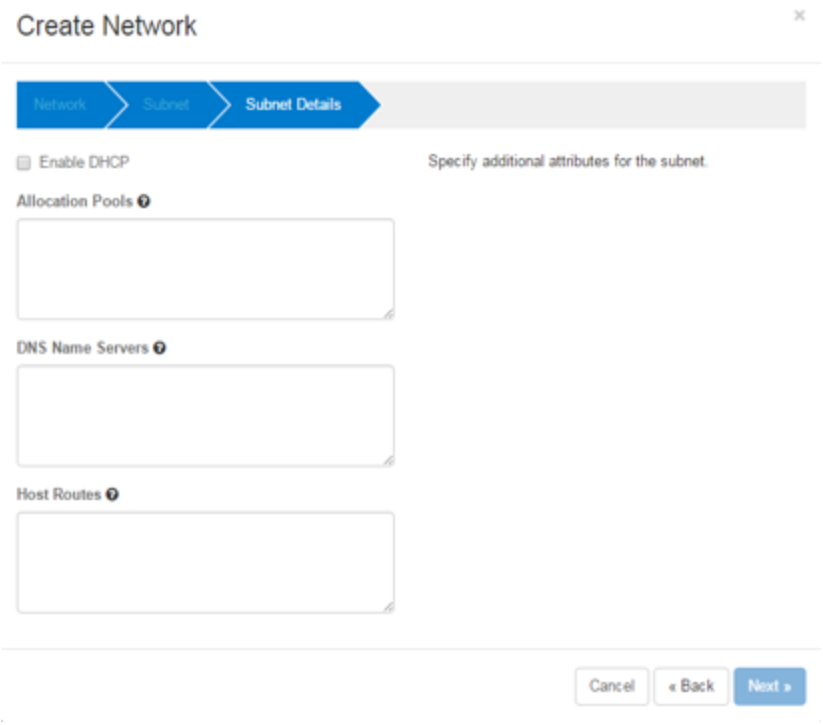

Create Network
Network
Subnet
Subnet Details
Network Name
Internal Network
Create a new network. In addition, a subnet associated with the network can be created in the next panel.
Admin State
UP
Create Subnet
Cancel
« Back
Next »

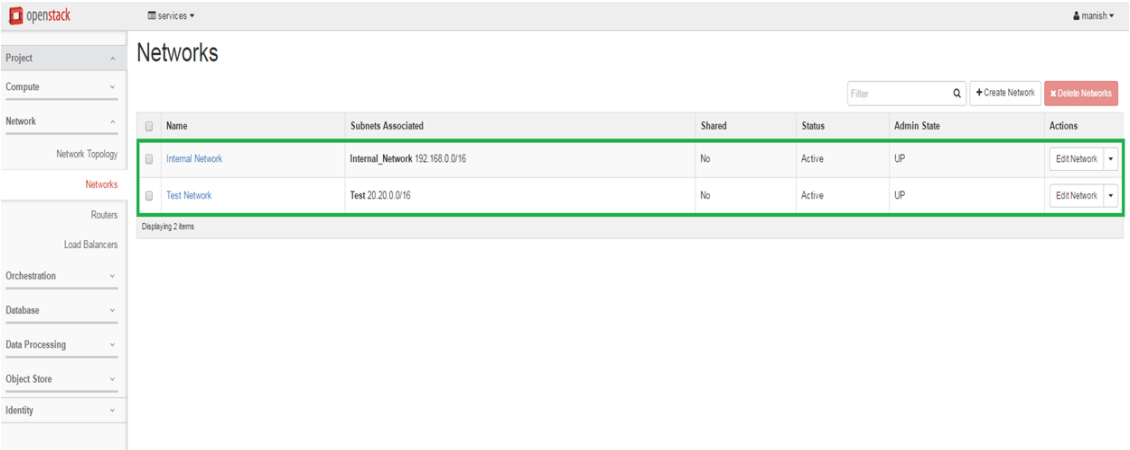

Create Network
Network
Subnet
Subnet Details
Subnet Name
Internal_Network
Create a subnet associated with the network. Advanced configuration is available by clicking on the "Subnet Details" tab.
Network Address
192.168.0.0/16
IP Version
IPv4
Gateway IP
192.168.1.1
Disable Gateway
Cancel
« Back
Next »


Create Network
Network
Subnet
Subnet Details
Enable DHCP
Specify additional attributes for the subnet.
Allocation Pools
DNS Name Servers
Host Routes
Cancel
« Back
Create

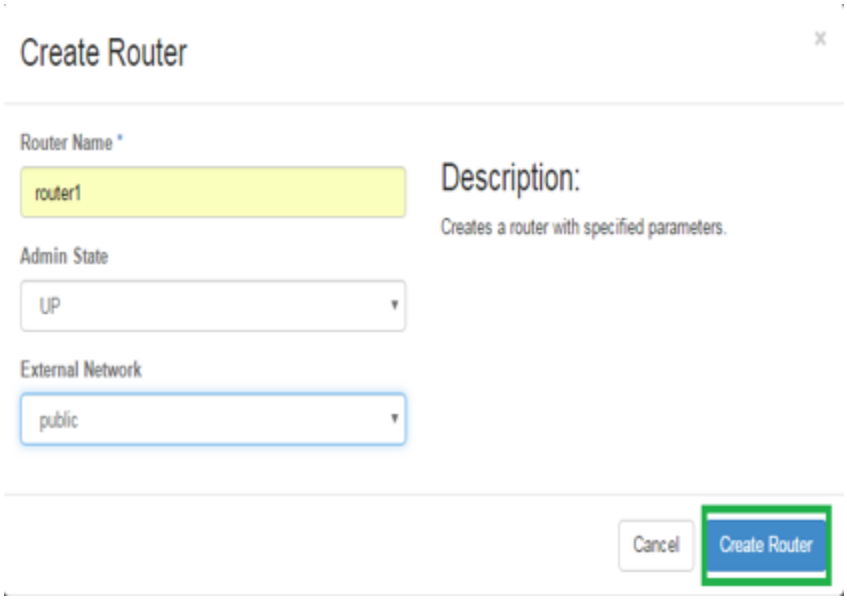

Create Network
Network
Subnet
Subnet Details
Network Name
Test Network
Create a new network. In addition, a subnet associated with the network can be created in the next panel.
Admin State
UP
Create Subnet
Cancel
« Back
Next »

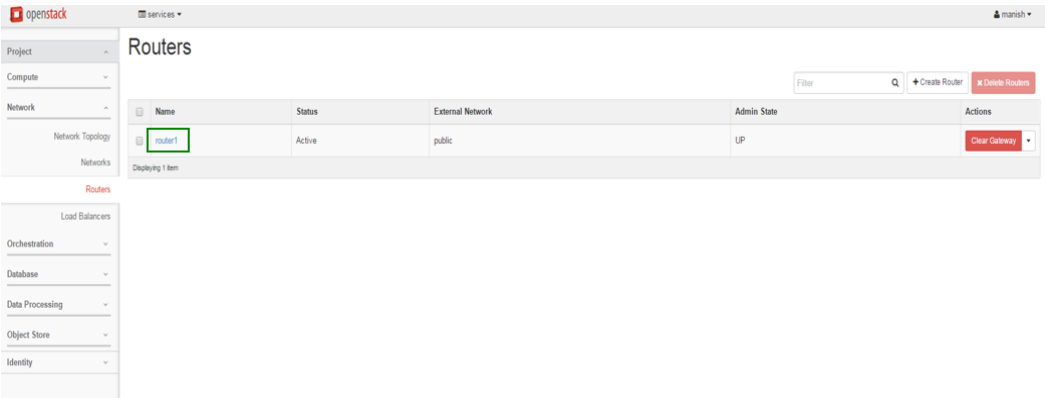

Create Network
Network
Subnet
Subnet Details
Subnet Name
Test
Network Address
20.20.0.0/16
IP Version
IPv4
Disable Gateway
Create a subnet associated with the network. Advanced configuration is available by clicking on the "Subnet Details" tab.
Cancel
« Back
Create

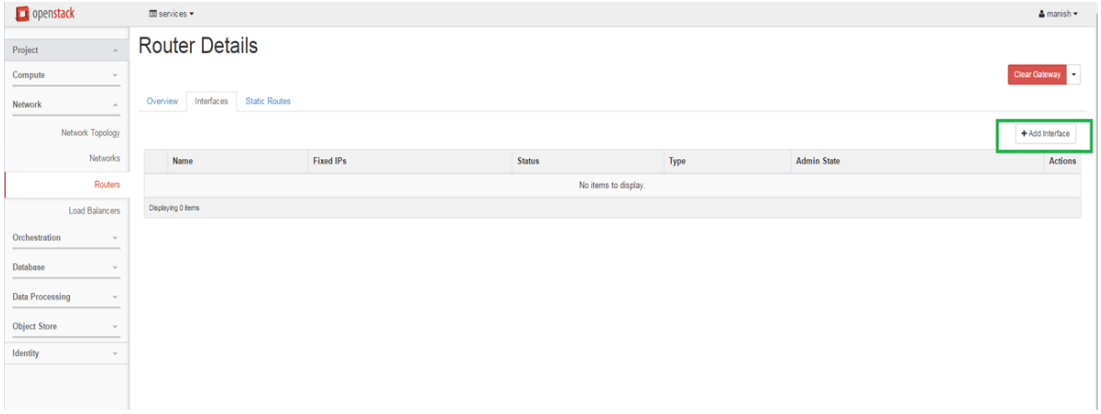

Create Network
Network
Subnet
Subnet Details
Enable DHCP
Allocation Pools
DNS Name Servers
Host Routes
Specify additional attributes for the subnet.
Cancel
« Back
Next »

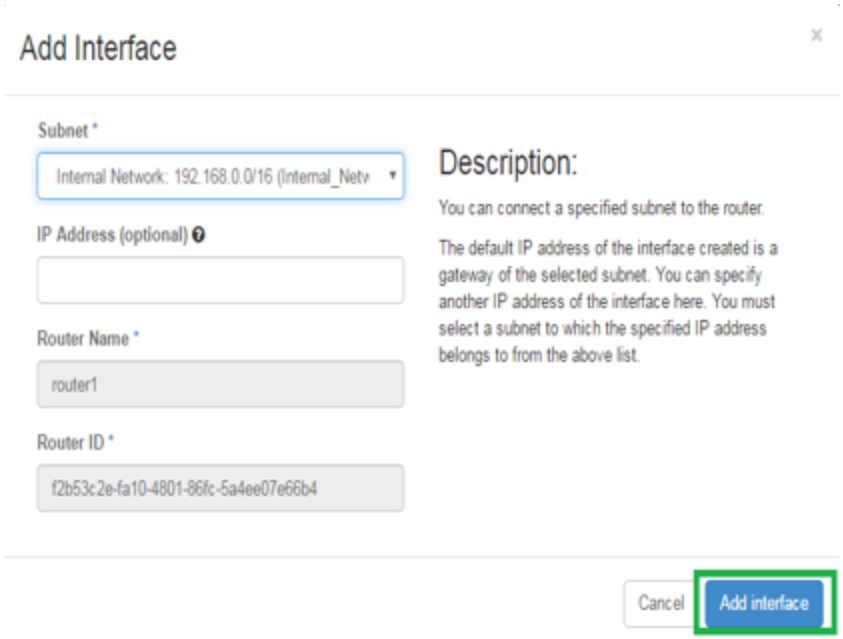


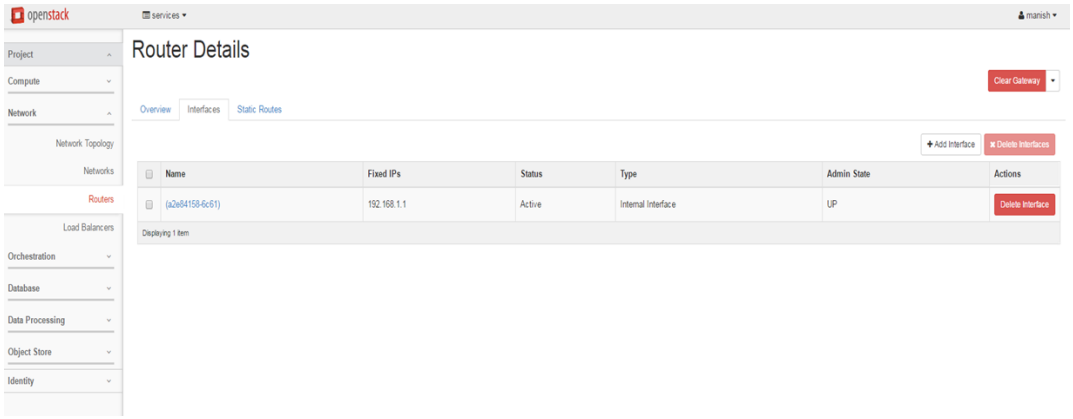
Create a Router





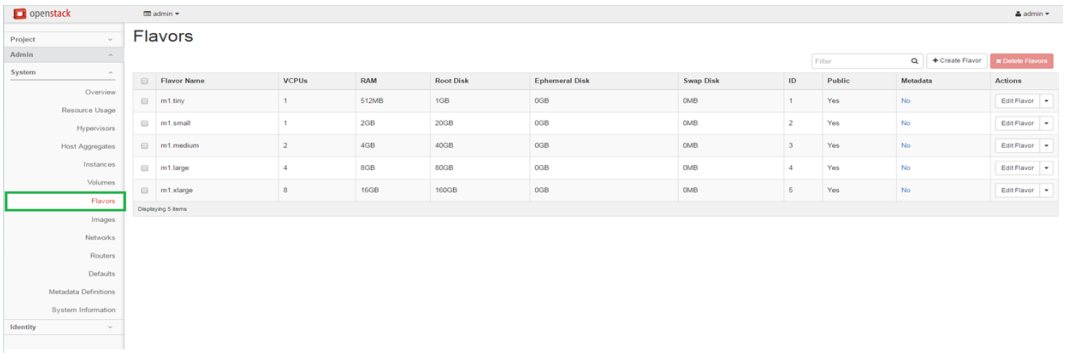

openstack
services
manish
Project
Compute
Network
Network Topology
Networks
Routers
Load Balancers
Orchestration
Database
Data Processing
Object Store
Identity
Router Details
Clear Gateway
Overview
Interfaces
Static Routes
+ Add Interface
Name
Fixed IPs
Status
Type
Admin State
Actions
No items to display.
Displaying 0 items


Add Interface
Subnet *
Internal Network: 192.168.0.0/16 (Internal_Netv
IP Address (optional)
Router Name *
router1
Router ID *
f2b53c2e-fa10-4801-86fc-5a4ee07e66b4
Description:
You can connect a specified subnet to the router.
The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs to from the above list.
Cancel
Add interface

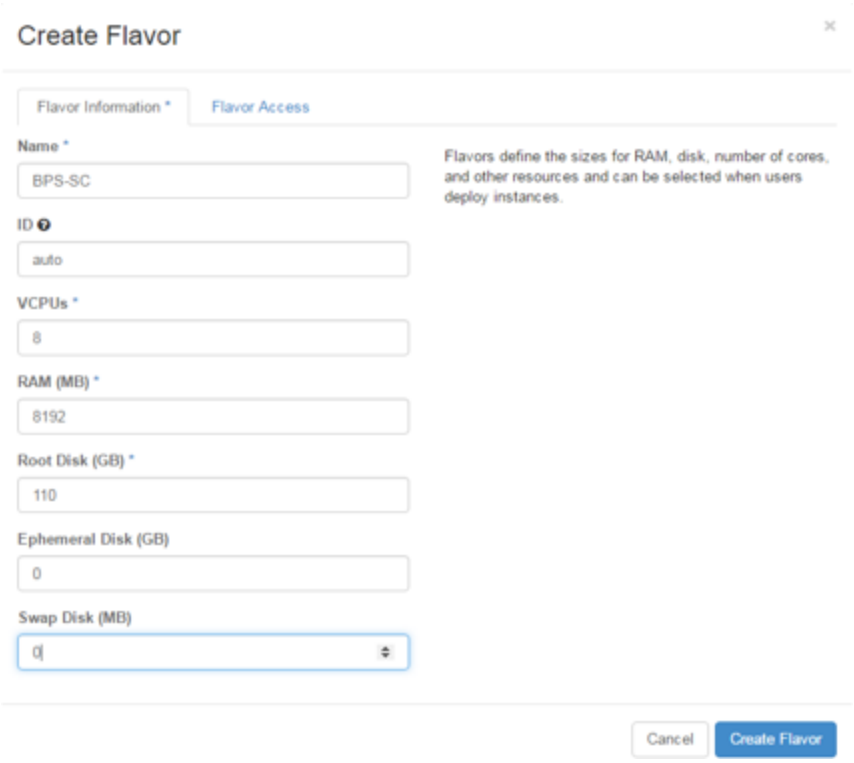

openstack
services
manish
Project
Compute
Network
Network Topology
Networks
Routers
Load Balancers
Orchestration
Database
Data Processing
Object Store
Identity
Router Details
Clear Gateway
Overview
Interfaces
Static Routes
+ Add Interface
Delete Interfaces
Name
Fixed IPs
Status
Type
Admin State
Actions
(a2e84158-6c61)
192.168.1.1
Active
Internal Interface
UP
Delete Interface
Displaying 1 item

Create Flavors

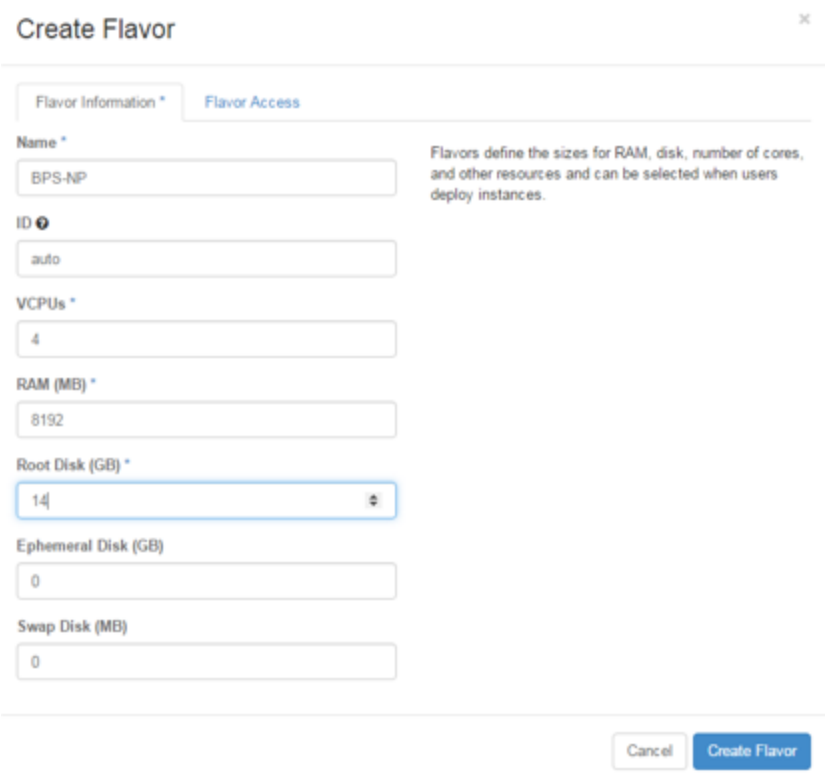
Note: Flavors can only be created using the Admin account.

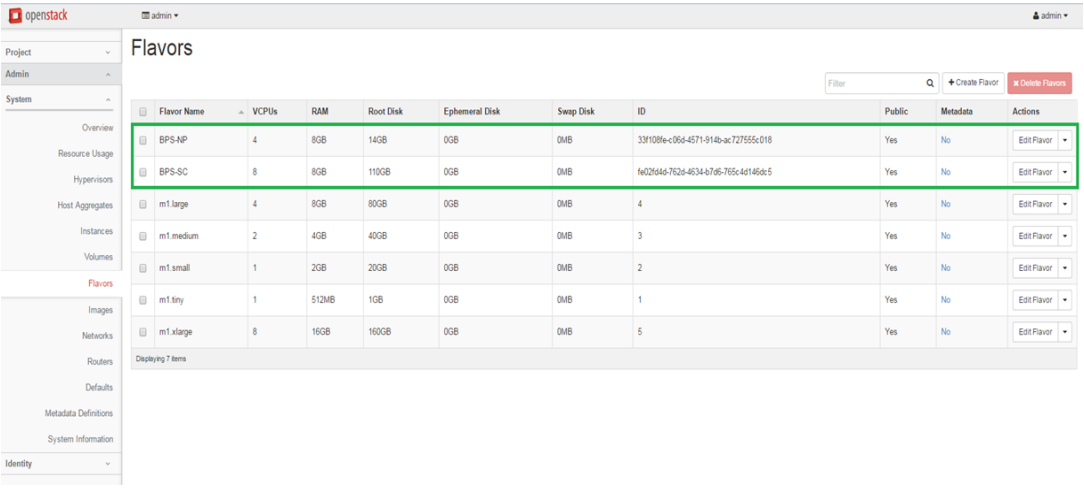


Note: The minimum Root Disk required to launch the System Controller (BPS vController) is 110 GB.



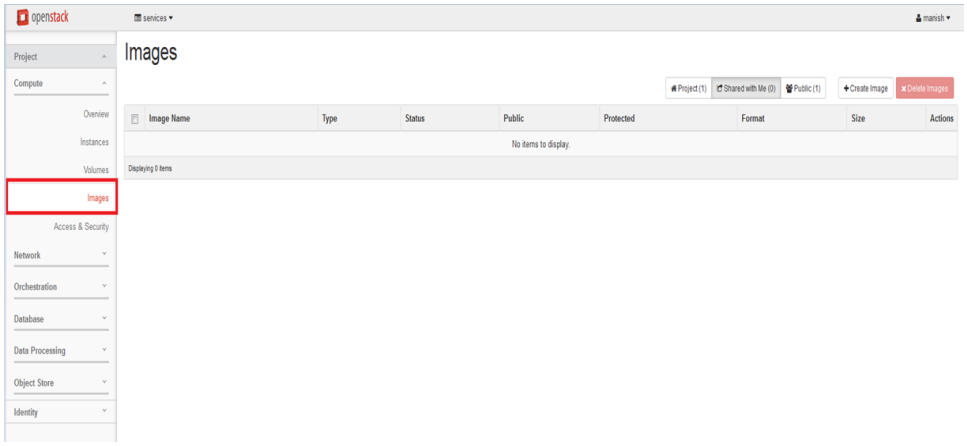
Note: The minimum Root Disk required to launch a virtual blade (BPS vBlade) is 14 GB.

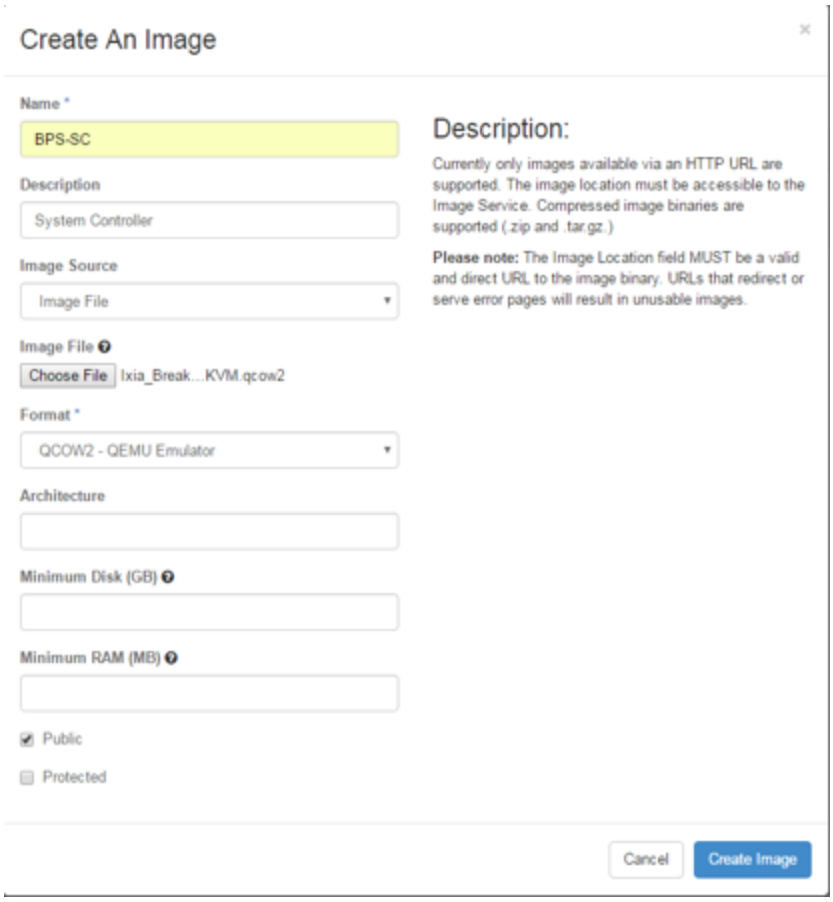


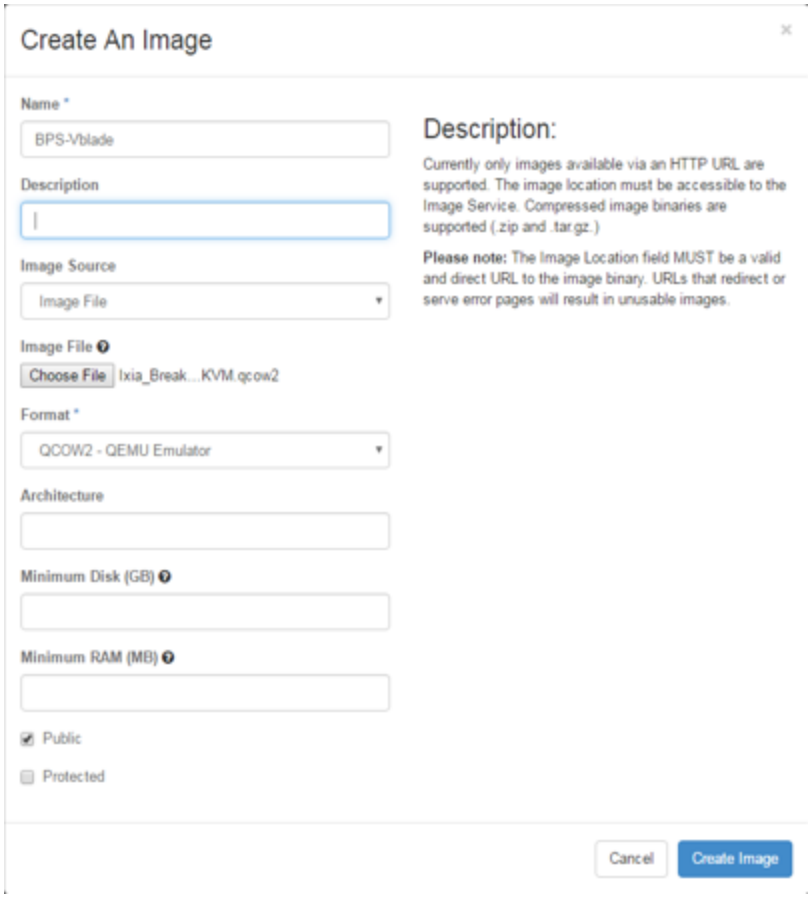


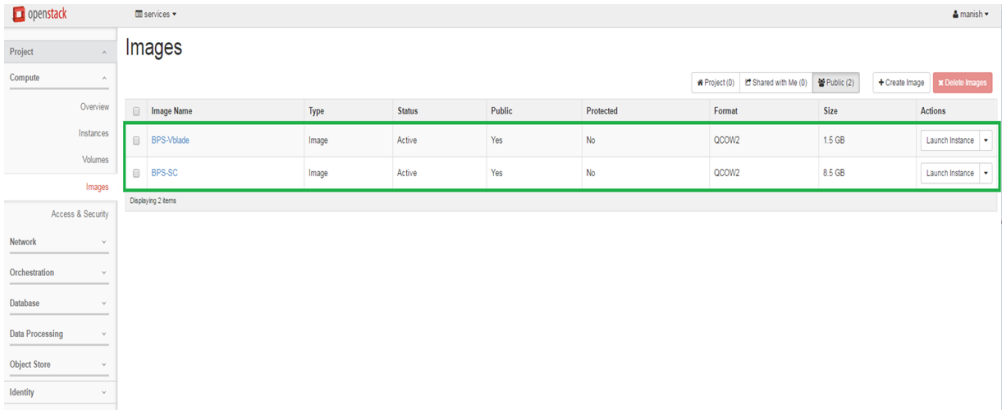
Add Images

Note: The BPS vController is also described as the System Controller.

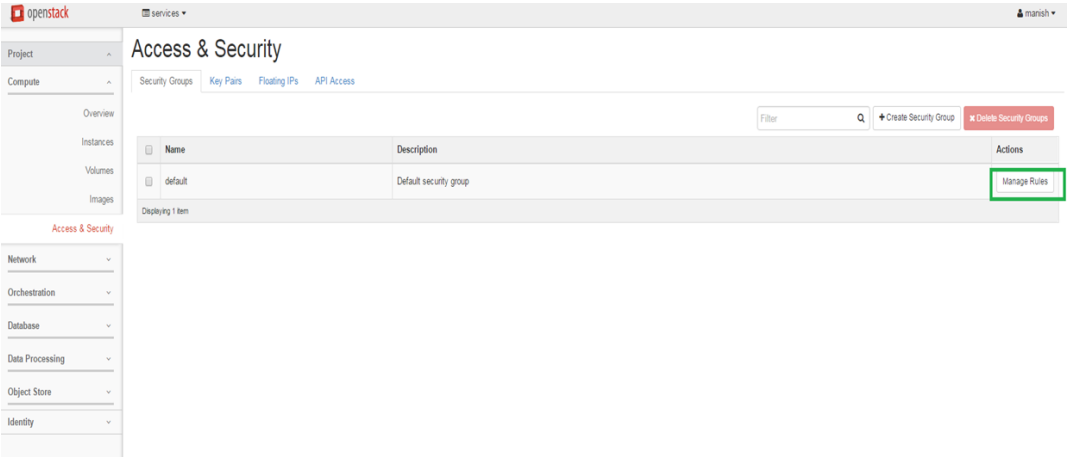

openstack
services
manish
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Database
Data Processing
Object Store
Identity
Images
Project (1)
Shared with Me (0)
Public (1)
Create Image
Delete Images
Image Name
Type
Status
Public
Protected
Format
Size
Actions
No items to display.
Displaying 0 items


Create An Image
Name *
BPS-SC
Description
System Controller
Image Source
Image File
Image File
Choose File
Ixia_Break...KVM.qcow2
Format *
QCOW2 - QEMU Emulator
Architecture
Minimum Disk (GB)
Minimum RAM (MB)
Public
Protected
Description:
Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)
Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.
Cancel
Create Image

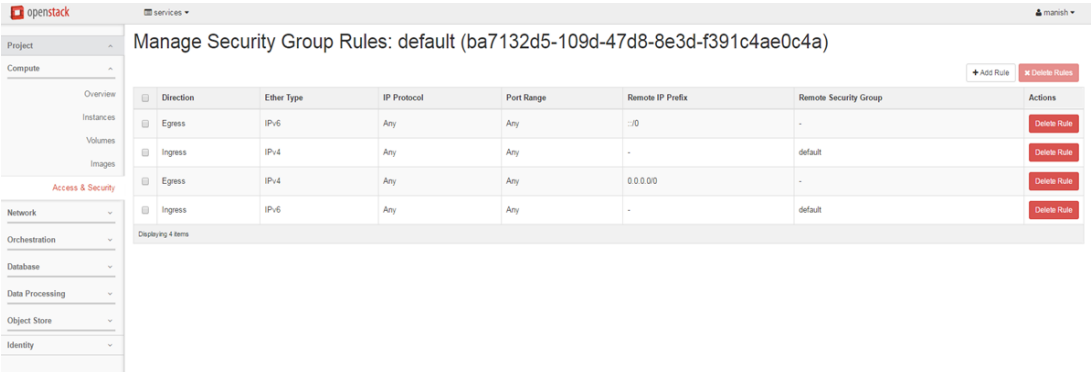

Create An Image
Name *
BPS-Vblade
Description
Image Source
Image File
Image File
Choose File
Ixia_Break...KVM.qcow2
Format *
QCOW2 - QEMU Emulator
Architecture
Minimum Disk (GB)
Minimum RAM (MB)
Public
Protected
Description:
Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)
Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.
Cancel
Create Image

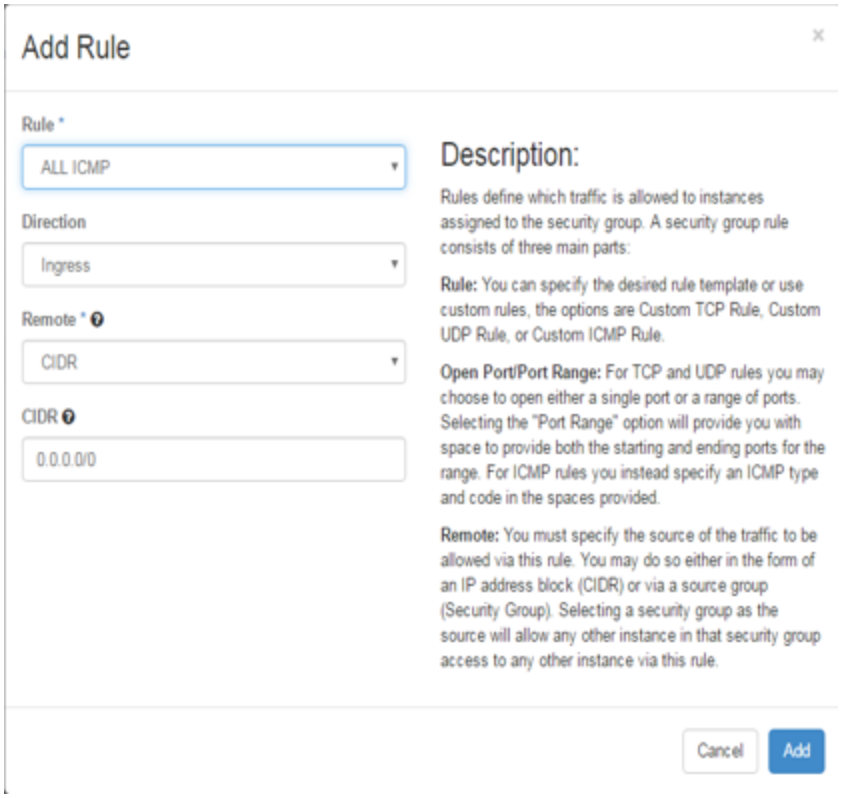

openstack
services
manish
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Database
Data Processing
Object Store
Identity
Images
Project (0)
Shared with Me (0)
Public (2)
+ Create Image
Delete Images
Image Name
Type
Status
Public
Protected
Format
Size
Actions
BPS-Vblade
Image
Active
Yes
No
QCOW2
1.5 GB
Launch Instance
BPS-SC
Image
Active
Yes
No
QCOW2
8.5 GB
Launch Instance
Displaying 2 items

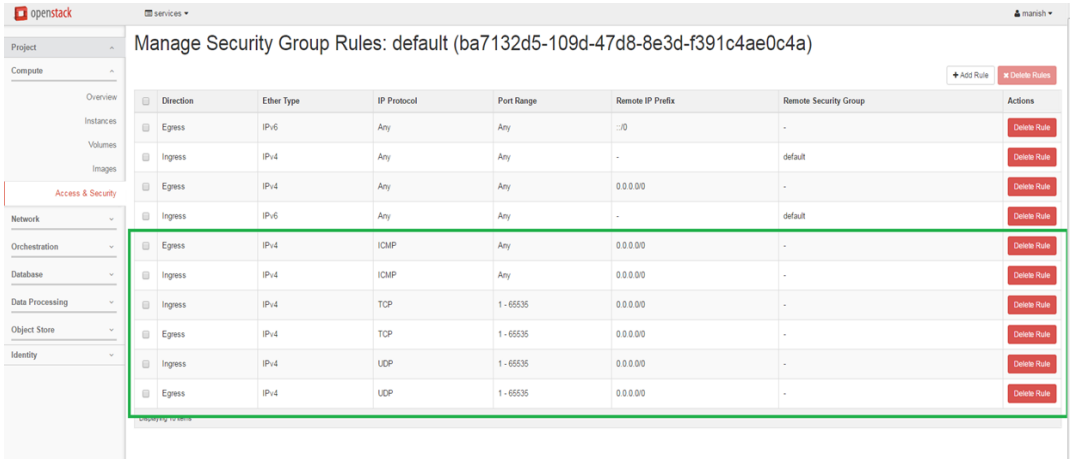
Security Group Management

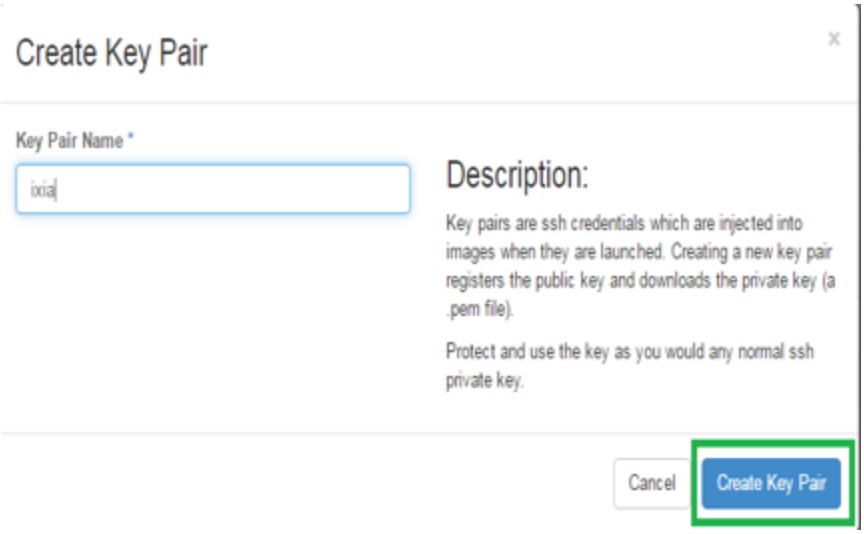


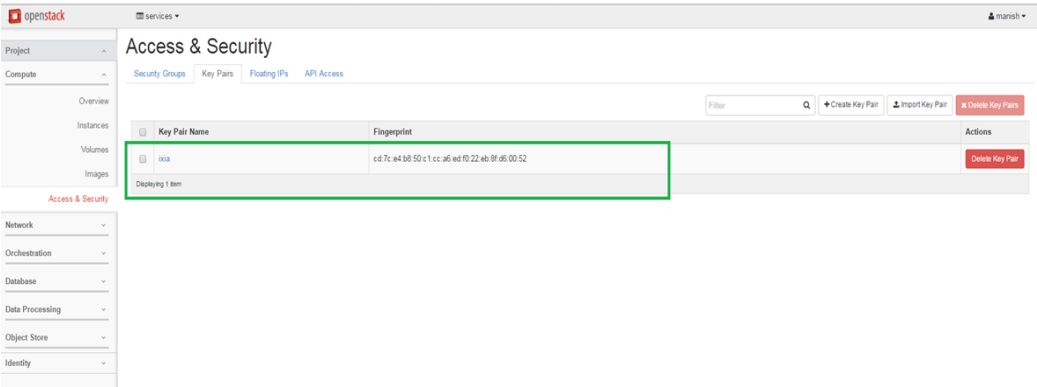
Note: All Egress traffic and intercommunication in the default group are allowed and all ingress from outside of the default group is dropped by default. To avoid dropped traffic, add the appropriate rules.



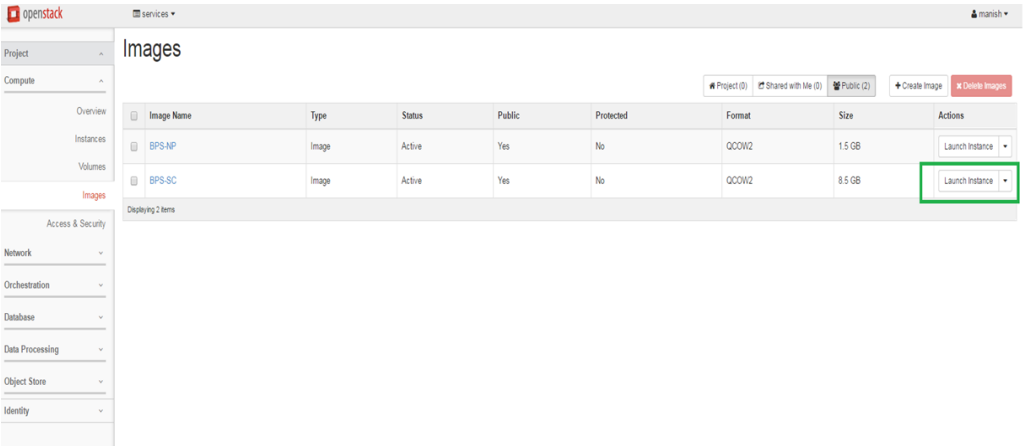

Add Rule
Rule *
ALL ICMP
Direction
Ingress
Remote *
CIDR
CIDR
0.0.0.0/0
Description:
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.
Cancel
Add

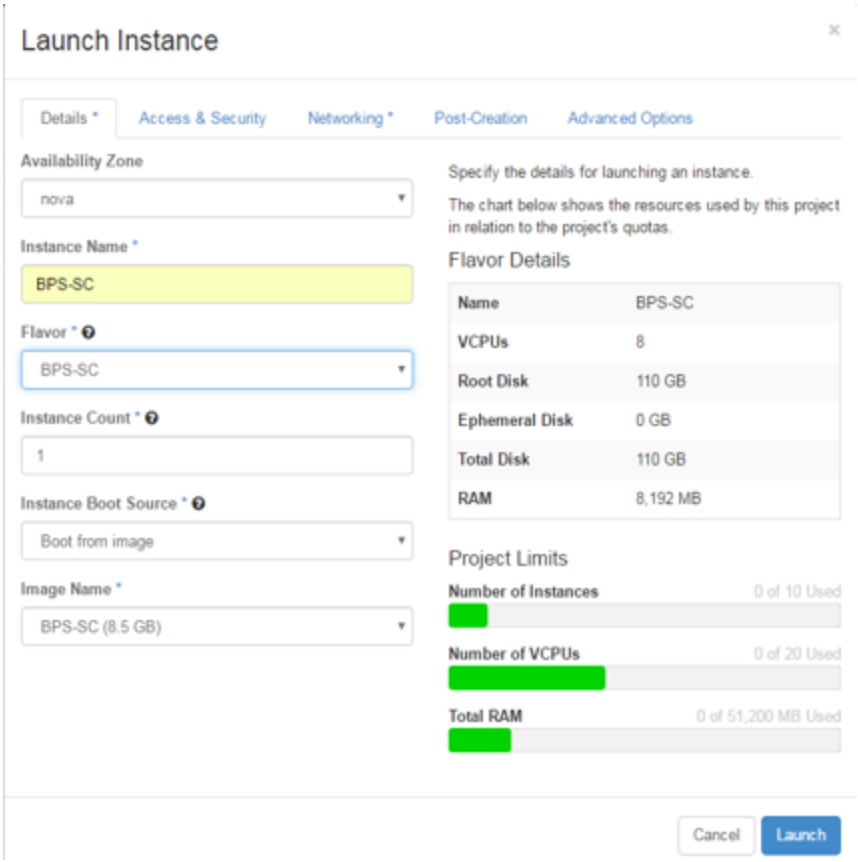

openstack
services
manish
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Database
Data Processing
Object Store
Identity
Manage Security Group Rules: default (ba7132d5-109d-47d8-8e3d-f391c4ae0c4a)
+ Add Rule
Delete Rules
Direction | Ether Type | IP Protocol | Port Range | Remote IP Prefix | Remote Security Group | Actions
Egress | IPv6 | Any | Any | ::/0 | - | Delete Rule
Ingress | IPv4 | Any | Any | - | default | Delete Rule
Egress | IPv4 | Any | Any | 0.0.0.0/0 | - | Delete Rule
Ingress | IPv6 | Any | Any | - | default | Delete Rule
Egress | IPv4 | ICMP | Any | 0.0.0.0/0 | - | Delete Rule
Ingress | IPv4 | ICMP | Any | 0.0.0.0/0 | - | Delete Rule
Ingress | IPv4 | TCP | 1 - 65535 | 0.0.0.0/0 | - | Delete Rule
Egress | IPv4 | TCP | 1 - 65535 | 0.0.0.0/0 | - | Delete Rule
Ingress | IPv4 | UDP | 1 - 65535 | 0.0.0.0/0 | - | Delete Rule
Egress | IPv4 | UDP | 1 - 65535 | 0.0.0.0/0 | - | Delete Rule

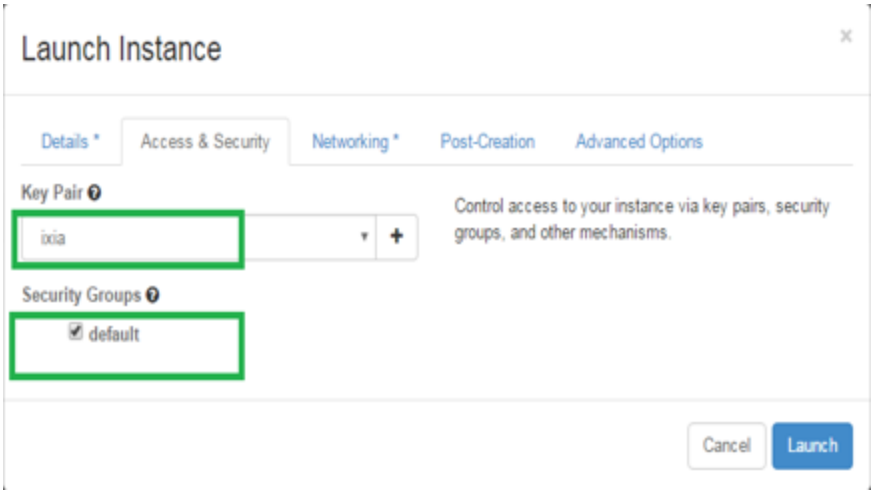


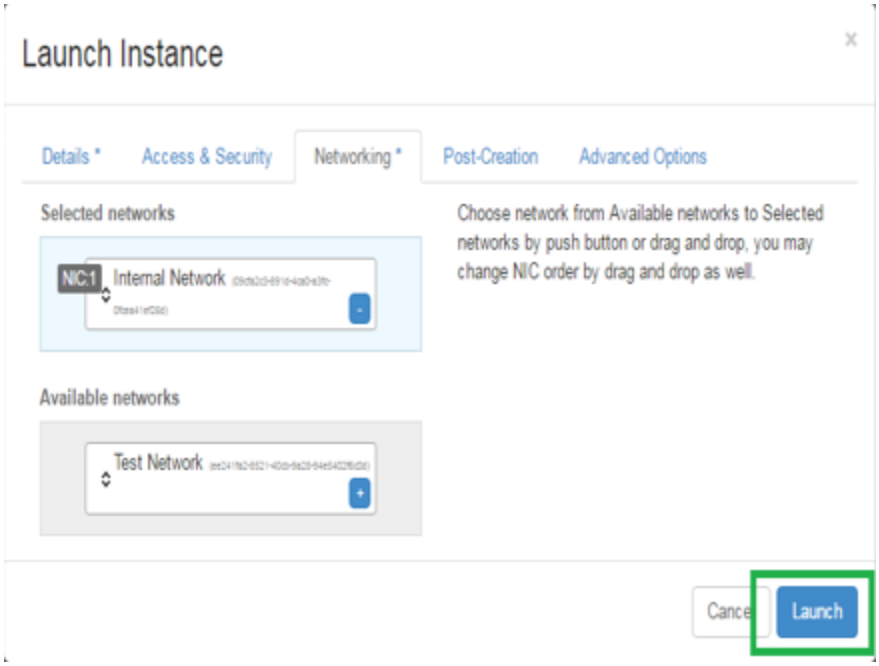


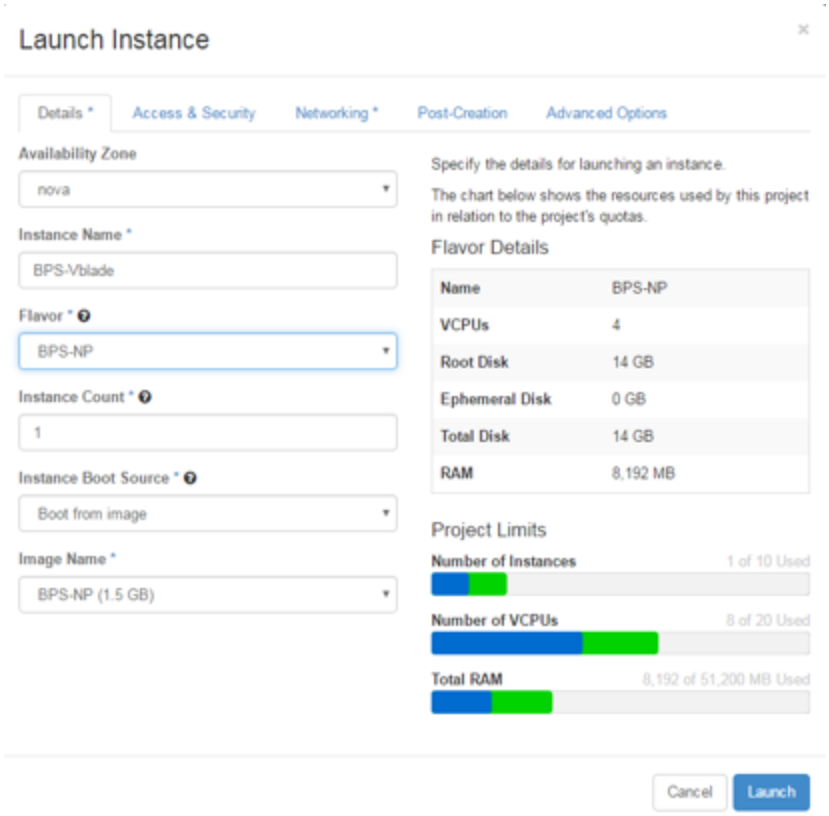
Launch Instances

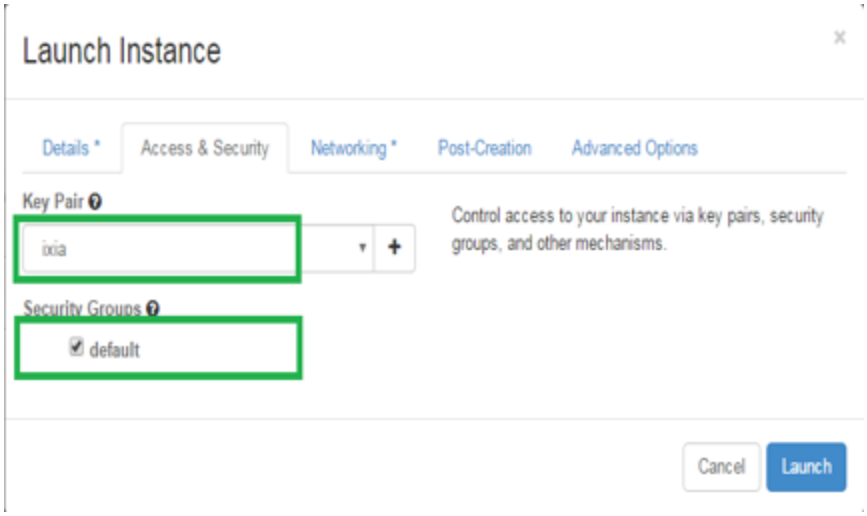


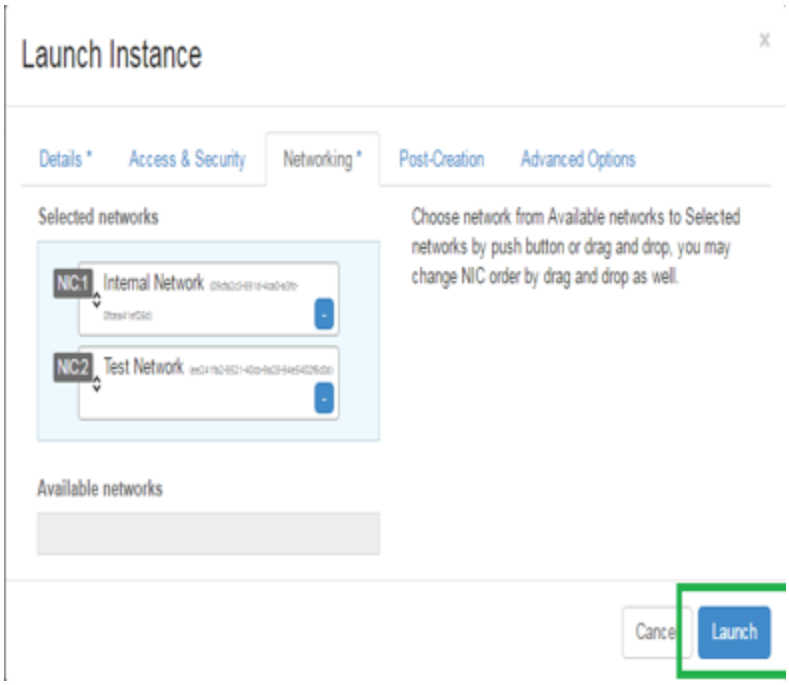

Launch Instance
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Availability Zone
nova
Instance Name *
BPS-SC
Flavor *
BPS-SC
Instance Count *
1
Instance Boot Source *
Boot from image
Image Name *
BPS-SC (8.5 GB)
Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.
Flavor Details
Name BPS-SC
VCPUs 8
Root Disk 110 GB
Ephemeral Disk 0 GB
Total Disk 110 GB
RAM 8,192 MB
Project Limits
Number of Instances 0 of 10 Used
Number of VCPUs 0 of 20 Used
Total RAM 0 of 51,200 MB Used
Cancel
Launch

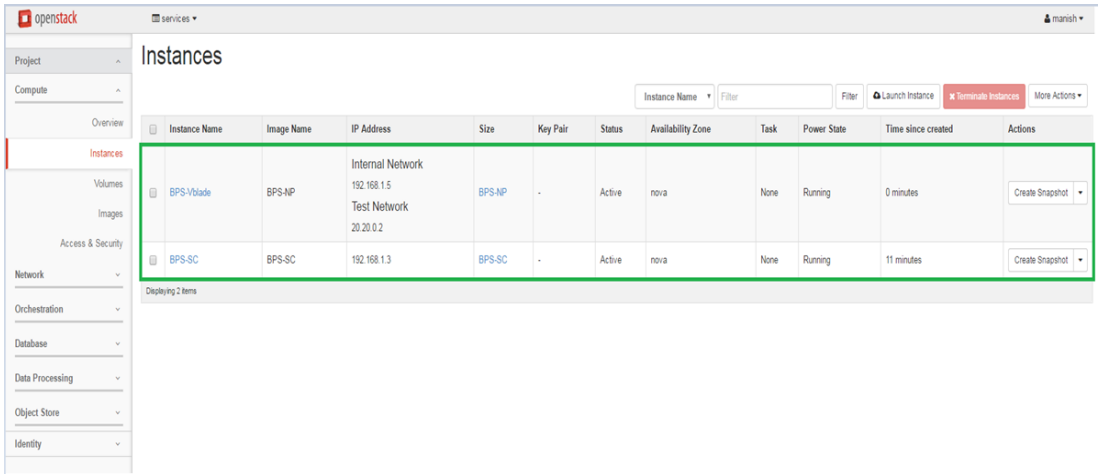

Launch Instance
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Key Pair
ixia
Control access to your instance via key pairs, security groups, and other mechanisms.
Security Groups
default
Cancel
Launch

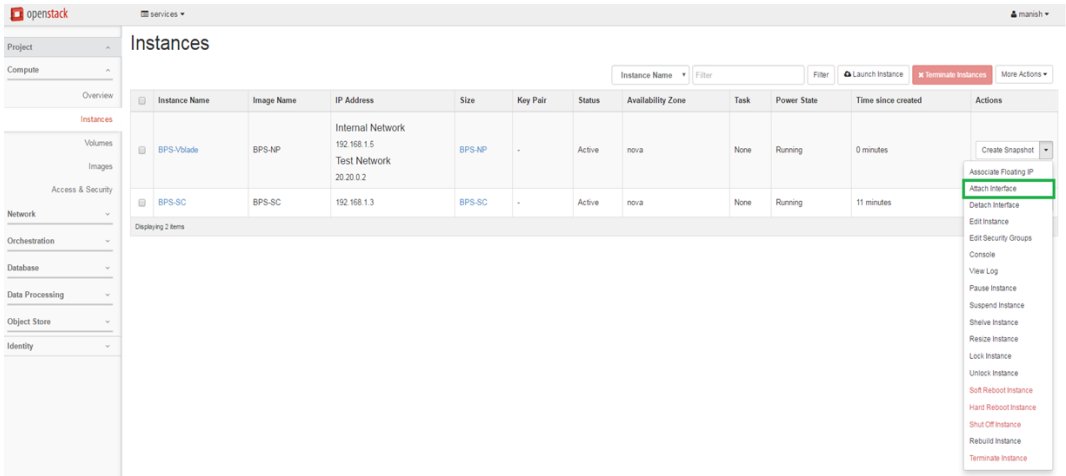

Launch Instance
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Selected networks
NIC 1
Internal Network
Choose network from Available networks to Selected networks by push button or drag and drop, you may change NIC order by drag and drop as well.
Available networks
Test Network
Cance
Launch


Launch Instance
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Availability Zone
nova
Instance Name *
BPS-Vblade
Flavor *
BPS-NP
Instance Count *
1
Instance Boot Source *
Boot from image
Image Name *
BPS-NP (1.5 GB)
Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.
Flavor Details
Name BPS-NP
VCPUs 4
Root Disk 14 GB
Ephemeral Disk 0 GB
Total Disk 14 GB
RAM 8,192 MB
Project Limits
Number of Instances 1 of 10 Used
Number of VCPUs 8 of 20 Used
Total RAM 8,192 of 51,200 MB Used
Cancel
Launch

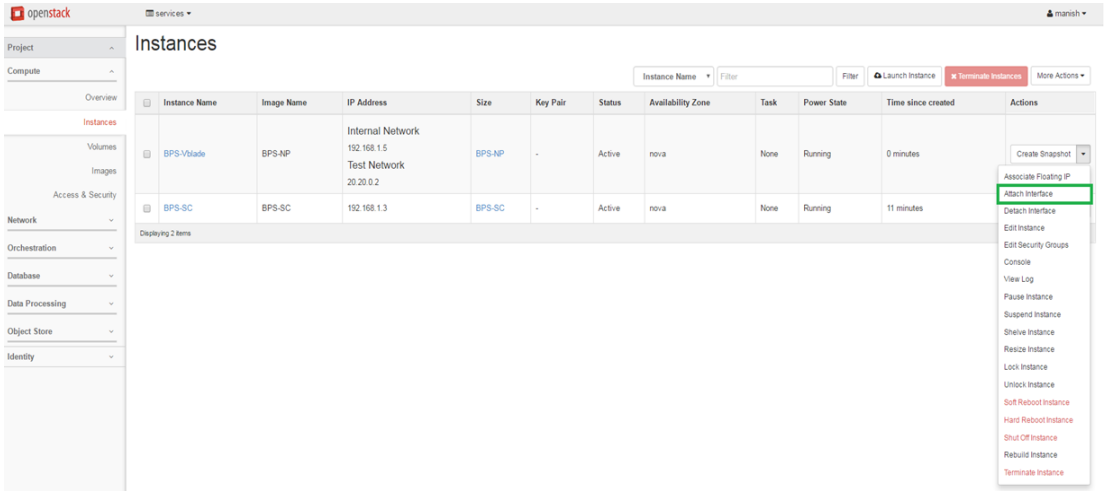

Launch Instance
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Key Pair
ixia
Control access to your instance via key pairs, security groups, and other mechanisms.
Security Groups
default
Cancel
Launch

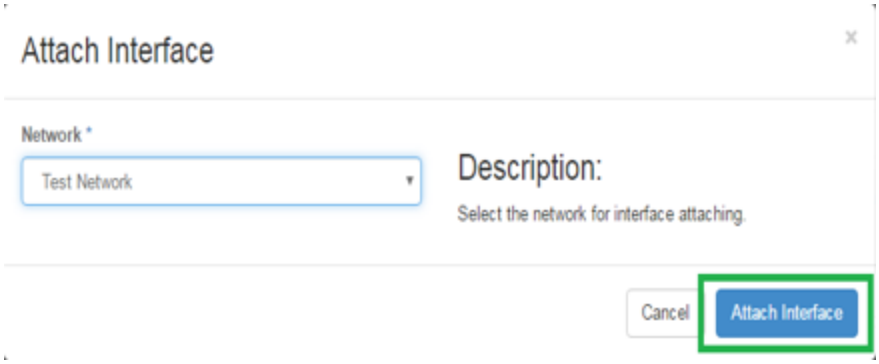

Launch Instance
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Selected networks
Choose network from Available networks to Selected networks by push button or drag and drop, you may change NIC order by drag and drop as well.
NIC1
Internal Network
NIC2
Test Network
Available networks
Cance
Launch

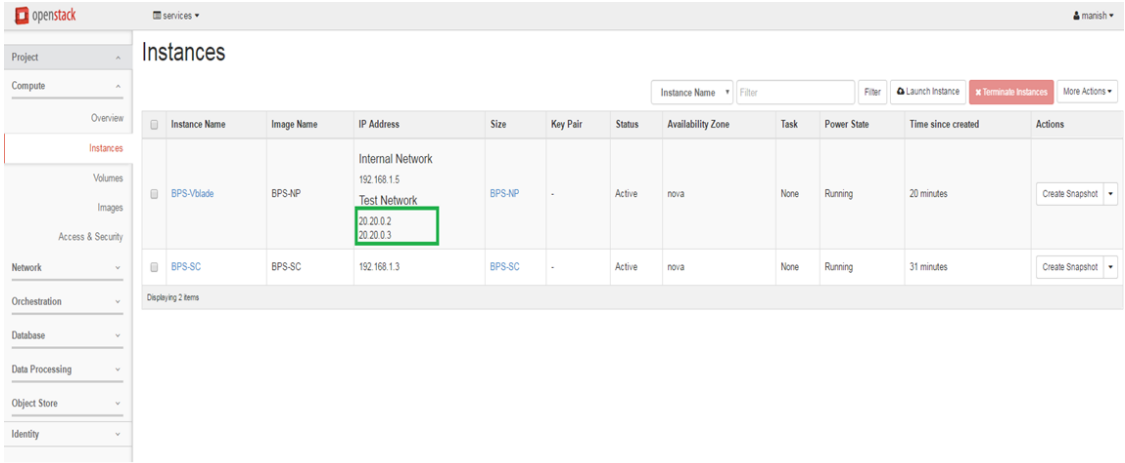




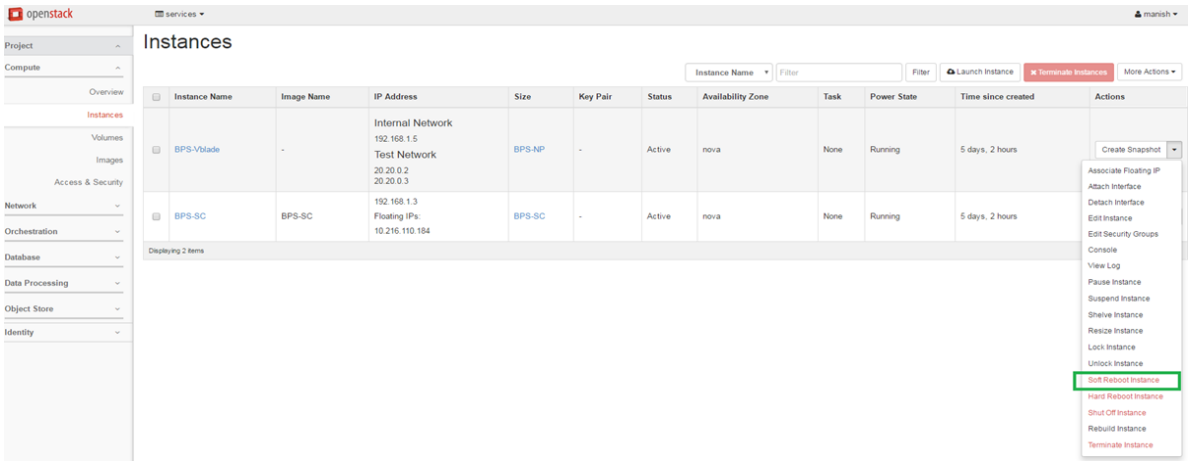
Define Multiple Test NICs





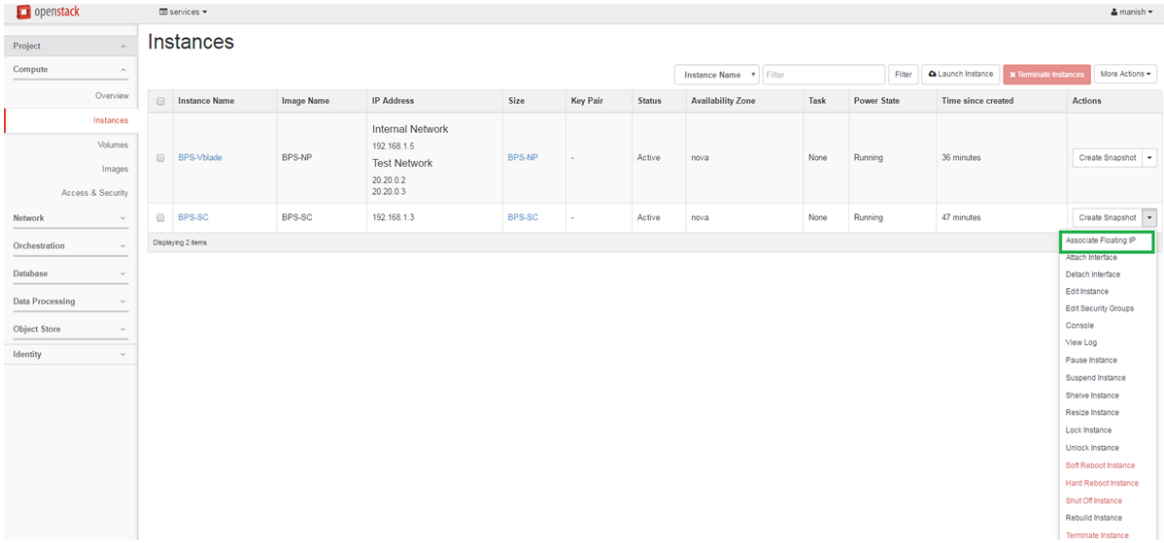


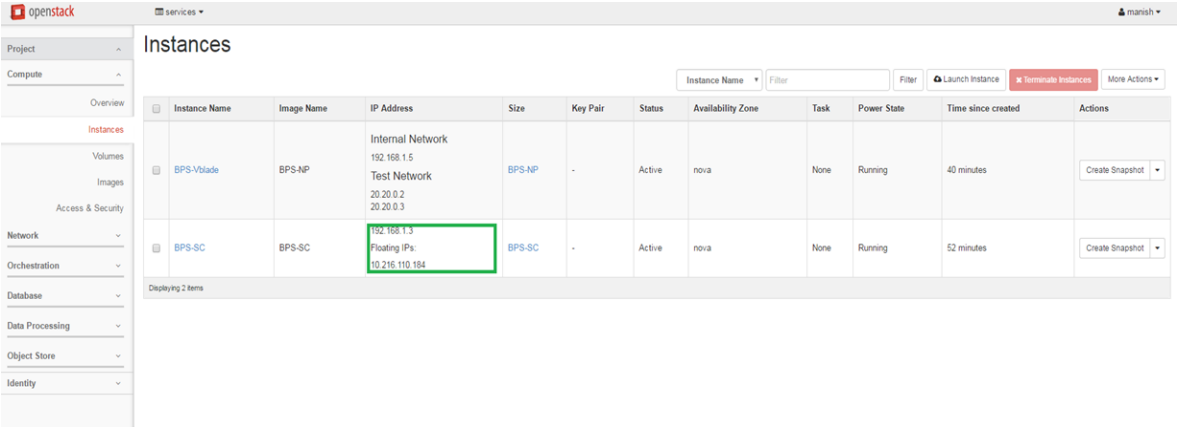
Note: After attaching the interface, the instance needs to be rebooted/service restarted in order for the change to be reflected in the BPS VE user interface. This step will complete this procedure.



Associate Floating IP Address

Note: Associating a floating IP address allows the BPS vController to be accessed from a LAN.





Configure the OpenStack Environment

This sections describes several options that can be used to configure your OpenStack environment for BPS VE.

Allow All MAC and IPs through OpenStack

By default, OpenStack allows only one MAC and one IP address through the test networks. The workaround to remove this limitation is to disable port-security on the test ports.

Perform the following tasks to allow all MACs and IPs through OpenStack:

1. Add the following line in `/etc/neutron/plugins/ml2/ml2_conf.ini` file to enable the ml2

port_security extension driver:

```
extension_drivers = port_security
```

2. Run the following command to restart the neutron services:

```
service restart neutron-server

service restart neutron-dhcp-agent

service restart neutron-l3-agent

service restart neutron-metadata-agent

service restart neutron-plugin-openvswitch-agent
```

3. Run the following command to list the neutron ports:

```
neutron port-list
```

4. Search for the test ports used on the VLMs and run the following commands on them:

```
neutron port-update <port-id> --no-security-groups
```

```
neutron port-update <port-id> --port-security-enabled=False
```

Note: In order to update a batch of ports with the above port security commands, you can use the following script:

a. Create an update_port_security.sh file with the following contents:

```
vi update_port_security.sh
#! /bin/bash
if [ $# -gt 1 ]; then
echo "Incorrect usage!"
echo -e "./update_port_security.sh [port_IP_format]\n"
echo -e "ex.:\n./update_port_security.sh 192.168."
exit 1
elif [ $# -eq 1 ]; then
PORT_IP=$1
echo -e "Searching for ports starting with IP: $PORT_IP"
else
PORT_IP="192.168."
echo -e "No IP selected!\nSearching for ports with default IP: $PORT_ IP"
fi
echo ""
echo "Grabbing the ports list..."
PORTS=$(neutron port-list | grep $PORT_IP | awk '{print $2}')
NUM_PORTS=$(neutron port-list | grep $PORT_IP | awk '{print $2}' | wc -l)
echo "Done!"
if [ -z "$PORTS" ]; then
echo "No ports found starting with IP $PORT_IP!"
exit 1
else
echo "Found $NUM_PORTS ports starting with IP $PORT_IP!"
fi
echo ""
```

```
ERRORS=0
ERROR_PORTS=""
echo -e "Disabling port security on the ports...\n"
for PORT in $PORTS;
do
neutron port-update $PORT --no-security-groups
FST=$?
neutron port-update $PORT --port-security-enabled=False
SND=$?
if [ $FST -eq 0 ] && [ $SND -eq 0 ]; then
echo "Successfully disabled port security on port $PORT!"
else
echo "Error on disabling port security for port $PORT!"
ERRORS=1
ERROR_PORTS=$ERROR_PORTS" "
fi

echo ""
done
if [ $ERRORS -eq 0 ]; then
echo "Finished updating all the ports!"
exit 0
else
echo "Found errors on updating the following ports: $ERROR_PORTS"
exit 1
fi
```

b. Run the following command to give it exec permissions.

```
chmod +x update_port_security.sh
```

The script applies the command only on a specific subset of ports, identified by an IP format (for example, 192.168.X.X). The test networks intended for creating for IxVM OpenStack use will have associated a subnet. You can easily identify the ports on which you must apply the configurations, based on the IPs associated by the test network in use. For example, setting subnet 192.168.10.0/24 on a test network results in test ports having allocated IPs from that range—192.168.10.2, 192.168.10.3, and so on).

c. Run the script.

```
./update_port_security.sh
```

By default, the script searches for ports starting with 192.168. as the IP. You can change this IP by providing an additional parameter when running the script. For example, `./update_port_security.sh 172.16.`, updates the ports having IPs with the 172.16.X.X format.

```
./update_port_security.sh 172.16.
```

This page intentionally left blank.

CHAPTER 2 BPS VE Install on Hyper-V

This section of the guide describes how to install BPS VE on Microsoft's Hyper-V.

Hyper-V Setup and Installation

Hyper-V is a virtualization technology tool from Microsoft allows you to create one or multiple virtual machines on Windows.

Note: Ixia has tested and supports Hyper-V on Windows Server 2019 Standard.

Note: BPS VE only supports the Hyper-V default drivers, "hv_netvsc".

Notes on unsupported features

The following are not supported:

- DPDK
- SR-IOV
- PCI-Passthrough

To install BPS VE on Hyper-V, you will need to log on to the Ixia support website (https://support.ixiacom.com/software-downloads/35471) and download the following Virtual Hard Disk (VHD) files:

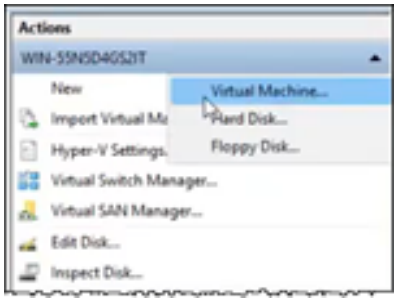
- Ixia_BreakingPoint_Virtual_Controller_9.xx.x_hyperV.vhd
- Ixia_BreakingPoint_Virtual_Blade_9.xx.x.KVM_.vhd

Perform the following steps to install BPS VE on Hyper-V:

Note: Please leave any setting that is not specifically mentioned in the procedure at the default.

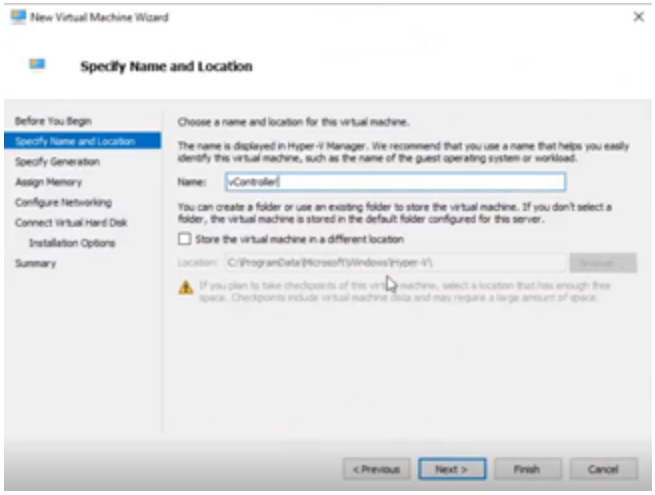
BPS VE Controller installation

1. In the **Actions** panel of your Hyper-V Manager, select **New** > **Virtual Machine**.



The **Before You Begin** dialog is displayed. Select **Next**.

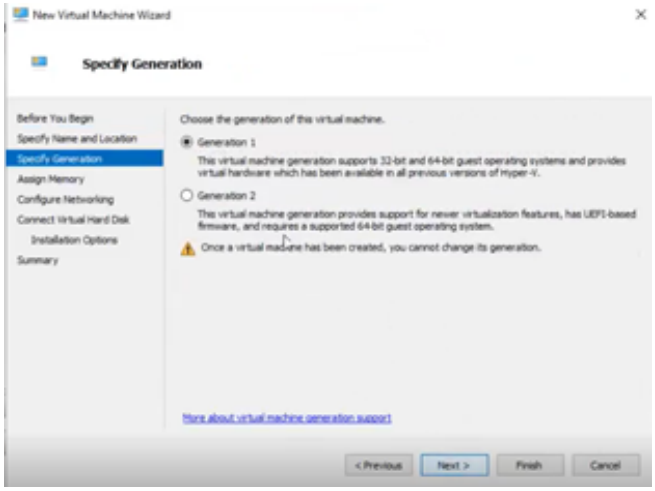
2. The **Specify Name and Location** dialog is displayed.



In the **Name** field, enter a name for the vController virtual machine. For our example, we used the name, "vController".

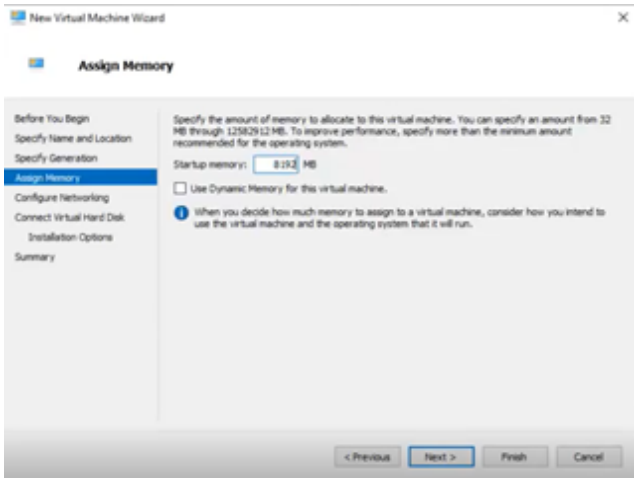
Select **Next**.

3. The **Specify Generation** dialog is displayed.



Select **Generation 1** and then select **Next**.

4. The **Assign Memory** dialog is displayed.

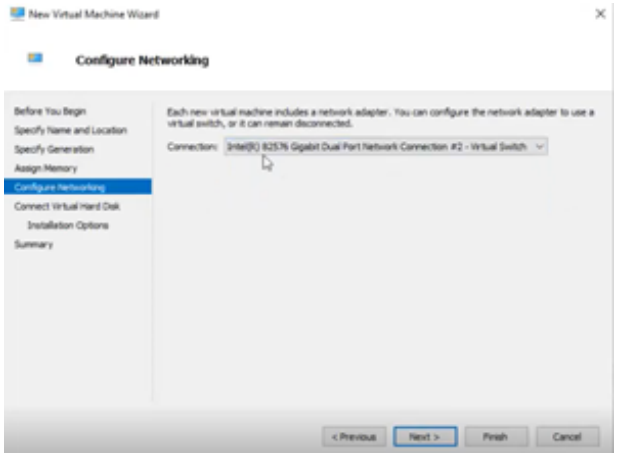


Enter the minimum value of `8192` **MB** (or greater) in the **Startup Memory __ MB** field.

> **Note:** Do NOT select Dynamic Memory.

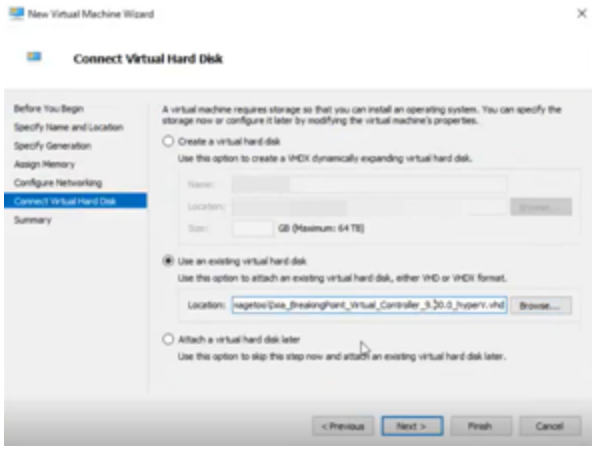
Select **Next**.

5. The **Configure Networking** dialog will display. This dialog allows you to select the network adapter that will be used as the virtual network for BPS VE management.



Select a network adapter from the **Connection** drop-down. Then select **Next**.

6. The **Connect Virtual Hard Disk** dialog will display. This dialog is used to select the BPS VE Controller virtual hard disk file that you downloaded from the Ixia Support website. Select Use an existing virtual hard disk.



Select **Use an existing virtual hard disk**.

In the **Location** field, enter the path, or select **Browse**, to indicate the path to the Ixia_ BreakingPoint_Virtual_Controller_9.xx.x_hyperV.vhd file.

Select **Next**.

7. A Summary of the virtual machine that will be installed is displayed. Select **Finish**.

   The new virtual machine (named "vController" in our example) will now appear in the **Virtual Machines** panel of your Hyper-V Manager user interface.

   a. Open the options menu of the virtual machine that you created and select **Settings**.

   b. Select **Processor** in the displayed **Hardware** panel.

   c. Set the **Number of virtual processors** option to "4".

   d. Select **OK**.

8. Open the options menu of the virtual machine that you created and select **Start**, to start the BPS VE Controller VM.

   Note: After the VM is up and running, you can connect to it to see the assigned controller IP address. You can enter the controller IP address into your HTML browser URL field to access the BPS VE management UI.

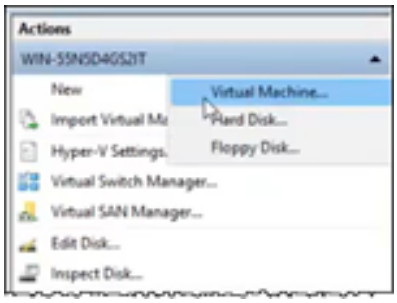
BPS VE vBlade installation

Note: vBlade installation is almost identical to vController installation with the following notes and exceptions.

- On Step 2 - Specify Name and Location, we recommend entering a name that indicates that a vBlade is being deployed.
- On step 5 - Configure Networking, you must select the same network adapter that was used for the BPS VE Controller virtual network. This effectively allows the BPS VE Controller and vBlade to communicate on the same network.
- On step 6 - Connect Virtual Hard Disk, you need to select the **vBlade** .vhd file (Ixia_ BreakingPoint_**Virtual_Blade**_9.xx.x.KVM_.vhd) that you downloaded from the Ixia support

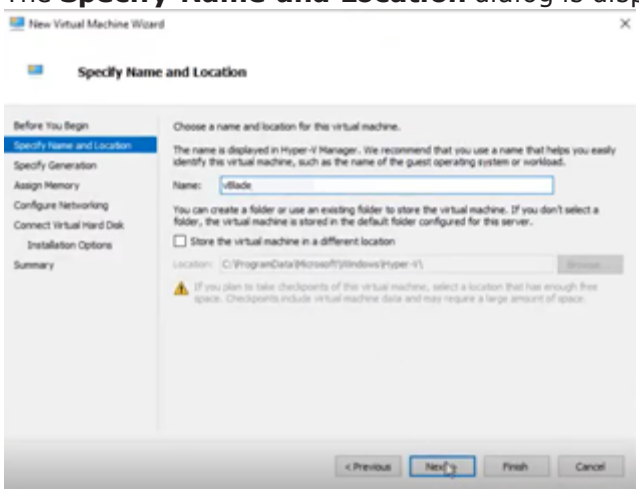
site instead of the vController .vhd file.

1. In the **Actions** panel of your Hyper-V Manager, select **New** > **Virtual Machine**.



The **Before You Begin** dialog is displayed. Select **Next**.

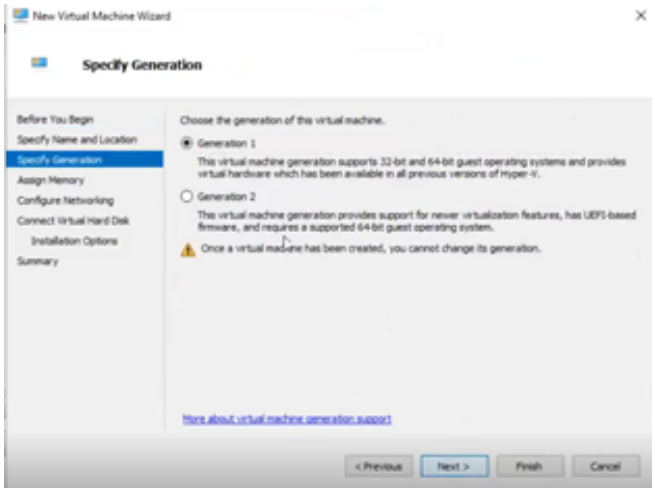
2. The **Specify Name and Location** dialog is displayed.



In the **Name** field, enter a name for the vBlade virtual machine. For our example, we used the name, "vBlade".

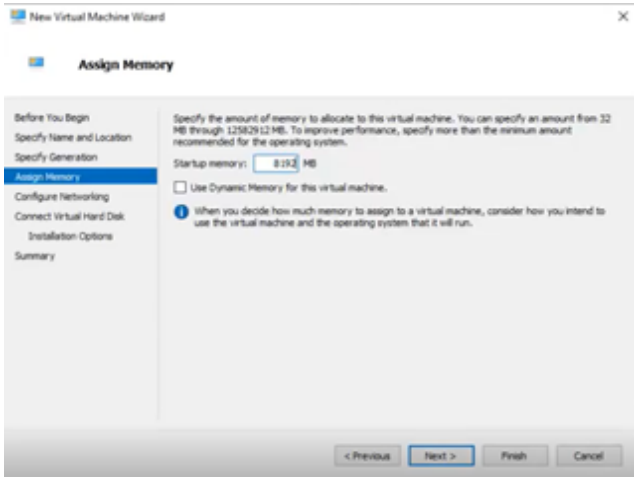
Select **Next**.

3. The **Specify Generation** dialog is displayed.



Select **Generation 1** and then select **Next**.

4. The **Assign Memory** dialog is displayed.

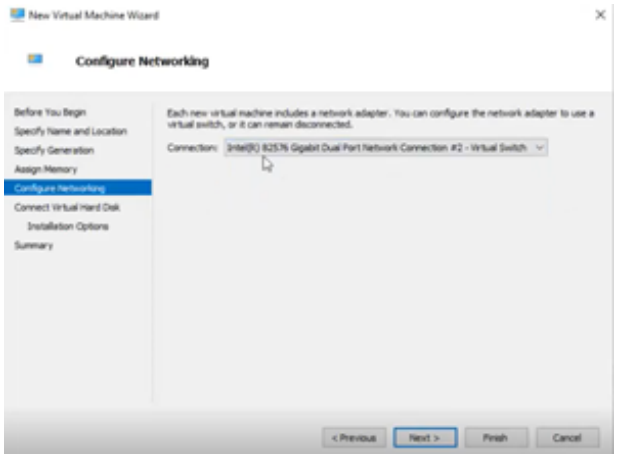


Enter the minimum value of `8192` **MB** (or greater) in the **Startup Memory __ MB** field.

> **Note:** Do NOT select Dynamic Memory.

Select **Next**.

5. The **Configure Networking** dialog will display.

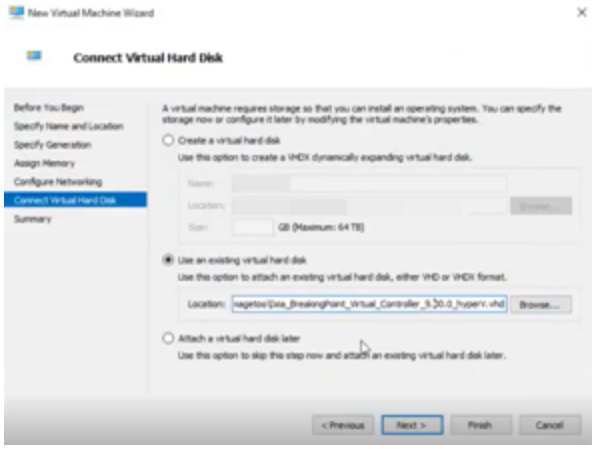


Use the **Connection** drop-down to select the same network connection that was used for the BPS VE controller virtual network.

> **Note:** There is a Hyper-V limitation that only supports one network adapter when the VM is deployed. If you would like to add additional network adapters to the VM, they must be added after deployment.

Then select **Next**.

6. The **Connect Virtual Hard Disk** dialog will display. This dialog is used to select the BPS VE Controller virtual hard disk file that you downloaded from the Ixia Support website. Select Use an existing virtual hard disk.



Select **Use an existing virtual hard disk**.

In the **Location** field, enter the path, or select **Browse**, to indicate the path to the Ixia_ BreakingPoint_Virtual_Controller_9.xx.x_hyperV.vhd file.

Select **Next**.

7. A Summary of the virtual machine that will be installed is displayed. Select **Finish**.

   The new virtual machine (named “vBlade” in our example) will now appear in the **VirtualMachines** panel of your Hyper-V Manager user interface.

   a. Open the options menu of the virtual machine that you created and select **Settings**.
   b. Select **Processor** in the displayed **Hardware** panel.
   c. Set the **Number of virtual processors** option to "4".
   d. Select **OK**.

8. Open the options menu of the virtual machine that you created and select **Start**, to start the BPS VE Controller VM.

   Note: After the VM is up and running, you can connect to it to see the assigned controller IP address. You can enter the controller IP address into your HTML browser URL field to access the BPS VE management UI.

CHAPTER 3 BPS VE Install on Alibaba Cloud

Ixia provides the files and resources required to deploy BPS VE on Alibaba Cloud.

Alibaba Cloud Setup and Installation

Note: BPS VE is supported on Alibaba Cloud (compatible but NOT certified).

Alibaba Prerequisites

The following will be required on Alibaba for successful BPS VE deployment:

- A Virtual Private Cloud (VPC)
  - Please note the VPC ID because it will be needed for the deployment.
- A Virtual Switch
  - Please note the Virtual Switch ID because it will be needed for the deployment.
- A Security Group
  - Ixia recommends that all inbound and outbound ports should be opened. Specific information about the Open Port Requirements can be found in Appendix B.
  - Please note the Security Group ID because it will be needed for the deployment.

Required BPS VE Files

Log on to the Ixia support website (https://support.ixiacom.com/) and download the following BPS VE on Alibaba deployment files.

- Ixia_BreakingPoint_Virtual_Controller_9.xx_Alibaba_ROS_Template.yaml
- Ixia_BreakingPoint_Virtual_Blade_9.xx_Alibaba_ROS_Template.yaml

Perform the following steps to install BPS VE on Alibaba:

Note: Please leave any settings that are not specifically defined at the default.

1. Import the BPS VE image files on to Alibaba.
   a. Install OSS Browser (https://www.alibabacloud.com/help/doc-detail/61872.htm?spm=a2c63.p38356.879954.7.16d21cb4FDjEi8#concept-xmg-h33-wdb)
   b. Upload an image using the OSS Browser.
   c. After the upload has completed, you will be able to see these images in the Elastic Compute Service. Note the image IDs because they will be needed during the installation.

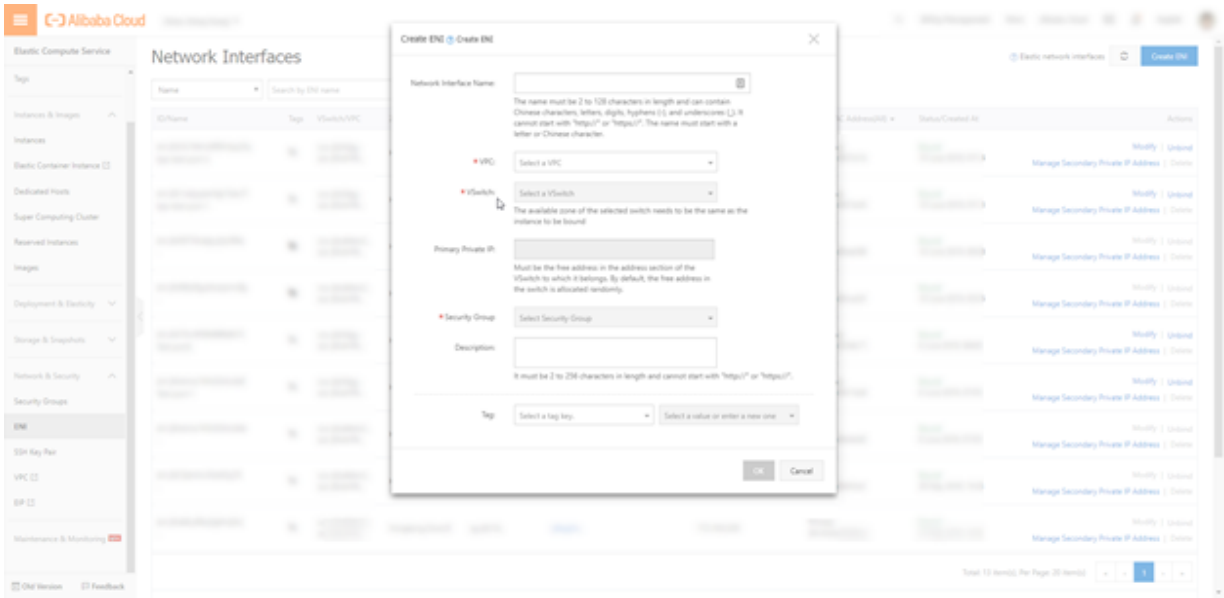
2. In the Alibaba Cloud user interface, select **Resource Orchestration Service**.
3. Select **Stack Management**.
   a. Use the drop-down selector to pick the region where you want to install the BPS VE virtual machine.
   b. Select the **New Resource Stack** button.
   c. Configure the **Template source** field as: **Enter directly**.
   d. Use a text editor to open the Ixia_BreakingPoint_Virtual_Controller_9.xx_Alibaba_ROS_Template.yaml file. Copy and paste all of the text in this file into the **Template data** field.
   e. Click **Next**.
   f. In the **Stack Name** field, type a name or label for the stack.
   g. **In the Parameters section**:
      i. Enter the BPS VE Controller **imageID** value. This value is displayed in the **ECS Console** under the **Images** column. This text can be copied and pasted into the **imageID** field.
      ii. Enter the ID of the VPC that will be used with this VM into the **vpc** field.
      iii. Enter the ID of the virtual switch that will be used with this VM into the **vswitch** field.
      iv. Enter the name of the security group that will be used with this VM into the **SecurityGroupName** field.

      Note: Ixia recommends that all inbound and outbound ports should be opened. Specific information about the Open Port Requirements can be found in Appendix B.

   h. Select **Create**.
4. Repeat steps 2 and 3 to deploy the BPS VE vBlade. On step 3d, you will copy and paste text from the Ixia_BreakingPoint_**Virtual_Blade**_9.xx_Alibaba_ROS_Template.yaml file.

Note that Alibaba Cloud will not allow you to create a VM with more than one network interface. If you want to add additional network interfaces for testing, you will need to use the Alibaba Cloud GUI to attach additional interfaces to the VM.

To attach a new ENI to the instance you must first create it from **Elasti Compute Service** -> **Network and Security** -> **ENI**.


Alibaba Cloud
Elastic Compute Service
Network Interfaces
Create ENI
Network Interface Name
VPC
VSwitch
Primary Private IP
Security Group
Description
Tag
OK
Cancel

This page intentionally left blank.

CHAPTER 4 BPS VE Install on Amazon Web Services

This section of the guide describes how to install BPS VE on Amazon Web Services.

BPS on AWS Overview

This section of the document provides a straightforward workflow that will assist you while deploying the Breaking Point AMIs in Amazon Web Services (AWS). It will also help you create a sample setup for your device under test.

This document assumes you are familiar with the basics of the Amazon AWS Virtual Private Cloud (VPC) and Elastic Compute Cloud (EC2) features. If not, we encourage you to study the tutorials provided by Amazon at https://aws.amazon.com/training/intro_series/.

BPS VE AMI Deployment

This section of the document discusses the following methods for BreakingPoint AMI Deployment on Amazon Web Services.

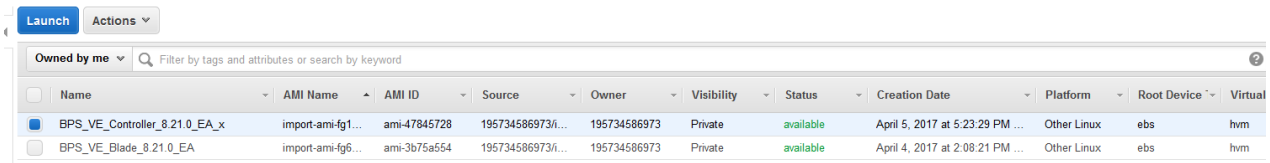
- AMI Deployment below
- CloudFormation Template Generator on page 71

AMI Deployment

Note: You can find the AMIs for the Ixia BreakingPoint System Controller and Ixia BreakingPoint vBlade on the EC2 console (**Instances** > **Launch Instance** > **Community AMIs**) using the AMI IDs or by searching for Ixia BreakingPoint.

To deploy BPS VE on Amazon EC2, you need to perform the following steps:

1. Select **EC2 Dashboard** > **Images** > **AMIs**.
2. Select the **BPS AMIs** and select **Launch** and then follow the steps in the wizard.

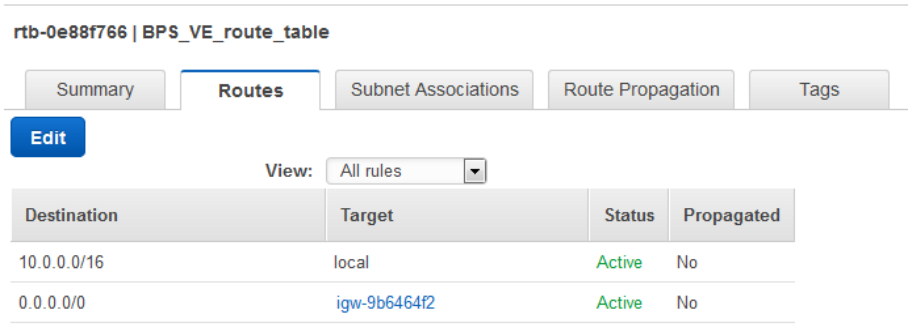


3. Choose an instance type based on your computing needs:
   - vController Minimum requirements: 8vCPUs, 8 GB RAM, 100 GB HDD
   - vBlade Minimum requirements 4vCPUs, 8 GB RAM, 10 GB HDD
4. On the **Configuration Instance Details** page, select:
   a. Create a new VPC (you can also select an existing VPC)
      i. Create the VPC and assign a subnet block, e.g.: IPv4 CIDR block = 10.0.0.0 /16
      ii. Configure the VPC subnets (at least two subnets are required at this stage, one for External Management and one for Internal Management), for example:
         - 10.0.0.0 /24 ; ixia-management - used to access the vController WebUI (BPS GUI)
         - 10.0.1.0 /24 ; ixia-control - used for the internal communication between vController and vBlade

ggircu_ixia_control	subnet-5104912b	available	vpc-7ab53812 | ggircu_BPS_VE_...	10.0.1.0/24
ggircu_ixia_management	subnet-9d0792e7	available	vpc-7ab53812 | ggircu_BPS_VE_...	10.0.0.0/24

Note: Optionally, you can use the same subnet for External Management and Internal Management. In this scenario, please remember to add both of the network interfaces (attached to the vController instance) as well as the primary network interface (eth0 - attached to the vBlade instance) to the same management subnet.

      i. Create the route table (the table controls the routing for the subnet)
         i. Go to **Route Tables** and select **Create Route Table**
         ii. To ensure that your instances can communicate with the Internet, you must also attach an Internet gateway to your VPC
         iii. Go to **Internet Gateways** and select **Create Internet Gateway**
         iv. Open the **Create Internet Gateway** context menu and select **Attach to your VPC**
         v. Go back to the route table configuration > **Select Routes** > **Add another route**
         vi. Add a route over the Internet gateway (the destination is 0.0.0.0/0, and the target is the Internet gateway you just created)



ii. Go to **VPC** > **Subnets**, then select your subnets and change the **Current Route Table** to the route table you just created

b. For **Subnet**, select:

i. ixia-management, when deploying the vController instance

ii. ixia-control, when deploying the vBlade instances

c. **Auto-assign Public IP**:

- Use subnet settings

d. **Network interfaces**:

i. **vController** - When deploying the controller instance, make sure you add a **second network interface** (vController has two management interfaces):

- The 1st interface must be added to the **External Management** subnet: eth0
- The 2nd interface must be added to the **Internal Management** subnet: eth1

> **Note:** If you start an instance with more than one network interface, it will no longer use a regular public IP address. If you connect to instances in your VPC using public IPs, you will need to assign an **Elastic IP** to the BPS vController instance.

ii. **vBlade**

- Has only one management interface
- Needs to be in the same IP subnet with the vController Internal Management IP

> **Note:** You can add one vBlade test interface before launching the instance.After launching the instance, you can add more vBlades using the EC2 console.

5. Under **Add Storage**, the default storage size should be enough.
6. Under **Add Tags**, the recommendation is to add some tags to allow easily finding the instance, e.g., set the Key to Username and set the value to your login.
7. Configure the security group, e.g.:

a. **Inbound**

i. HTTPS must be allowed only from your personal or corporate network IP (range)

ii. HTTP must be allowed only from your personal or corporate network IP (range)

iii. SSH must be allowed only from your personal or corporate network IP (range)

iv. TCP traffic on port 8880 must be allowed only from your personal or corporate network IP (range)

v. ALL traffic must be allowed within the security group (if configuring different security groups for the vController and the vBlade, make sure that ALL traffic is allowed between the security groups)

sg-f390a798 | bpsVPCx

Summary | **Inbound Rules** | Outbound Rules | Tags

Edit

Type	Protocol	Port Range	Source
HTTP (80)	TCP (6)	80	109.100.41.154/32
HTTP (80)	TCP (6)	80	::/0
ALL Traffic	ALL	ALL	sg-f390a798
SSH (22)	TCP (6)	22	109.100.41.154/32
SSH (22)	TCP (6)	22	::/0
Custom TCP Rule	TCP (6)	8880	109.100.41.154/32
Custom TCP Rule	TCP (6)	8880	::/0
DNS (TCP) (53)	TCP (6)	53	109.100.41.154/32
DNS (TCP) (53)	TCP (6)	53	::/0
HTTPS (443)	TCP (6)	443	109.100.41.154/32
HTTPS (443)	TCP (6)	443	::/0

b. **Outbound**

i. Traffic must be allowed to any IP address

It is highly recommended not to allow arbitrary (inbound) access to your BPS VE instances – only IPs from your company or home should be allowed to access this machine. This will help to protect any confidential data stored on this instance/network.

8. Review the settings you've selected and then select **Launch**.
9. Select an existing key pair (or create a new one) and check the **I acknowledge** check box. Select **Launch Instances**.

> **Note:** In the current version, BPS VE instances cannot be accessed using the Amazon key-pair.

CloudFormation Template Generator

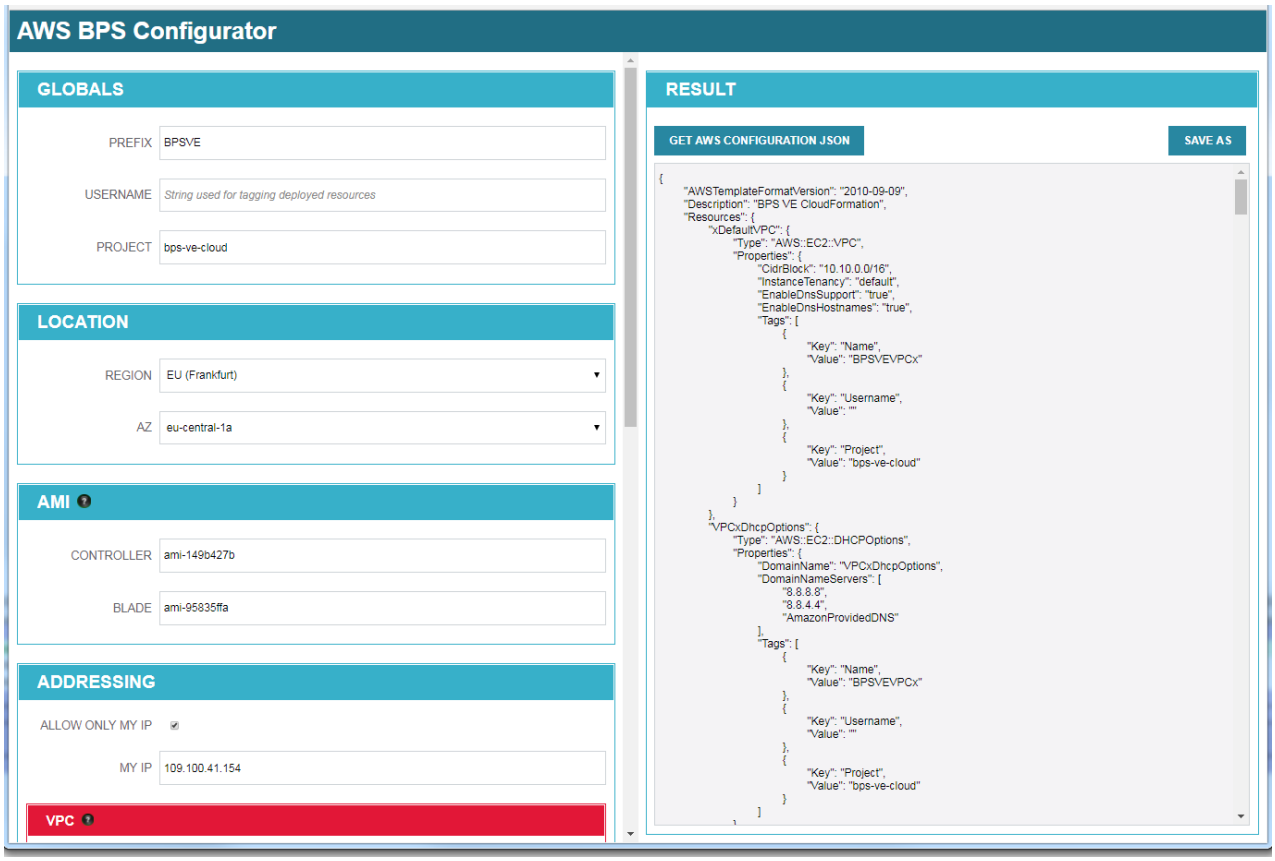
The deployment of Breaking Point AMIs can be automated by using CloudFormation templates. This option automates most of the manual steps that have been detailed in the AMI Manual Deployment section.

In order to generate a CloudFormation template, you can use the following helper page:

bps-deploy.s3-website.eu-central-1.amazonaws.com.

Note: The AWS BPS Configurator helper page described below is supported on the Mozilla Firefox and Chrome web browsers.

Note: When deploying a CloudFormation template generated by the AWS BPS Configurator helper page, the maximum number of IPs supported by the instance type will be automatically configured on the elastic network interfaces (ENIs) connected to the vBlade.



The helper page offers various configuration options including:

- AMI selection for BPS System Controller and vBlade
- AWS Deployment Region and Availability Zone
- VPC configuration
- Test and Management IP range configuration
- System Controller and vBlade instance types
- Number of vBlades
- Number of Test Ports per vBlade

CloudFormation templates are generated by selecting **Generate AWS Configuration JSON**. These templates can be used as-is or can serve as a starting point for further customization.

Note: When deploying a CloudFormation template in AWS, the vBlades are automatically connected to the BPS System Controller and will appear in the **Administration** > **VM Deployment** > **Manage Virtual Chassis** window.

Parameter			Description
Globals	Prefix		Insert the prefix. This string will be appended to the name of the resources that the AWS CloudFormation template generates.
	Username		Insert the username tag. AWS CloudFormation Resource Tags property is used to apply tags to resources, which can help you identify and categorize those resources.
	Project		Insert the project tag. AWS CloudFormation Resource Tags property is used to apply tags to resources, which can help you identify and categorize those resources.
Location	Region		Select a Region that specifies where your resources are managed.
	AZ		Select the Availability Zone. Availability zones are isolated locations within data center regions from which public cloud services originate and operate.
AMI	Controller		Insert the ID of the vController AMI. You can find the AMIs for the Ixia BreakingPoint System Controller and Ixia BreakingPoint vBlade on the EC2 console (**Instances** > **Launch Instance** > **Community AMIs**) using the AMI IDs or by searching for Ixia BreakingPoint.
	Blade		Insert the ID of the vBlade AMI. You can find the AMIs for the Ixia BreakingPoint System Controller and Ixia BreakingPoint vBlade on the EC2 console (**Instances** > **Launch Instance** > **Community AMI**s) using the AMI IDs or by searching for Ixia BreakingPoint.
Addressing	Allow only My IP		Use this setting in order to not allow arbitrary (inbound) access to your BPS instances. When enabled, only the specified IP will be allowed to access these machines. This helps protect any confidential data stored on these instances and the rest of the network.
	MY IP		The IP address to be used in the security rules. Your public IP address is automatically filled in.
	VPC	Name	Insert the name of the VPC. It can only contain alphanumeric characters.
		CIDR	Insert the IPv4 address range for your VPC as a Classless Inter-Domain Routing (CIDR) block. CIDR notation is a compact representation of an IP address and its associated routing prefix. The notation is constructed from an IP address, a slash ('/') character, and a decimal number.

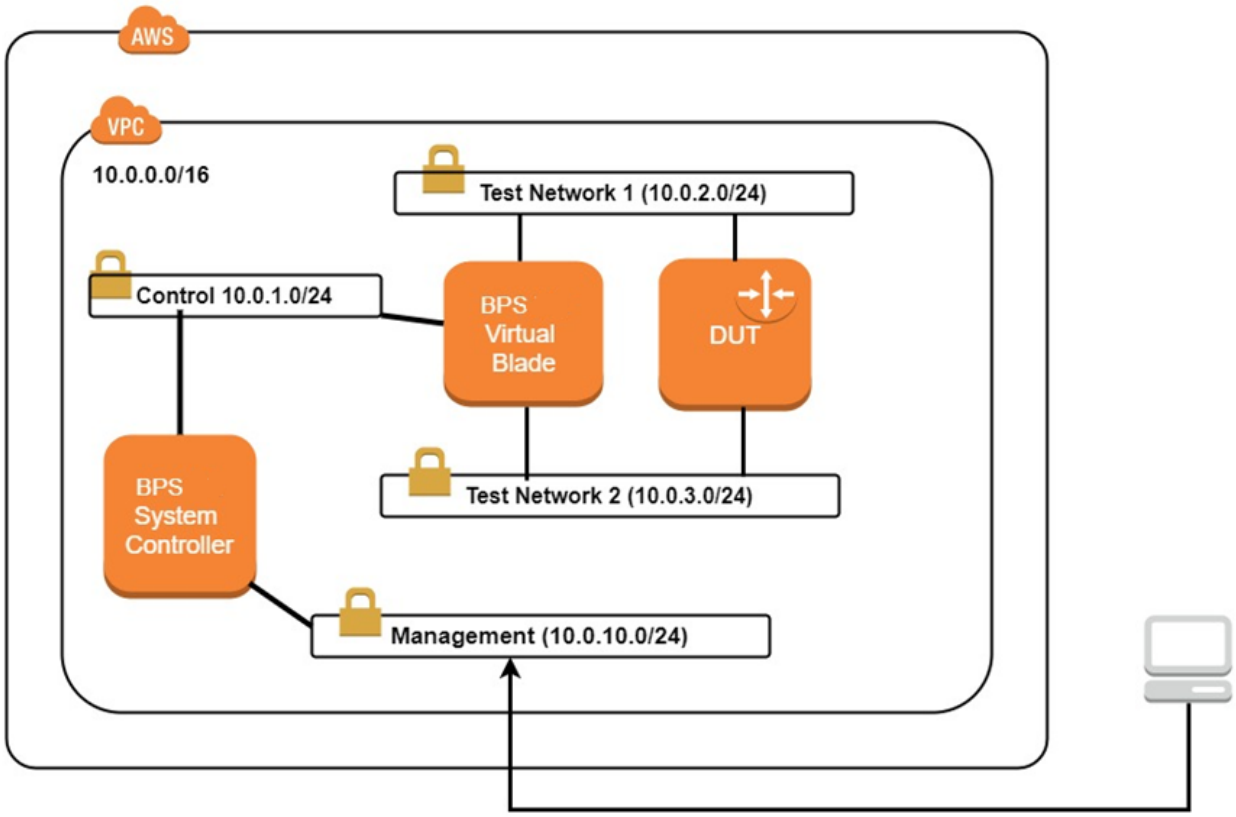
Parameter			Description
	Management Subnet	Name	Insert the name of the Management Subnet. It can contain only alphanumeric characters.
		CIDR	Insert the IPv4 address range for your Management Subnet, as a Classless Inter-Domain Routing (CIDR) block. CIDR notation is a compact representation of an IP address and its associated routing prefix. The notation is constructed from an IP address, a slash ('/') character, and a decimal number.
	Test Subnet	Name	Insert the name of the Test Subnet. It can contain only alphanumeric characters.
		CIDR	Insert the IPv4 address range for your Test Subnet, as a Classless Inter-Domain Routing (CIDR) block. CIDR notation is a compact representation of an IP address and its associated routing prefix. The notation is constructed from an IP address, a slash ('/') character, and a decimal number.
Instance Configuration	Controller	Instance Type	When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. Select an instance type for the BPS vController based on the requirements of the application or software that you plan to run on your instance.
	Blade	Index	The index of the blade.
		Instance Type	When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. Select an instance type for the BPS vBlade based on the requirements of the application or software that you plan to run on your instance.
		Port Count	Specify the number of ports per vBlade (from one to eight virtual test ports). *Please note that an extra-port will be added for management purposes. The maximum number of IP Addresses per Network Interface depends on the Instance Type. Make sure to consult http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html in order to check the limits for the maximum number of network interfaces, IPv4/IPv6 addresses per Interface per Instance Type.

Configuring Test Interfaces on AWS

BPS on Amazon Web Services requires additional test interfaces that will be used for sending test traffic into your network. These interfaces must be configured to connect to private subnets (not connected to the internet) with permissive security rules to allow many different (and unconventional) types of traffic to flow through your network. Each interface that you add should share a subnet with a single interface on your device. The minimum number of network interfaces that must be added is two.

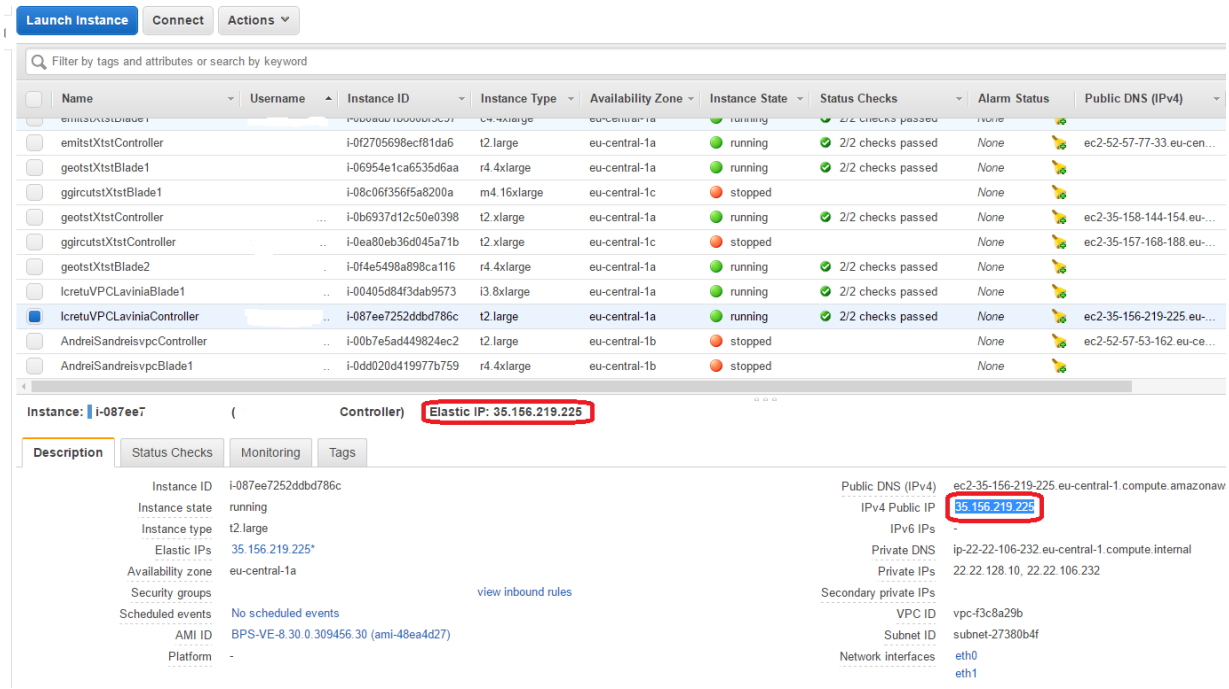
Please ensure that there is network connectivity between the outbound BPS VE vBlade Test Interfaces and the interfaces of the Device Under Test.

An example configuration is shown below.



Running a Test on AWS

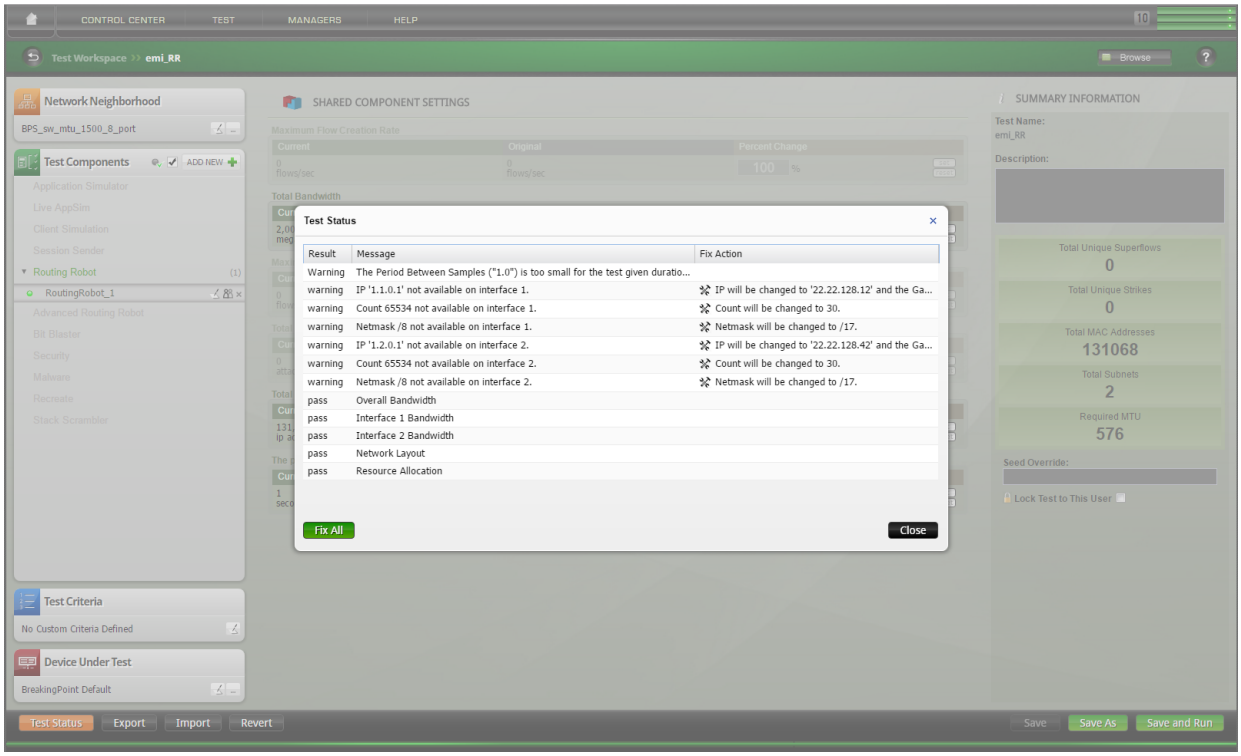
In order to run a test, enter the Elastic IP of the vController instance into the URL field of your HTML browser.



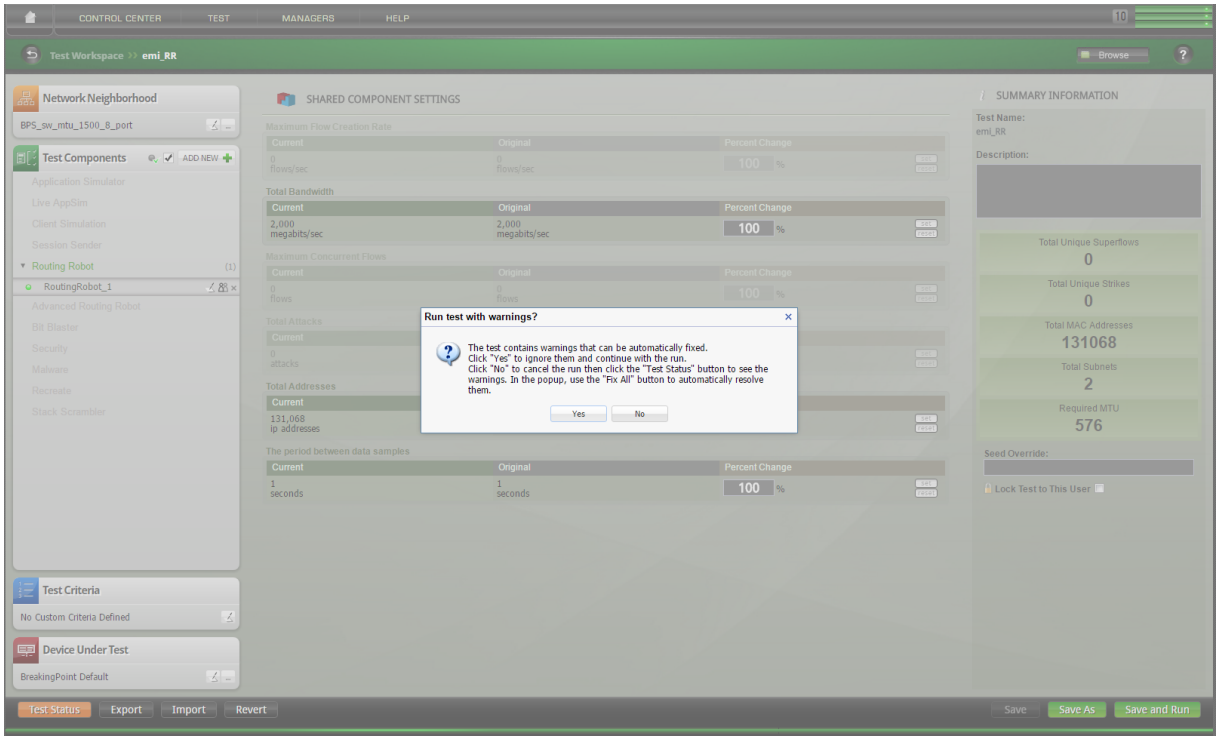
The BreakingPoint user interface will display. For detailed information regarding the user interface, please see the BreakingPoint User Guide.

When running in the AWS environment, the test IPs configured in the BreakingPoint Network Neighborhood should match the IPs assigned to the Test Interfaces on the vBlade instance for the corresponding test. This ensures proper network connectivity between BreakingPoint and any Device Under Test.

BreakingPoint will automatically detect any mismatch between the IPs configured in the Network Neighborhood and the IPs assigned to the test interfaces and indicate the status on the **Test Status** button. When the Test Status details window is opened, you will be given the option to automatically match the IP addresses by selecting the **Fix All** button.



If the option to match IP addresses is ignored, a warning message will display when you attempt to run the test



.

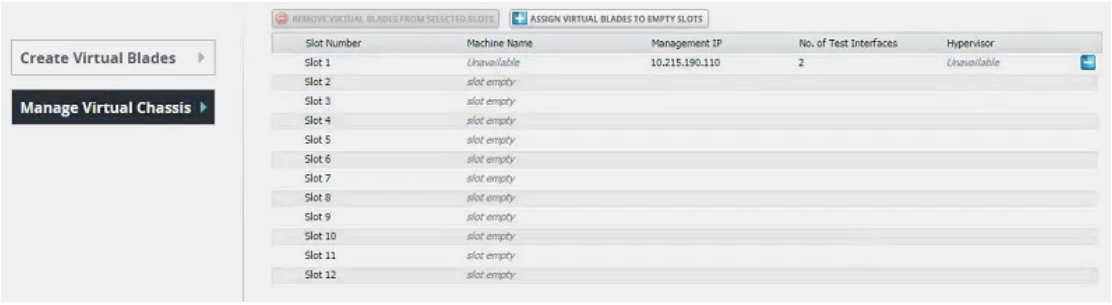
Unassign/Assign a vBlade

Note: To ensure proper vBlade operation, Ixia recommends that vBlades are in the powered ON state before they are unassigned.

To assign or unassign a vBlade:

1. Select **Manage Virtual Chassis**.
2. On the **Assign Virtual Blades To Empty Slots** tab. Select the plus (assign) or minus (unassign) icon that is displayed at the right side of a slot's row (as shown in the image below).

* **Management IP** = The management IP of the vBlade instance



Note: For BPS on AWS - When manually deploying the vBlade instance, you can attach one more network interface to your instance during launch (in addition to the management interface). After you've launched your instance, you can attach more network interfaces using the EC2 console. Please make sure that after you attach more interfaces, you reboot the vBlade instance (using the EC2 console) in order for the changes to take effect.

Note: Unassigning a vBlade will only break the connection between the controller and the vBlade. The vBlade will not be removed or powered off.

This page intentionally left blank.

CHAPTER 5 BPS VE Install on Microsoft Azure RM Services

This chapter describes:

- How to prepare your subscription/location for BreakingPoint solution deployment
- How to deploy BPS VE on Microsoft Azure

Deploy BPS VE to Azure Subscription

All helper scripts can be found at: https://github.com/OpenIxia/BreakingPoint.

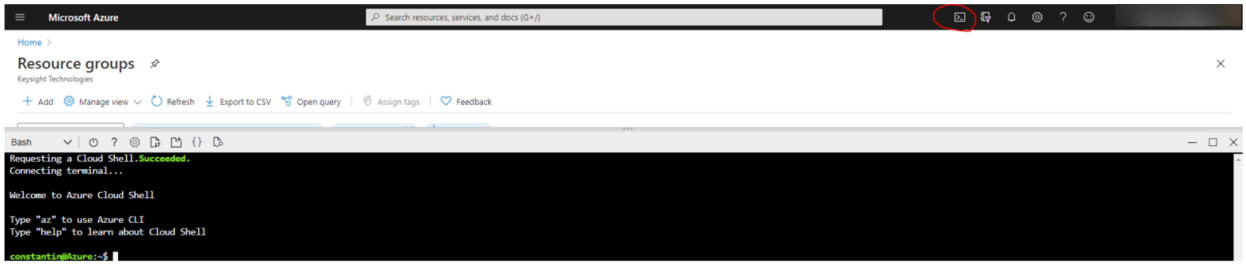
The following script should be used to copy the image:

https://github.com/OpenIxia/azure-bps/blob/<version-branch>/AzureCLI/BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash

> **Important!** Please use the script from the GitHub branch corresponding to the release you want to install. For example, for BPS 9.10 Update 1, you would use the following script: https://github.com/OpenIxia/azure-bps/blob/9.10U1/Deployment/AzureCLI/BreakingPoint_Prepare_VMImages_Azure_CLI_Script.bash

The location of the image files for each specific release are hardcoded into the scripts.

Step 1: Open the Azure Bash terminal.



Step 2: To download the raw file, run the following command:

```
wget https://raw.githubusercontent.com/OpenIxia/azure-bps/master/AzureCLI/BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash
```

Optionally, since this script is directly related to the BPS version that is being copied, it can be renamed with the version number of the image to be better identified in the future.

Example: `mv BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash Ixia_BreakingPoint_9.10.110_Azure_Prepare_VMImages_AzureBash_Script.bash`

Step 3: Add executable and write permissions.

```
chmod 777 Ixia_BreakingPoint_9.10.110_Azure_Prepare_VMImages_AzureBash_Script.bash
```

Step 4: Make corrections if line endings have changed.

```
dos2unix Ixia_BreakingPoint_9.10.110_Azure_Prepare_VMImages_AzureBash_Script.bash
```

Step 5: Execute the script to transfer the image to the current subscription.

```
Ixia_BreakingPoint_9.10.110_Azure_Prepare_VMImages_AzureBash_Script.bash
<destination-resource-group-name> <destinationstorage-account-name>
```

Syntax description:

<**destination-resource-group-name**> Must exist before running the Bash script. Stores the virtual machine images. Syntax rule is alphanumeric, underscore, parentheses, hyphen, period (except at end).

<**destination-storage-account-name**> Length 3 to 24 chars, numbers and lower-case letters only. If it does not exist will be created.

Example 1:

./Ixia_BreakingPoint_9.10.110_Azure_Prepare_VMImages_AzureBash_Script.bash Ixia_Images_RG bpsvhds

> **Note:** Ensure to use a previously created Resource Group in the first argument. The second argument, "destination-storage-account-name", will be created automatically if it does not exist.

> **Note:** The Resource Group zone must be in the same subscription and location with the future desired deployment.

Example 2:

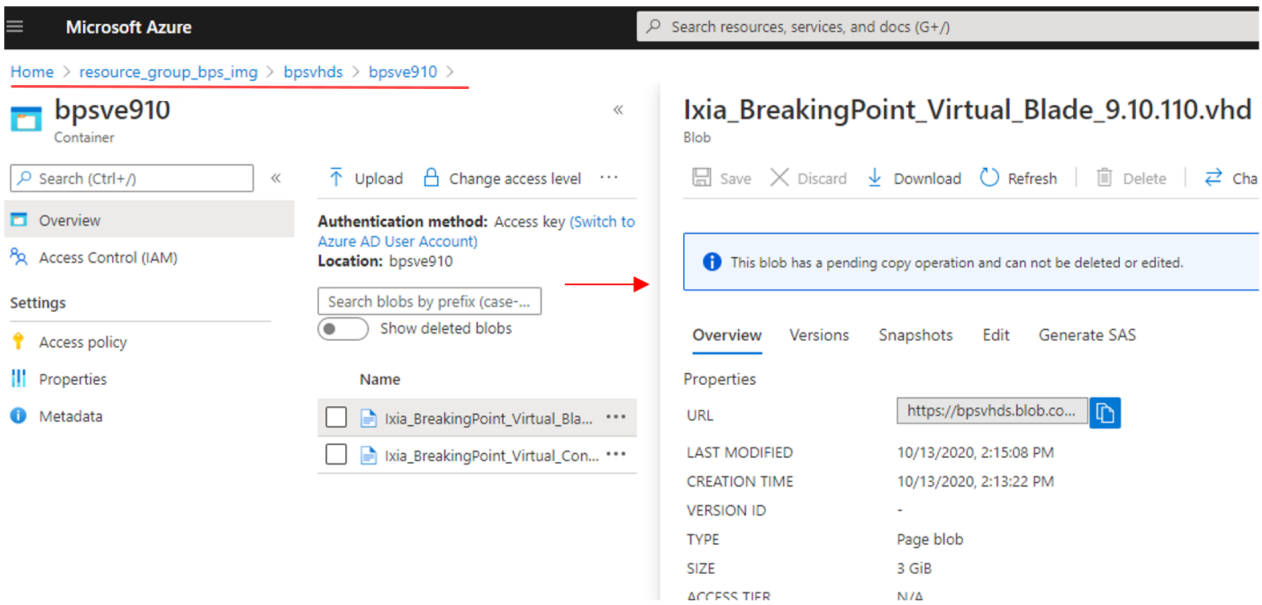
`./Ixia_BreakingPoint_9.10.110_Azure_Prepare_VMImages_AzureBash_Script.bash resource_group_bps_img bpsvhds`

> **Note:** If the destination-storage-account-name does not exist, a validation error will be displayed as shown below. **This error should be ignored**.
> "The Resource 'Microsoft.Storage/storageAccounts/bpsvhds' under resource group 'resource_group_bps_img' was not found. For more details please go to https://aka.ms/ARMResourceNotFoundFix"

```
Bash
constantin@Azure:~$ ./BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash resource_group_bps_img bpsvhds
bash: ./BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash: Permission denied
constantin@Azure:~$ chmod 777 BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash
constantin@Azure:~$ ./BreakingPoint_Azure_Prepare_VMImages_AzureBash_Script.bash resource_group_bps_img bpsvhds
Checking that the destination resource group already exists
Creating a new storage account
ValidationError: The Resource 'Microsoft.Storage/storageAccounts/bpsvhds' under resource group 'resource_group_bps_img' was not found. For more details please go to https://aka.ms/ARMResourceNotFoundFix
Creating: bpsvhds under resource group: resource_group_bps_img
{- Finished ..
  "accessTier": "Hot",
  "allowBlobPublicAccess": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2020-10-13T18:16:21.633473+00:00",
  "customDomain": null,
  "enableHttpsTrafficOnly": true,
  "encryption": {
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
        "enabled": true,
        "keyType": "Account",
        "lastEnabledTime": "2020-10-13T18:16:21.711584+00:00"
```

Wait for the image copy to finish. Some “pending” messages may be displayed during the copy procedure.

During the execution of the script, the 2 images (management and vBlade (np)) that are required for deployment will be copied into the specified Resource Group. An image file that has not finished being copied will display the following information message, “This blob has a pending copy operation and can not be deleted or edited” as shown in the following image.



Note: Image transfer may take 1 hour or more to complete depending on the file location.

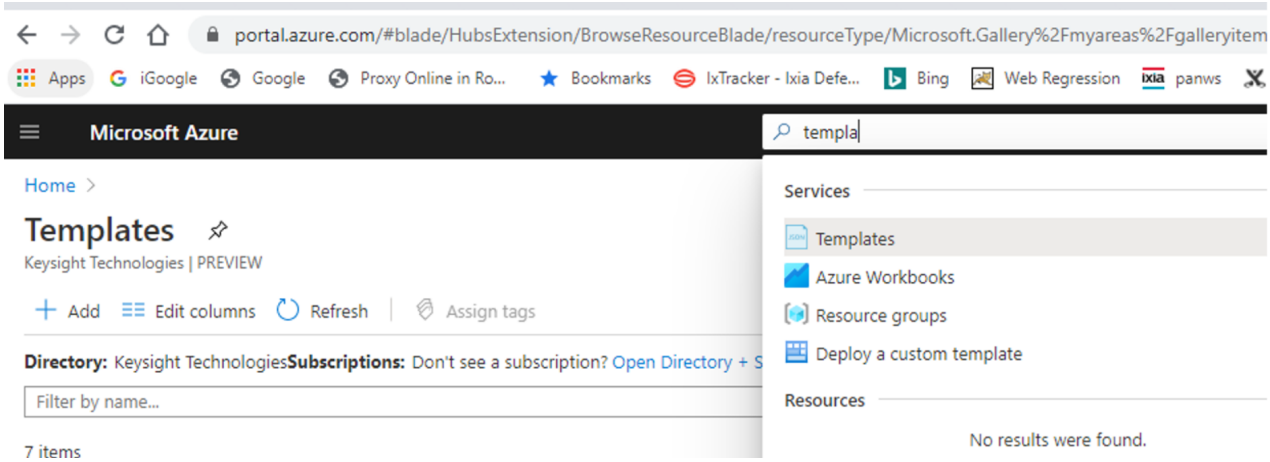
Note: If the shell gets disconnected, the bash script will continue to copy in the background.

Actual deployment

Depending on the setup under test, one of the predefined templates found in GitHub can be used to deploy: https://github.com/OpenIxia/azure-bps/blob/master/AzureResourceManager/BPS/

To deploy an Azure Resource Manager (ARM) JSON template:

Step 1: Open templates from the Azure portal.



Step 2: Add a new template by copying and pasting the JSON content from the desired GitHub template.

https://github.com/OpenIxia/azure-bps/blob/master/AzureResourceManager/BPS/

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Templates > Add template >

ARM Template

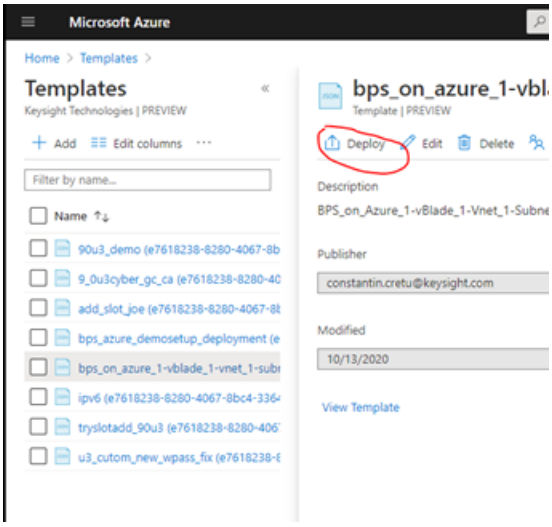
```
1  {
2      "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3      "contentVersion": "1.0.0.0",
4      "parameters": {
5          "IxiaImagesResourceGroupName": {
6              "defaultValue": "Ixia_Images_RG",
7              "type": "String"
8          },
9          "MgmtSecurityRuleSourceIpPrefix": {
10             "defaultValue": "42.42.42.42/32",
11             "type": "String"
12         },
13         "BpsSystemControllerImageName": {
14             "defaultValue": "Ixia_BreakingPoint_Virtual_Controller_9.10.110",
15             "type": "String"
16         },
17         "BpsSystemControllerVmSize": {
18             "defaultValue": "Standard_F4s",
19             "type": "String",
20             "allowedValues": [
21                 "Standard_F4s",
22                 "Standard_F4s_v2"
23             ]
24         },
25         "BpsVirtualBladeImageName": {
26             "defaultValue": "Ixia_BreakingPoint_Virtual_Blade_9.10.110",
27             "type": "String"
28         },
```

OK

Note: The most basic template is: https://github.com/OpenIxia/azure-bps/blob/master/AzureResourceManager/BPS/BPS_on_Azure_1-vBlade_1-Vnet_1-Subnet_Demo_Use_Case_ARM_Template.json. This template will deploy 1 controller and one vBlade with several default options. You can use other available templates if needed or modify them according to your needs.

Step 3: After adding the template, Refresh the Templates view to see the newly added template.

Step 4: To deploy the template, select the template and then select Deploy.



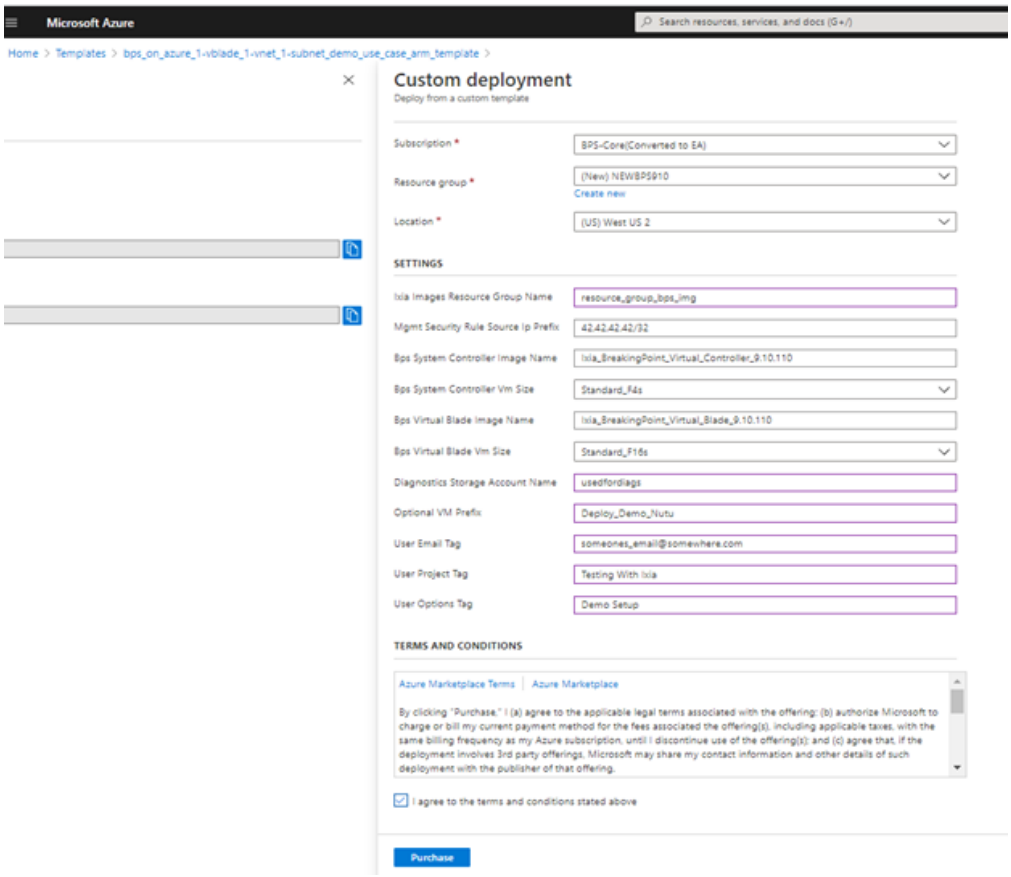
Depending on the template that is selected, different levels of customization are available.

The following is a list of some of the generic customizations that you may want to implement:

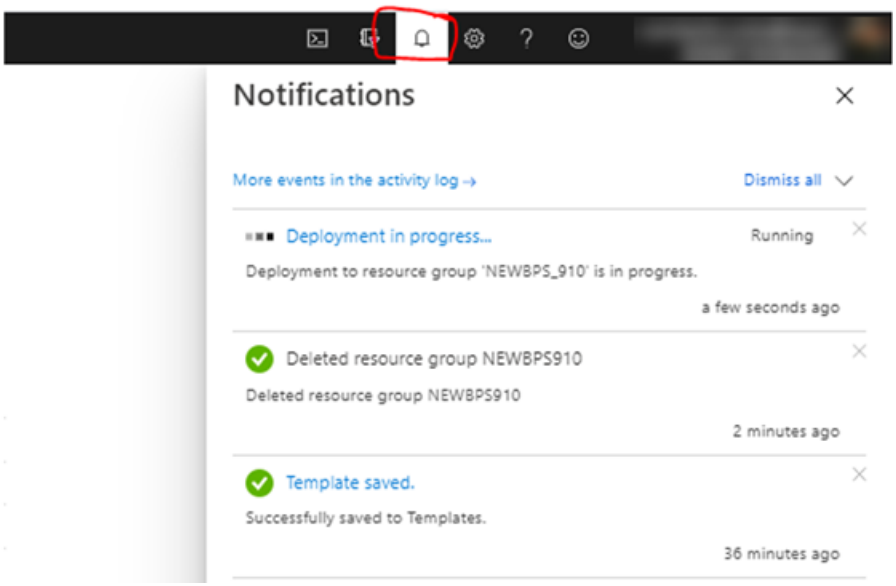
- **Resource Group**: The Azure resource group under which all the elements will be created (Virtual Machines, Networks, Subnets, NICs, Security Profiles, etc.). A new resource group can be created or a preexisting resource group can be used although it is recommended that use a different resource group than the one used to store the images copied from Ixia.
- **Location**: It is the location of the resource group above. It needs to match with the location of the images.
- **Ixia Images Resource Group Name**: The name of the resource group that the Ixia images are copied to during the initial upload performed by the Bash script.
- **Mgmt Security Rule Source Ip Prefix**: The BPS Control web server access will be blocked to all other IP (IP ranges) except this one. It should be the public IP address of the system that will be using the BreakingPoint Interface.
- **BPS System Controller / BPS Virtual Blade Image Name**: These values should match the names of the files copied from Ixia in the first step.
- **BPS System Controller VM Size**: Choose one of the following: Standard_F4/ Standard_F4s.
- **BPS Virtual Blade VM Size**: Choose one of the following: Standard_F16/ Standard_F8s/ Standard_F4s.
- **Diagnostics Storage Account Name**: A storage account that stores the log files from the VMs. It allows VM diagnostics.

Important! This Diagnostics Storage Account needs to exist anywhere in the subscription in the same location zone before the deployment of VMs. If the account does not exist, deployment of the VMs will fail as shown in the following image. The same storage account can be used for multiple deployments.

```
"message": "Storage account 'usedfordiags' not found. Ensure storage account is not deleted and belongs to the same Azure location as the VM."
```

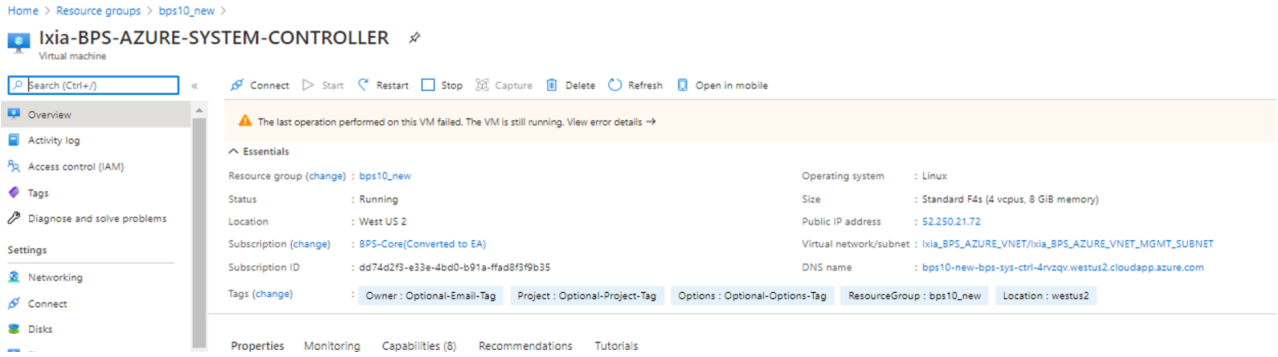


Step 5: Select the check box to agree to the displayed Terms and Conditions. Then select Purchase.



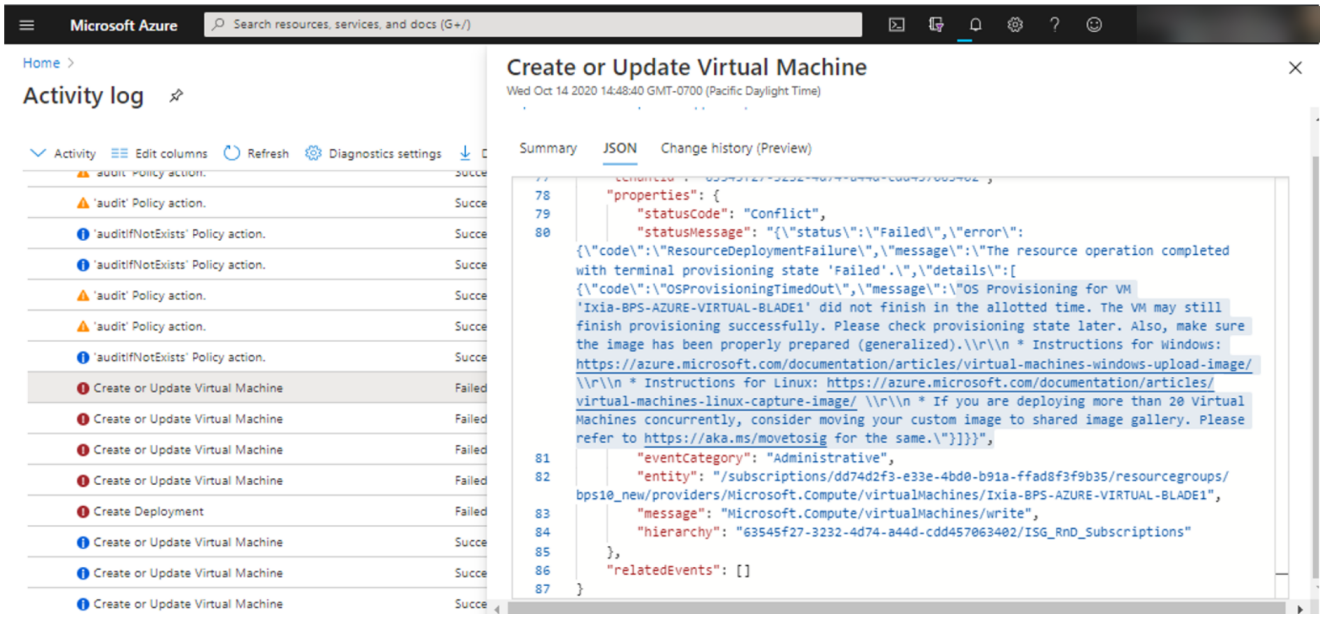
Step 6: Verify the BPS Controller status.

After approximately 10-15 minutes, the deployment should be completed. From the Azure home window, go to the Resource Group chosen for deployment in the template and select the BPS Controller. It should display the public IP and DNS that can be used for access.



Note: The "Deployment in progress" status may display for more than 30 minutes although the Virtual Machines should be starting or already running. You can verify the VM state as shown in the preceding image although Azure automation may not be able to validate the VM provisioning state.

Note: Please ignore the following error during this VM provisioning step. "The resource operation completed with terminal provisioning state 'Failed'. OS Provisioning for VM 'Ixia-BPS-AZURE-VIRTUAL-BLADE1' did not finish in the allotted time." This is a known issue.



Note: Make sure the Deployment Resource Group and the Resource Group containing the images are in the same zone. Azure does not support deploying to different subscriptions or locations for an image. In this scenario, the following error will display, "The Image '/subscriptions/XXXXXX/resourceGroups/image_resource_group/providers/Microsoft.Compute/images/Ixia_BreakingPoint_Virtual_Controller_9.10.110' cannot be found in 'westus2' region."

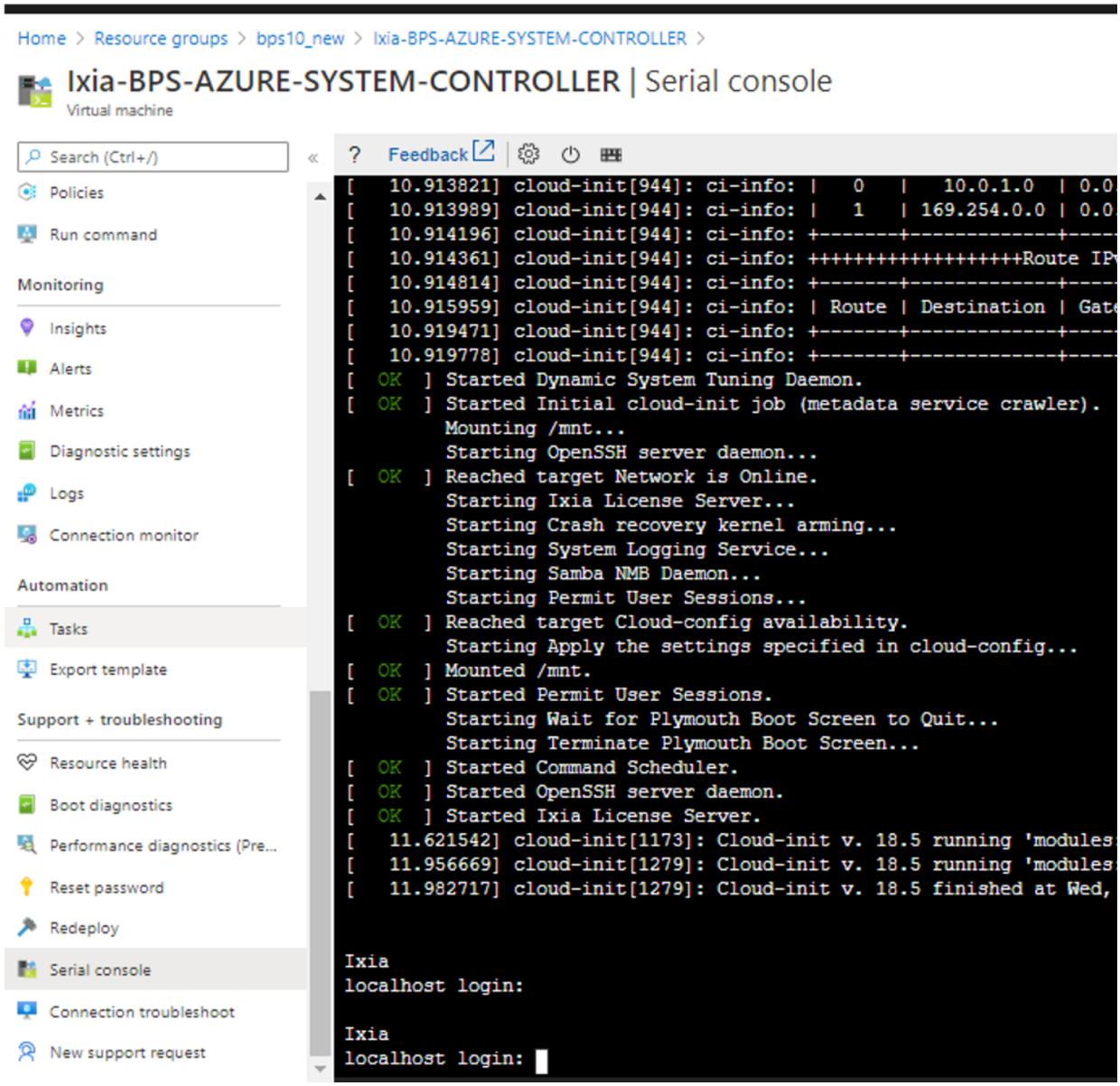
Step 7: Access the BPS controller public IP address using an HTML browser.

Log in using **username**: Admin | **password**: Admin.

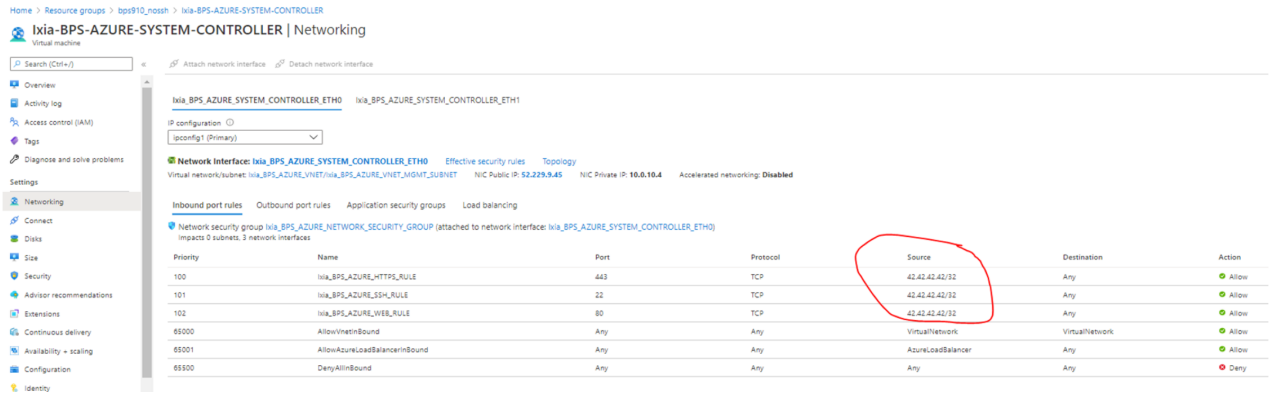
The BPS Controller system public IP provided should appear on the Virtual Machine details for the BPS Controller (see the Ixia-BPS_AZURE-SYSTEM-CONTROLLER image displayed in the previous step for reference).

Use the following troubleshooting steps if you cannot connect to the BPS Controller:

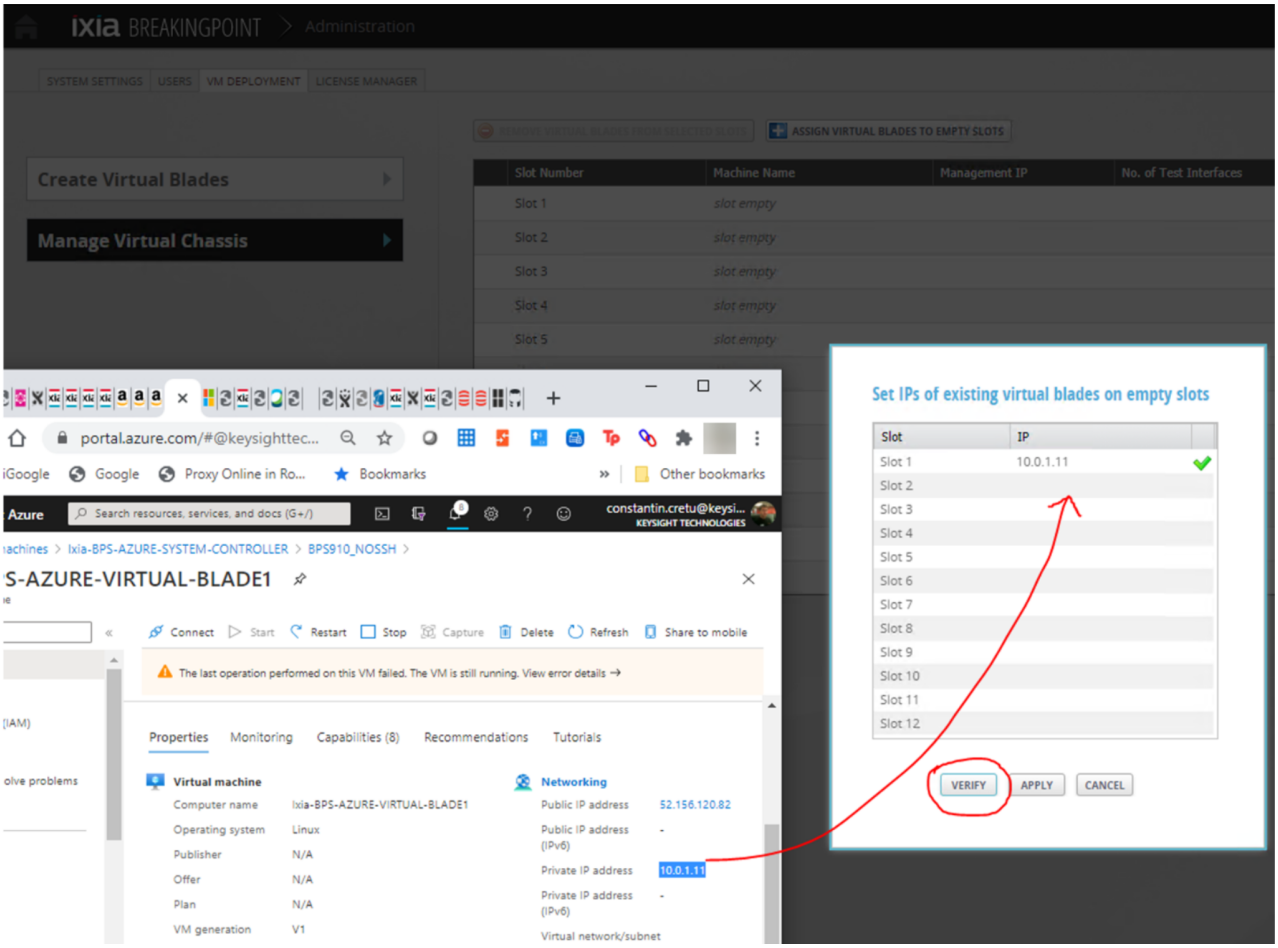
1. Access the VM and make sure it has an IP address and DNS assigned.
2. Access the serial console and check to see if the login prompt is displayed (as shown in the following image).



3. Verify that the your public IP address (where the browser/SSH client that will access the BPS Web is located) is added in the **Security Rules Exceptions** and is not blocked from accessing the BPS System Controller machine.



Step 8: From Administration, add the BPS vBlade.

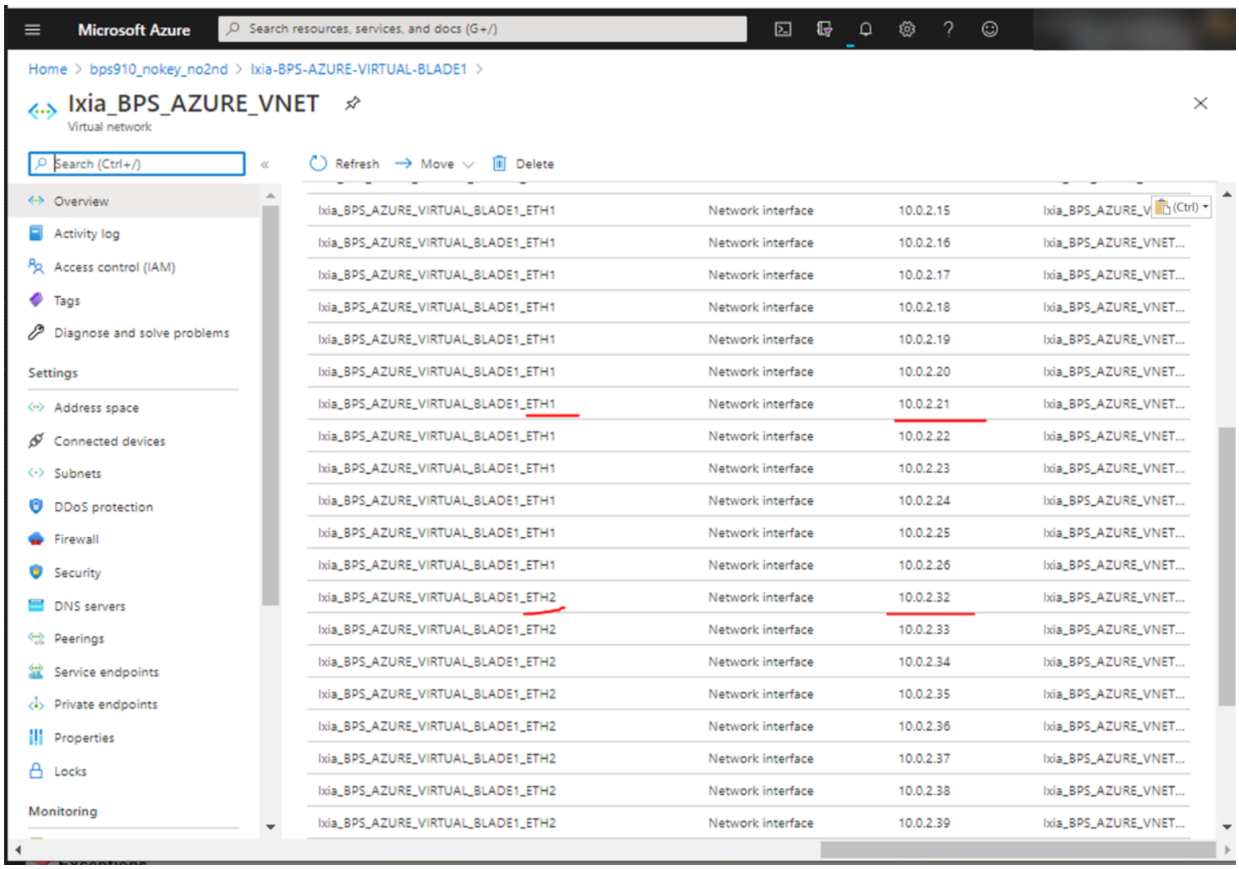


Step 9: From Administration, add the license.

Step 10: Configuration is complete.

Open BPS and start running traffic.

Note: When testing, ensure to use the IP addresses that were defined in the template at the time of deployment.



CHAPTER 6 BPS on Google Cloud Platform

BPS is supported on the Google Cloud Platform (GCP) to provide real-world application and threat simulation for complete performance and security testing. Ixia provides Jinja templates for installation and deployment of BPS on GCP.

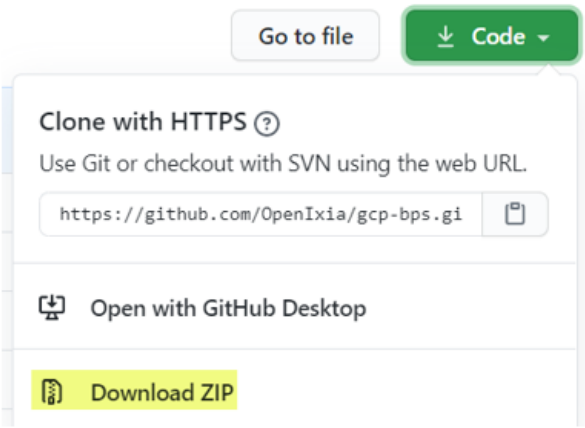
Getting the BPS on GCP Files

BPS on GCP installation templates, test examples and scripts, can be obtained from the OpenIxia GitHub at https://github.com/OpenIxia.

1. Enter https://github.com/OpenIxia in your HTML browser URL field.
2. In the **Find a repository**.. field (Type = All, Language=All), type "gcp-bps".
3. Select the **gcp-bps** link.

Alternatively, you can use the following URL to jump directly to this location: https://github.com/OpenIxia/gcp-bps.git

4. Select the **Code** button dropdown and then select **Download Zip** to download the repository.



5. Unzip the gcp-bps repository to an appropriate location for your environment.

Installation and Deployment

BPS is deployed and installed on the Google Cloud Platform using Jinja templates.

Tip: Template default values can be modified. For example, the IP addresses used for the BPS vController and vBlade instances that will be installed can be customized for your environment.

1. Download the BPS-GCP repository from GitHub as described in Getting the BPS on GCP files.
2. Run the dos2unix utility. The dos2unix command is a simple way to make sure that files that have been edited and uploaded from a Windows machine to a Linux machine work and behave correctly.
   a. **From Cloud Shell run**: ~ $ sudo apt-get install -y dos2unix
3. Locate the following files in the repository in preparation for the installation.

- **DeploymentManager** directory
  - BPS-on-GCP-1-vBlade-Demo-Use-Case-DM-Template.jinja
  - BPS-on-GCP-1-vBlade-Demo-Use-Case-DM-Template.jinja.schema
- **CloudShell** directory
  - GCP_VPC_Network_Peering_Cleanup_Bash_Script.bash

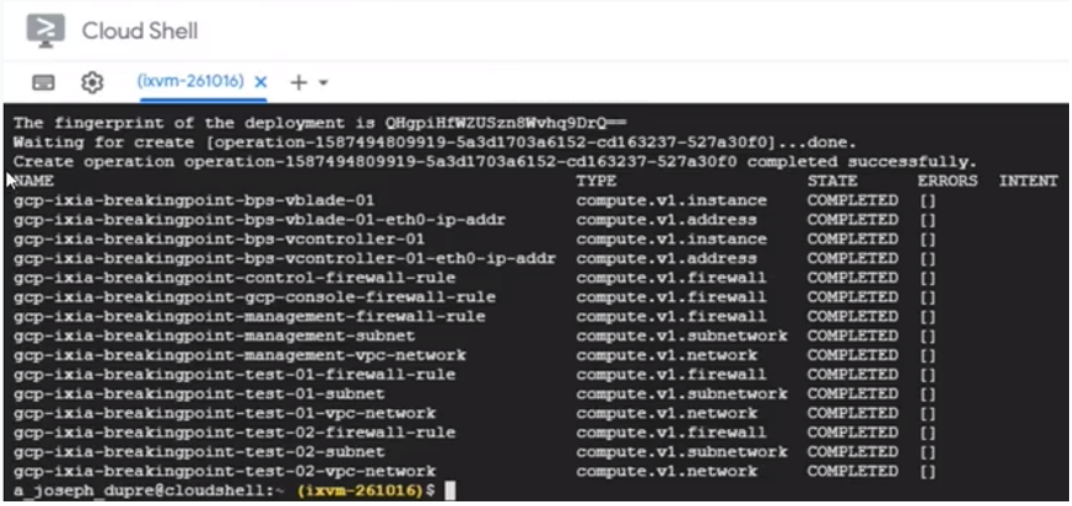
  Note that the cleanup bash script is used to cleanly uninstall VPC network peering.
  - GCP_VPC_Network_Peering_Deployment_Bash_Script.bash

4. **From Cloud Shell run**:

$ clear; gcloud deployment-manager deployments create open-ixia-gcp-bps --template BPS-on-GCP-1-vBlade-Use-Case-DM-Template.jinja

BPS on GCP will be installed and begin to boot up.

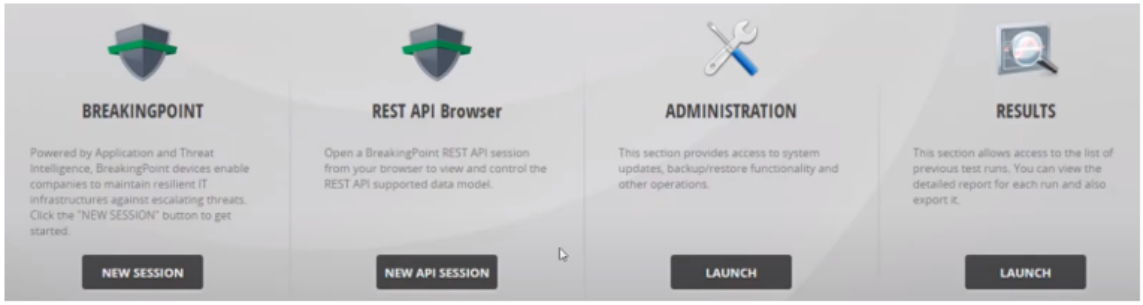
5. Verify successful installation:
   a. The installation of BPS on GCP can be viewed and verified from Cloud Shell as shown in the following example:



Note: Time to initialize - The initial boot from a template installation will take longer that subsequent boot ups due to database initialization and housekeeping.

b. Upon successful installation, the BPS vController and vBlade instances are installed and attached to the network. They will visible in the GCP Console under **Google Cloud Platform** > **VM Instances**.

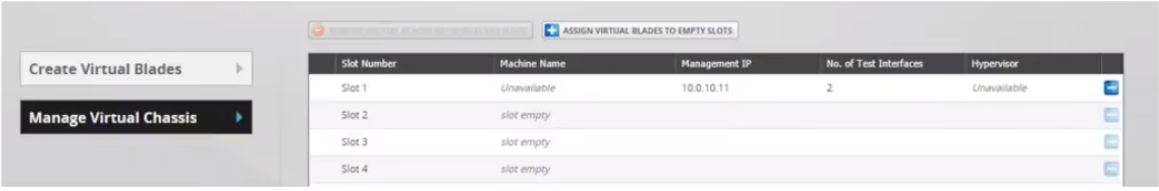
6. In the GCP Console under **Google Cloud Platform** > **VM Instances**, select the vBlade to view the vBlade instance details. The connected NIC cards are displayed. Observe and copy the Primary Internal IP address that is displayed for the **management-vpc-networ**k for future use.
7. Log into BPS VE (vController IP address).
8. The BreakingPoint Dashboard for application selection will display. Select the **Launch** button that is below the **Administration** icon.



9. In BreakingPoint, associate the vBlade with the vController which you are currently logged into.
   a. Select **VM Deployment**.
   b. Select **Manage Virtual Chassis**.
   c. Select Assign **Virtual Blade** to **Empty Slots**.

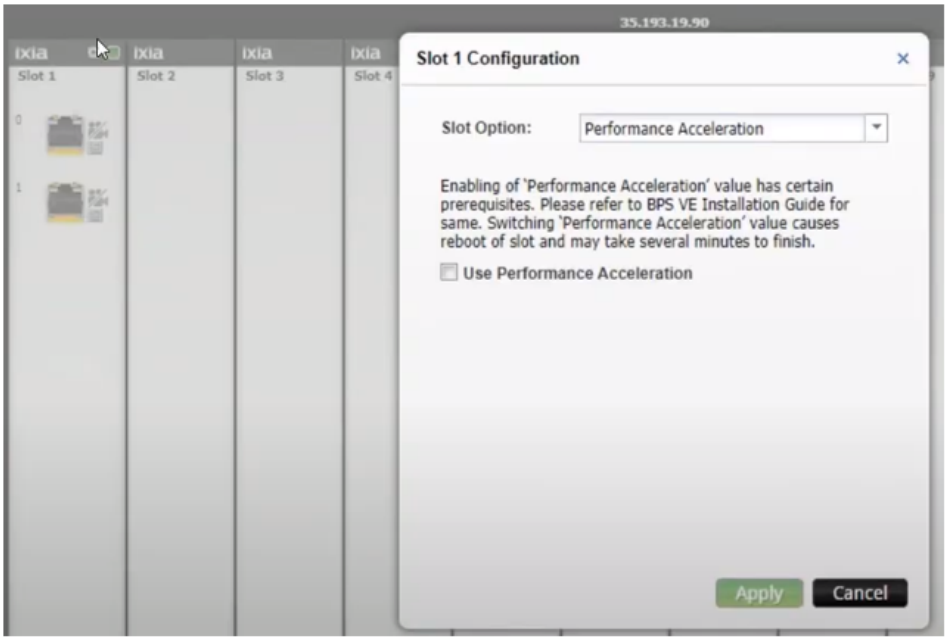
d. In the **Slot 1** field, paste or type in the IP address that you copied earlier from the vBlade **management-vpc-network** NIC.

e. Select **Apply**.



10. Return to the BPS Dashboard area. Select the **New Session** button that is below the BreakingPoint icon.
11. In BreakingPoint, go to the **Chassis** view.
12. Select the gear on the card and then configure the Slot Option to use **Performance Acceleration** and then select the **Use Performance Acceleration** check box.

This option is selected to get maximum performance out of instance types. DPDK needs to be turned on. The virtual card will reboot and then automatically re-attach in high performance mode.



13. Select **Apply**.
14. Log into the web interface of the system (username: admin, password: admin).
15. At the Ixia Web Platform, select the **BreakingPoint** icon.
16. At the BreakingPoint Dashboard, select the **Gear** icon > **Administration** > **System**.
17. Select the **License Manager** tab. Add licenses on your local server or select an external license server.

18. From the BPS Chassis view, take ownership of the performance acceleration ports so that they can be used for your testing.
19. Make the IP address ranges that are configured in Network Neighborhood match the IP addresses that are assigned to the VM instances. The IP address ranges are shown in the CCP Console under **VM instance details** in the **Primary internal IP - Alias IP Ranges** fields.
20. Connect the two Virtual Private Clouds that were created during the installation. **From Cloudshell**, run the script to create and connect the two VPCs that have been created.
    a. $ ./GCP_VPC_Network_Peering_Cleanup_Bash_Script.bash
    b. The connectivity can be verified under **VPC network peering** in the GCP Console. The fields, **Your VPC network** and **Peered VPC network** will show that the network peers are connected to each other.

> **Note:** GCP will not allow you to connect 2 NICs of the same instance to subnets of the same VPC.

This page intentionally left blank.

CHAPTER 7 Nested Environment Installation

This sections provides a detailed description of the steps required and resolve problems that may occur when attempting to deploy a vBlade in a nested OpenStack environment.

1. Log in into the Virtual Blade and check the "ixvmbps.log" in /etc/var/log. If the log has the following error: "This system does not support "SSSE3", then the following action needs to be performed:
   a. Nested OpenStack Setup-
      i. Edit "/etc/nova/nova.conf"
      ii. Add under "[libvirt]" - cpu_mode = host-model
      iii. Restart Nova services
      iv. Restart the vBlade
      v. Add the vBlade

   b. KVM from UI-
      i. Select the specific vBlade
      ii. Edit the vBlade settings
      iii. Go to "Processor"
      iv. Under "Configuration", set the "Model" to "Copy host CPU configuration"

   c. KVM from CLI-
      i. virsh edit <vBlade_name>
      ii. Add the following:

```
<cpu mode='host-model'>
<model fallback='allow' />
</cpu>
```

iii. Restart the vblade

iv. Add the vblade

2. To solve problem 2, log in into the Compute and Controller Node:
   a. Edit “/etc/nova/nova.conf”
   b. Add under "[neutron] " - allow_duplicate_networks = True
   c. Restart the Controller and Compute Node

CHAPTER 8 SR-IOV Installation and Configuration

This chapter describes SR-IOV installation and configuration.

SR-IOV Installation and Configuration on KVM

This section explains the installation and configuration steps for SR-IOV and PCI-Passthrough on Linux CentOS 7 64-bit for the following:

- Installation and Configuration for Intel

Installation and Configuration for Intel

Installation and Configuration on Linux CentOS 7 64-bit includes:

- SR-IOV Installation and Configuration
- PCI-Passthrough Installation and Configuration

SR-IOV Installation and Configuration

Hardware Requirements

The minimum hardware requirements to configure SR-IOV are:

- An Intel Ethernet Network Adapter supporting SR-IOV
- A server platform that supports Intel Virtualization Technology for Directed I/O (VT-d) and the PCI-SIG Single Root I/O Virtualizations and Sharing (SR-IOV) specification

Software Requirements

The software requirements to configure SR-IOV are:

- KVM (QEMU) over CentOS 7.0 64-bit

Recommended Driver Version

Refer to the Certified and Compatible Cards section in the BPS VE Install Guide to know about the recommended driver version.

Server Setup

1. Install Linux CentOS 7 64-bit.
2. By default, I/O Memory Management Unit (IOMMU) support is not enabled in the Linux CentOS 7 64-bit distribution. IOMMU support is required for a VF to function properly when assigned to a VM. The following kernel boot parameter is required to enable IOMMU support for Linux kernels:

   ```
   intel_iommu=on
   ```

   This parameter can be appended to the `GRUB_CMDLINE_LINUX` entry in `/etc/default/grub` configuration file.
3. Update grub configuration using the `grub-mkconfig` command.
4. Reboot the server for the iommu change to take effect.

   Skip this step if `cat /proc/cmdline shows intel_iommu=on`. After doing all the above steps, if issuing the command `cat /proc/cmdline` does not also show the `intel_iommu` option, this means that the `intel_iommu` option was not loaded into kernel and the `grub.cfg` was not generated from the `/etc/default/grub` configuration file as mentioned above.

   To update the `GRUB 2` configuration file manually, use the `grub2-mkconfig -o` command as follows:

- On BIOS-based machines, run the following command as root on hypervisor:

  ```
  ~]# grub-mkconfig -o /boot/grub/grub.cfg
  ```

- On UEFI-based machines, run the following command as root on hypervisor:

  ```
  ~]# grub-mkconfig -o /boot/efi/EFI/ubuntu/grub.cfg
  ```

  Run the `cat /proc/cmdline` again to check if the `intel_iommu` option has been enabled.

5. Run the `lspci` command to verify that Ethernet Controller in the server is available.
6. The Linux CentOS 7 64-bit installation does not create Virtual Functions (VFs) by default. The server adapters support from 1 to 64 maximum VFs (depending on the platform) per PF (Physical Function). You can create the VFs in the following two ways:
   a. modprobe

      For 1G: `modprobe igb max_vfs=8,8`

      For 10G: `modprobe ixgbe max_vfs=8,8`

      For 40G: `modprobe i40e max_vfs=8,8`

      This method applies to activating eight VFs per PF.
   b. Updating the sriov_numvfs device configuration `echo 8 > /sys/class/net/[device_name]/device/sriov_numvfs`

      [device_name] = name of the interface on which you want to enable the VFs

Example: `echo 8 > /sys/class/net/eth1/device/sriov_numvfs`

7. Module options are not persistent from one boot to the next. To ensure that the desired number of VFs are created each time the server is power-cycled, append the above command to the `rc.local` file, which is located in the `/etc/rc.d/` directory. The Linux OS executes the `rc.local` script at the end of the boot process.

Warning:

Errors and informational messages during ixg / ixgbe / i40e driver load are logged in the `/var/log/`messages file. It is a good practice to review this file to confirm that the driver loaded successfully without warnings or errors.

8. Run the `lspci` command to confirm that the VF was successfully created.

   Now you can start adding the Virtual Functions inside the Virtual Blades.

9. In the **Virtual Machine** window (virt-manager), select **Add Hardware** to open the **Add New Virtual Hardware** wizard.
10. Select **PCI Host Device** and then select a virtual function that you just activated. Now you can switch on the VM.
11. Run the `lsmod` command on the VM to check whether the igbvf / ixgbevf / i40evf driver was loaded properly.

PCI-Passthrough Installation and Configuration Server Setup

1. Install Linux CentOS 7 64-bit.
2. Deploy a machine on this setup and open the it from the Virtual Machine Manager.
3. Go to the show machine info section (Select the bulb).
4. To open the **Add New Virtual Hardware** wizard, select **Add Hardware**.
5. Select **PCI Host Device** and then select the physical port from the NIC available in the server.
6. Select **Finish**.

   You can see the new PCI device inside the machine.

7. Switch on the machine.

SR-IOV / PCI-Passthrough Limitations

SR-IOV / PCI-Passthrough Not Supported on Management while bridges / vSwitches / Open vSwitch are configured on Test interfaces

Having SR-IOV virtual functions or PCI-Passthrough devices configured as management networks on the Virtual Controller / Virtual Blade are not supported, if the test/backplane networks are configured with virtual switches (VMware) or bridges/OVS (KVM/OpenStack).

Malicious Driver Detection Feature

When the malicious driver detection feature is enabled on ixgbe interfaces, running Raw or Ethernet/VLAN traffic will cause the interfaces to go down.

To disable this feature, run the following command on KVM / OpenStack platforms:

```
insmod ixgbe.ko MDD=0,0
```

Setup MTU 9000 on the Physical Function and Virtual Functions

In order to run jumbo frames tests you will need to configure MTU 9000 on the Physical Functions and Virtual Functions (VFs).

Having MTU mismatches between the PFs and VFs will cause traffic to get dropped inside the Intel board.

Changing the MTU can be done in the following way:

- Physical function

  ```
  ifconfig INTERFACE_NAME mtu 9000
  ```

- Virtual function

  The MTU configuration is controlled from within the Virtual Blade so please make sure that you have the same MTU as the Physical Function.

SR-IOV Installation and PCI-Passthrough Installation and Configuration

This section explains the installation and configuration steps for SR-IOV and PCI-Passthrough on VMware ESXi 6.0 for the following:

- Installation and Configuration for Intel

Installation and Configuration for Intel

Installation and Configuration on VMware ESXi 6.0 includes:

- SR-IOV Installation and Configuration
- PCI-Passthrough Installation and Configuration

SR-IOV Installation and Configuration Hardware Requirements

The minimum hardware requirements to configure SR-IOV are:

- An Intel Ethernet Network Adapter supporting SR-IOV
- A server platform that supports Intel Virtualization Technology for Directed I/O (VT-d) and the PCI-SIG Single Root I/O Virtualizations and Sharing (SR-IOV) specification

Software Requirements

The software requirements to configure SR-IOV are:

- VMware ESXi 6.0

Recommended Driver Version

Refer to the **Certified and Compatible Platform Versions** section in the *IxVM Reference Guide* to get information on the recommended driver version.

Server Setup

To setup the server for installing and configuring SR-IOV:

1. Install VMware ESXi.
2. Enable SSH on the host to access the console for CLI configuration.
3. Run the `lspci` command to verify that the Ethernet Controller is available in the server.

Note:

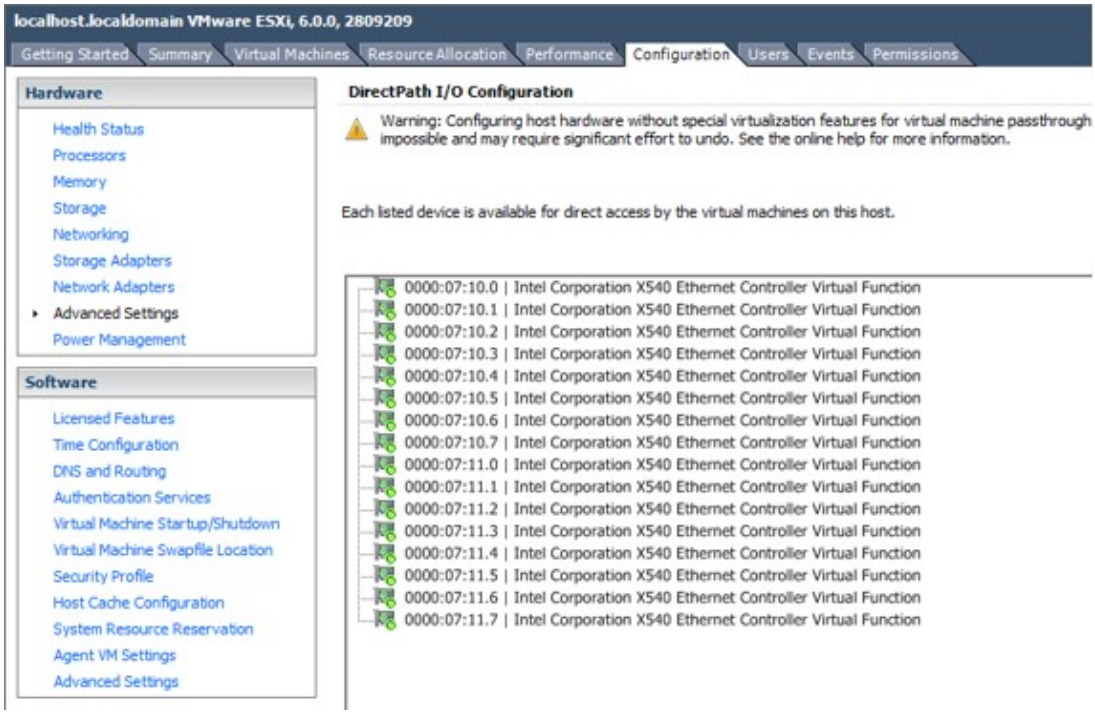
By default, the VMware ESXi installation does not create a VF. The server adapters support from 1 to 64 maximum VFs.

Run the following command to activate SR-IOV.

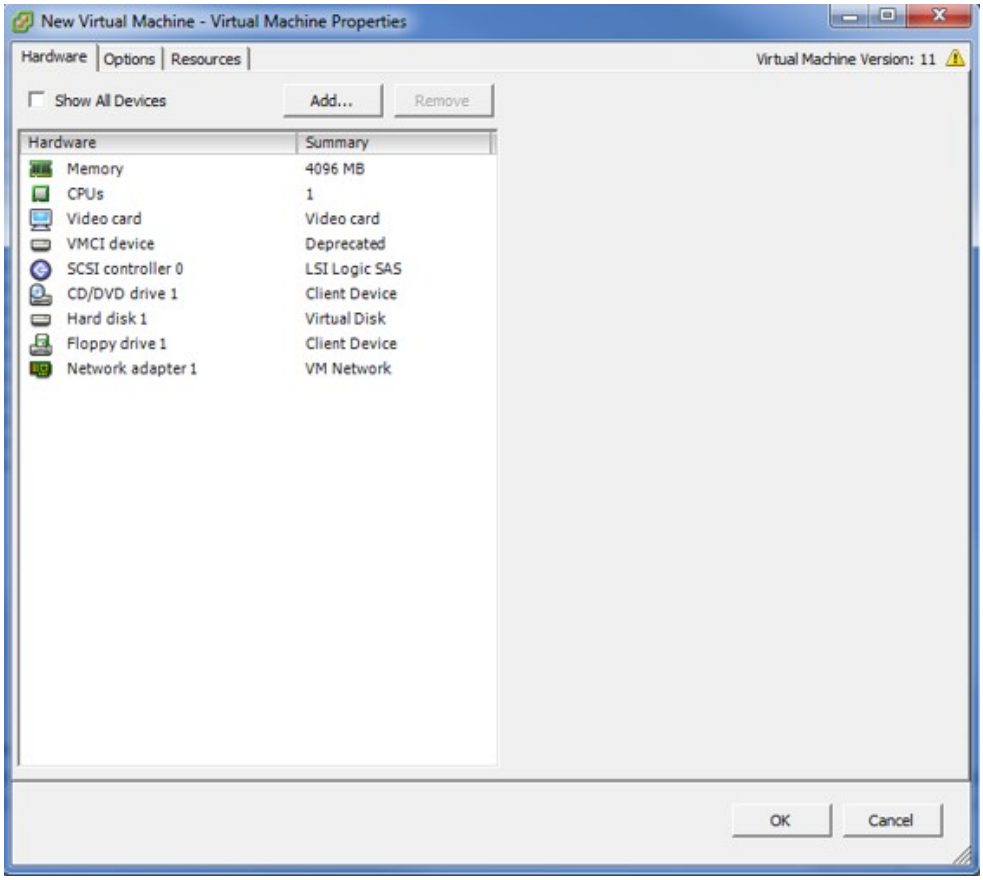
For 10G: `esxcfg-module ixgbe -s max_vfs=8,8`

For 40G: `esxcfg-module i40e -s max_vfs=8,8`

4. Reboot the server.
5. Run the `lspci` command to confirm that the VF was successfully created.
6. Check the VMWare vSphere Client to confirm that you are able to see the VFs.
7. Select **Configuration > Advanced Settings**.



Now you can start adding the VFs inside the VM cards.

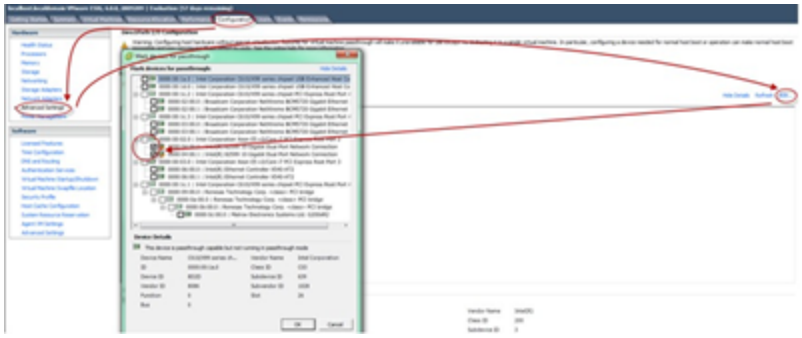


8. Select **Add > PCI Device**. Select **Next**.
9. Select a Virtual Function from the list and then select **Finish**.

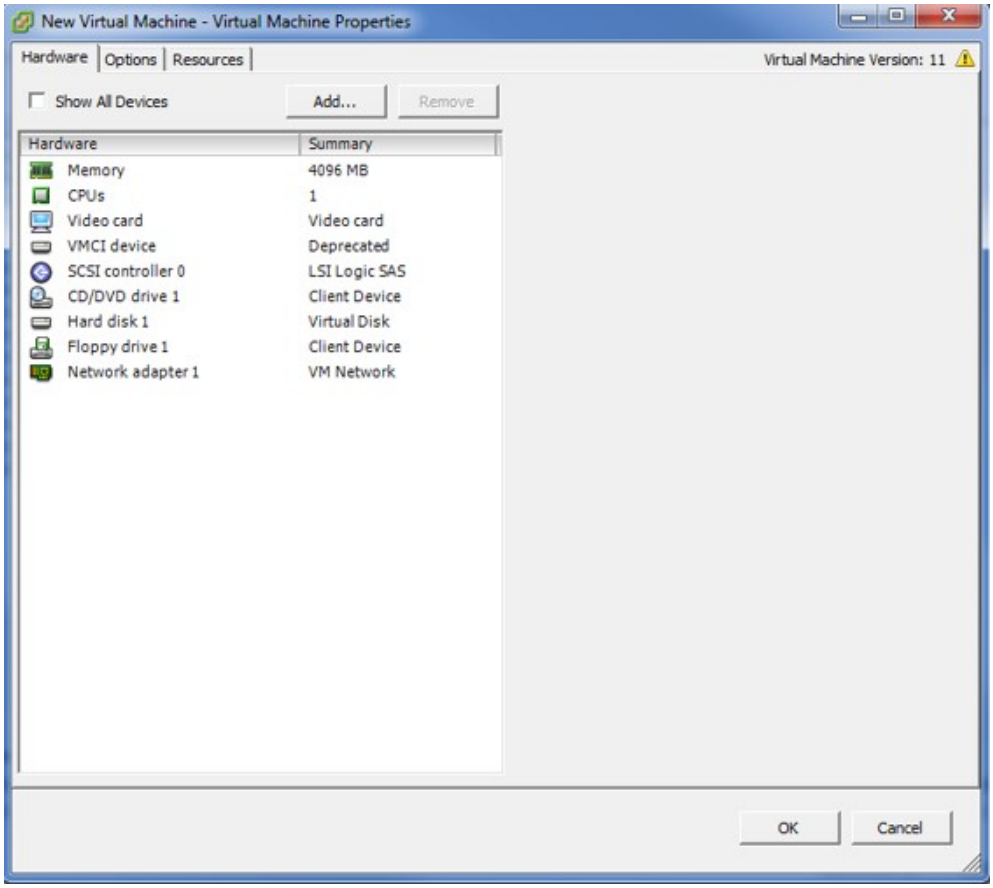
   You can now switch on the system.

PCI-Passthrough Installation and Configuration Server Setup

1. Install VMware ESXi.
2. Enable SSH on the Host to access the console for CLI configuration.
3. Run the `lspci` command to verify that the Ethernet Controller is available in the server.
4. Select **Configuration > Advanced Settings > Edit** to mark the devices that you want for PCIPassthrough.



5. Reboot the server.
6. Now you can start adding the PCI devices inside the VM cards.



7. Select **Add > PCI Device**. Select **Next**. You can now switch on the system.

SR-IOV / PCI-Passthrough Limitations

SR-IOV / PCI-Passthrough Not Supported on Management while bridges/vSwitches/Open vSwitch are configured on Test interfaces

Having SR-IOV virtual functions or PCI-Passthrough devices configured as management networks on the Virtual Controller / Virtual Blade are not supported, if the test/backplane networks are configured with virtual switches (VMware).

Setup MTU 9000 on SR-IOV interfaces

Maximum Transmission Unit (MTU) setup is required for different testing scenarios when the MTU size must be increased/decreased from the standard 1500 on the ESXi hypervisor network interface.

The following steps explain how to setup MTU 9000 on SR-IOV interfaces:

1. Create a new vSwitch and add the desired interface (SR-IOV).
2. Edit the newly created vSwitch and set MTU to 9000.
3. Remove the vSwitch created in step 1.
4. Check in the Command Line Interface (CLI) that MTU has the configured value as follows:

```
[root@localhost:~] esxcli network nic list
Name PCI Device Driver Admin Status Link Status Speed Duplex MAC Address MTU
Description
------ ------------ ------ ------------ ----------- ----- ------ -------------
---- ---- --------------------------------
vmnic0 0000:01:00.0 ixgbe Up Up 10000 Full 24:6e:96:33:37:e8 9000 Intel
Corporation Ethernet Controller 10 Gigabit X540-AT2
vmnic1 0000:01:00.1 ixgbe Up Up 10000 Full 24:6e:96:33:37:ea 9000 Intel
Corporation Ethernet Controller 10 Gigabit X540-AT2
```

Deploy a vBlade with SR-IOV Virtual Functions

Perform the following procedure to deploy a vBlade on a network adapter that supports Virtual Function single root I/O virtualization (SR-IOV).

To use virt-install to assign a PCI device, use the --host-device parameter.

1. Identify the device, getting the available interfaces.

```
ls /sys/class/net/

br0  br_1  br-em1  br-em2  em1  em1_0  em1_1  em2  em2_0  em2_1  em3  em4  lo  p2p1  p2p2  virbr0  virbr0-nic  vnet0
```

2. Find out which of these interfaces support SR-IOV (it must have a sriov_numvfs file).

```
ls /sys/class/net/<em1>/device/sriov_numvfs

/sys/class/net/<em1>/device/sriov_numvfs

<em1> is the interface name
```

3. Find the mapping of the VF to the PCI device.

```
ls -l /sys/class/net/<em1>/device/virtfn*


/sys/class/net/em1/device/virtfn0 -> ../0000:01:10.0
/sys/class/net/em1/device/virtfn1 -> ../0000:01:10.2

<em1> is the interface name
```

4. With the PCI id of the virtual function known (0000:01:10.2), append 'pci_' in front of it and replace any ':' and '.' with '_'. For example, "0000:01:10.2" becomes "pci_0000_01_10_2".
   a. Now a Virtual Load Module can be deployed using the virt-install command as shown in the following screenshot.

```
virt-install --name VirtualLoadModule --ram 4096 --vcpus 2 --network bridge=br0,model=virtio --host-device=pci_0000_01_10_1--host-device=pci_0000_01_

Where:
        --disk                                  =    location of the qcow2 disk
        --name                                  =    name of the virtual machine that you are going to create
        --ram                                   =    amount of RAM assigned to the machine. Should be 4GB RAM
        --vcpus                                 =    number of vCPUs assigned to the machine
        device=disk                             =    device should be disk type
        bus=virtio                              =    bus type. Should be virtio only.
        format=qcow2                            =    image format. Should be qcow2.
        --boot                                  =    where to boot from. Should be "hd"
        --vnc --noautoconsole                   =    to enable the console
        --network bridge=br0,model=virtio       =    adding a management network bridge, model virtio
        --host-device=pci_0000_01_10_1          =    adding a test network using PCI identifier
        --serial unix,path=/tmp/LinuxChassis    =    unix serial device in order to get the Discovery Service working fine
        --serial pty                            =    pty serial device for "virsh console" access
```

CHAPTER 9 Disk Expansion

Disk Expansion allows you to resize a disk file to provide additionally required disk space to a BPS VE Controller VM.

Note: This enhancement is only for the BPS VE vController; no support is provided for the expansion of a vBlade VM.

Supported platforms

Disk expansion is available on all platforms: ESXi, KVM, OpenStack, Azure and AWS.

Methods for expanding the disk.

- CLI: ESXi, KVM and OpenStack
- GUI: ESXi, OpenStack, Azure and AWS

Note: For all supported platforms, regardless as to whether the expansion is performed from the CLI or GUI, the initial boot time after expansion will take longer than the average boot time. Boot time after expansion ranges from 2 min: 30 seconds on ESXi to almost 6 min on the other supported platforms.

Disk Expansion using the CLI

Important! All of the following steps for disk expansion MUST be done while the vController is offline, otherwise data corruption may occur.

The commands used to modify the disk using the CLI differ from platform to platform, but the principles are basically the same:

- Resize the disk file to provide additionally required disk space to the VM
- Expand the root partition inside the VM
- Expand the root filesystem

Follow the procedure for your specific platform:

- KVM
- VMware ESXi
- OpenStack

KVM

> ❗ **Important!** All of the following steps for disk expansion MUST be done while the vController is offline, otherwise data corruption may occur.

1. The first step requires the installation of the libguestfs-tools package using the appropriate command for your operating system.

(For CentOS)

```
sudo yum -y install libguestfs-tools
```

(For Ubuntu)

```
sudo apt-get install libguestfs-tools
```

2. A KVM guest needs a disk file to run in order to store its own data. For our example, we will use a .qcow2 file named, vController.qcow2.In this example we will resize the virtual disk to allow for an additional 5G of disk space.

```
qemu-img resize vController.qcow2 +5G
```

3. Step 2 extended the .qcow2 image. After the .qcow2 image is extended, any VM which runs based on this .qcow2 file will have some unallocated disk space. This space is unavailable for use, since we cannot add that space to the root partition at VM runtime. Therefore, libguestfs-tools puts at our disposal a mechanism which alters the partition table and extends a certain partition and the filesystem which resides on it.

   To reorganize the partition table, this tool needs a reference image which will allow the resize to occur without jeopardizing data. In this step, we will make a copy of the original image and rename the copy as vController-exp.qcow2.

```
cp vController.qcow2 vController-exp.qcow2
```

4. Finally, we will resize the partition and the root filesystem on it.

```
virt-resize -d --expand /dev/vda1 vController-exp.qcow2 vController.qcow2
```

This command uses the copy image in order to alter the partition table on the disk. At the same time, it extends the filesystem which resides on that partition (in our example it is /dev/vda1, -d is for debug messages). Consequently, we can erase vController-exp.qcow2, since we have no need for it. After those operations, we can use our qcow2 file to deploy VMs, and then observe the corresponding changes by running:

- fdisk -l (The parameter is lowercase L. Allows you to observe the extra disk space, and see the /dev/v(s)da1 partition enlarged.)
- df -h (You can now observe that the root filesystem uses all available space on /dev/v(s)da1.)

VMware ESXi

> ❗ **Important!** All of the following steps for disk expansion MUST be done while the vController is offline, otherwise data corruption may occur.

Notes

- Disk expansion is not available for ESXi 5.5, neither from GUI, nor from CLI. The commands used to modify the disk from CLI can be applied, no error is returned, but expansion will not take effect.
- The disk size modified in the CLI doesn't match the disk size displayed in the GUI for a version of ESXi newer than 6.0, but the memory can be used. The disk size is correctly shown in CLI and GUI when it is modified from GUI.
- When exporting the OVF file of the Controller that had the disk size modified in the CLI, the exported OVF disk size doesn't match the actual disk size set in the CLI. This scenario requires editing the OVF file by modifying the disk size with the value that was set in CLI before deploying a new VM.

Required packages:

- Vmkfstools - If you to make the expansion from the CLI (usually, an ESXi hypervisor has this package included in its vCLI).

Introduction:

On ESXi, the process of expansion is fully automated, through a custom service, namely expand_disk. The service runs at boot time and detects if there is unallocated disk space. By unallocated disk space we mean at least 1GB of disk space. If such space is detected, the service performs an alteration of the disk partition table in accordance with the free disk space. Mainly, it extends the root partition until we have no more free space, reserving 1GB for swap partition. Therefore, the new partition will have TOTAL_DISK_SPACE - 1GB.

Expansion Procedure

1. First, we check for the vController's disk name and associated path from the GUI so we know where to find it in the CLI.
   a. Then we access the CLI.
   b. After accessing "/vmfs/volumes" we add the associated path information obtained from the GUI until we have the full path to the disk file (which has the same name as the vController + .vmdk).
2. To expand the disk file, run the following command. Note that file is extended to 25G, it will NOT be extended by an additional 25G.

```
vmkfstools -X 25G FileName.vmdk
```

OpenStack

Important! Due to an OpenStack specific issue, we were not able to perform the recommended resize instance option through the CLI. See the following for more information: https://www.marksei.com/openstack-resize-instance-no-valid-host-found/

> **Important!** All of the following steps for disk expansion MUST be done while the vController is offline, otherwise data corruption may occur.

1. The first step requires the installation of the libguestfs-tools package using the appropriate command for your operating system.

(For CentOS)

```
sudo yum -y install libguestfs-tools
```

(For Ubuntu)

```
sudo apt-get install libguestfs-tools
```

2. An OpenStack guest needs a disk file to run in order to store its own data. In this scenario, the file is a disk file which can be found on the compute node where the VM was deployed. Run the following commands to locate the disk file.

```
openstack server show INSTANCE_ID, OS-EXT-SRV-ATTR:host attribute
```

The location on the compute node is:

/var/lib/nova/instances/INSTANCE_ID/disk

3. In this step we will resize the virtual disk to provide and additional 5G of disk space.

```
qemu-img resize disk +5G
```

4. Now we need to make a copy of the original image, renaming the copy disk-orig, in order to resize the disk without jeopardizing data

```
cp disk disk-orig
```

Finally, we can resize the partition and the root filesystem on it.

```
virt-resize -d --expand /dev/vda1 disk-orig disk
```

As we can see, the command uses the copy image, in order to alter the partition table on our disk at the same time, it extends the filesystem which resides on that partition. In our example it is /dev/vda1, (-d is for debug messages). Consequently, we can erase disk-orig, since we have no need for it. After these steps are completed, we can use our disk file to deploy VMs, and observe the corresponding changes by running:

- fdisk -l (The parameter is lowercase L. Allows you to observe the extra disk space, and see the /dev/v(s)da1 partition enlarged.)
- df -h (We can now observe that the root filesystem uses all available space on /dev/v(s)da1

Disk Expansion using the GUI

> **Important!** The disk controller must be shutdown before performing any of the disk expansion procedures described below.

The steps required to expand the disk size from GUI, differ from platform to platform.

Follow the procedure for your specific platform:

VMware ESXi

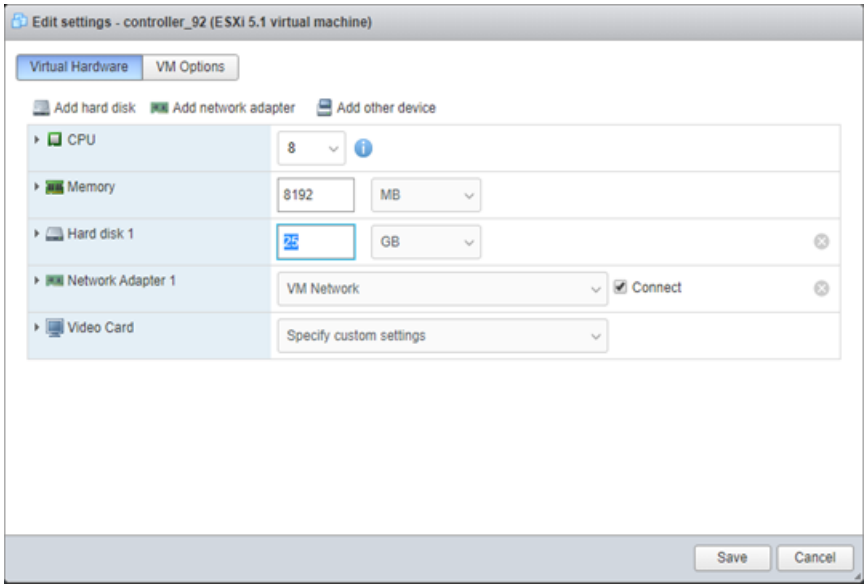
OpenStack

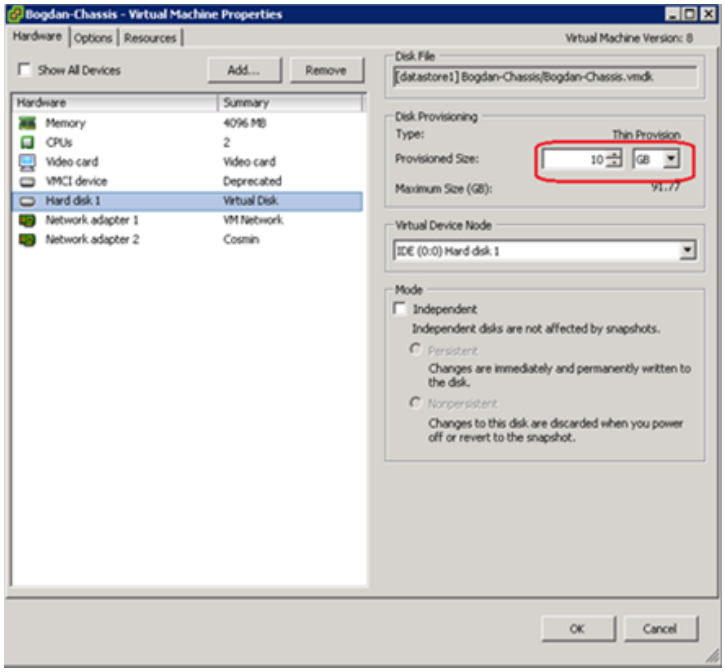
Microsoft Azure

AWS

VMware ESXi

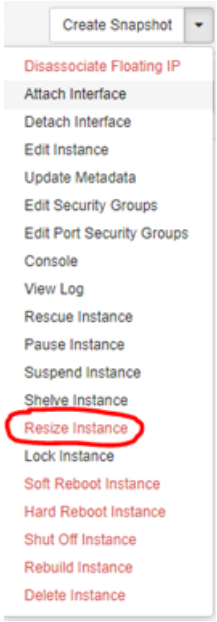
After the vController is offline, the disk size can be modified from settings. You can set the desired value of the Hard Disk from the **Hardware** menu. The new value should always be greater than the one that is currently set.



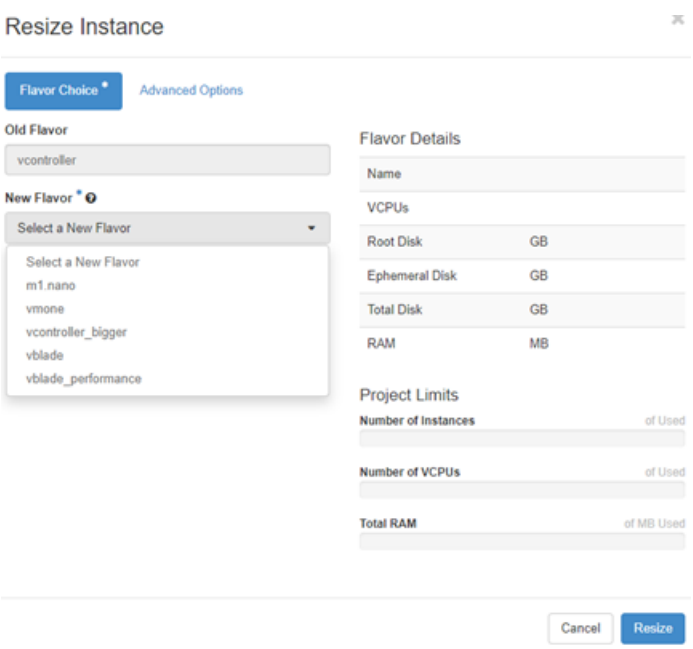


OpenStack

1. Select the arrow next to the **Create Snapshot** button and select **Resize Instance**.

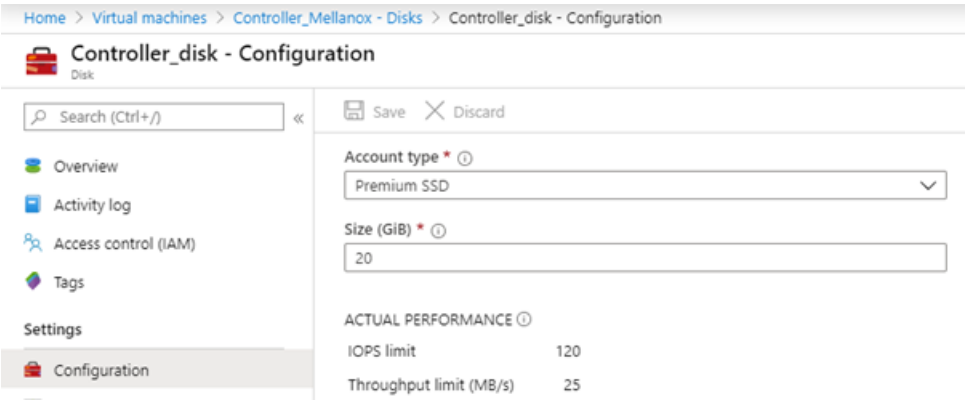


2. When the **Resize Instance** window is displayed, select a **Flavor** which provides a larger size for the vController. Then select **Resize**.



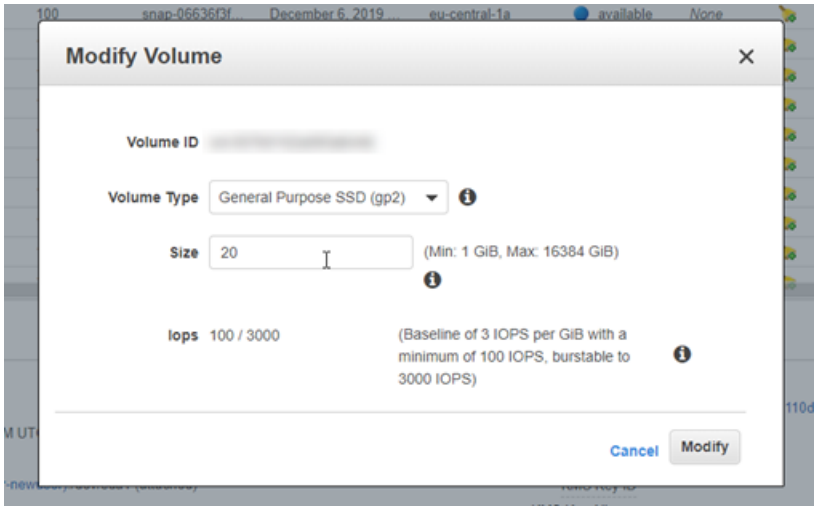
Microsoft Azure

The vController's disk size can be modified by accessing **Virtual Machines** > **Controller_Name** > **Disks** > **Select the disk** > **Configuration** and setting the size.



AWS

1. Choose **EC2** from the **services** list.
2. Select **Volumes** under the **ELASTIC BLOCK STORE** menu (on the left).
3. Choose the volume that you want to resize and then select **Modify Volume** from the volume's context menu.
4. Change the **Size** setting as required and then select **Modify**.


Modify Volume
Volume ID
Volume Type
General Purpose SSD (gp2)
Size
20
(Min: 1 GiB, Max: 16384 GiB)
Iops
100 / 3000
(Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
Cancel
Modify

CHAPTER 10 Cloud-init

Cloud-init is a tool that handles early initialization of Virtual Machine (VM) instances, by performing a set of configuration tasks.

BPS VE is integrated with the cloud-init package. When BPS VE Virtual machines start for the first time, cloud-init reads user input from the attached config drive and performs a set of configuration tasks. For example, during initialization, cloud-init can update the Virtual Blades' new user, setting the static IP, netmask, gateway, DNS server, etc.

Most of the major distributions are delivered as cloud-enabled images for use in cloud based environments. These images being smaller and supporting automatic configuration (using cloud-init) during startup, make it very attractive for deploying outside a cloud environment. But the deployment of image files outside a cloud based environment consumes limitless time. The recommended solution is to utilize the cloud-init tool to read the configuration information from the attached configuration drive.

VMware ESXi

Create Configuration Drive

The simplest method for creating a configuration drive is to use cloud-init's no cloud data source; by creating an ISO file system.

The ISO file system is created by the following files:

- meta-data
- user-data (optional)

meta-data file:

The meta-data file is effectively a YAML version as represented below.

```
instance-id: my-instance-id
```

The instance-id key is required. You can also include SSH public keys in this file as shown below.

```
instance-id: my-instance-id
public-keys:
- ssh-rsa AAAAB3NzaC1...
```

user-data file:

The user-data can be any of the various formats supported by cloud-init. For example, it could simply be a shell script.

```
#!/bin/sh
yum -y install some-critical-package
```

Or it could be a cloud-config YAML document:

```
#cloud-config write-files:
- path: /etc/profile.d/gitaliases.sh content:
alias gc="git commit"
alias gcv="git commit --no-verify"
runcmd:
- setenforce 1
```

Configuration Drive

Using the meta-data file and user-data file, you can create the configuration drive as follows:

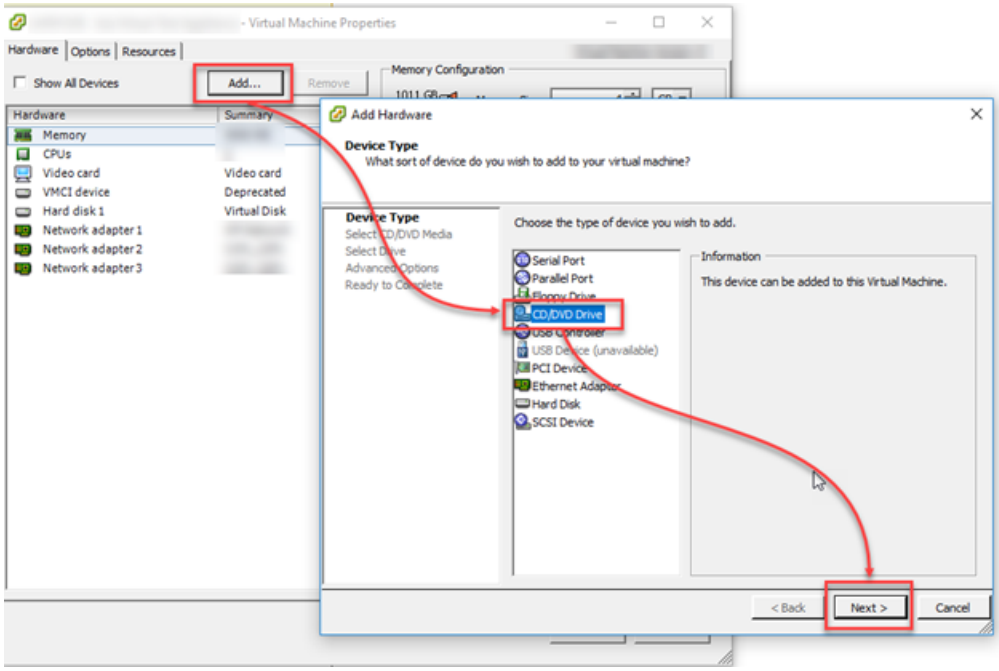
```
genisoimage -o Cloud_Init_Configuration.iso -V cidata -r -J meta-data user-data
```

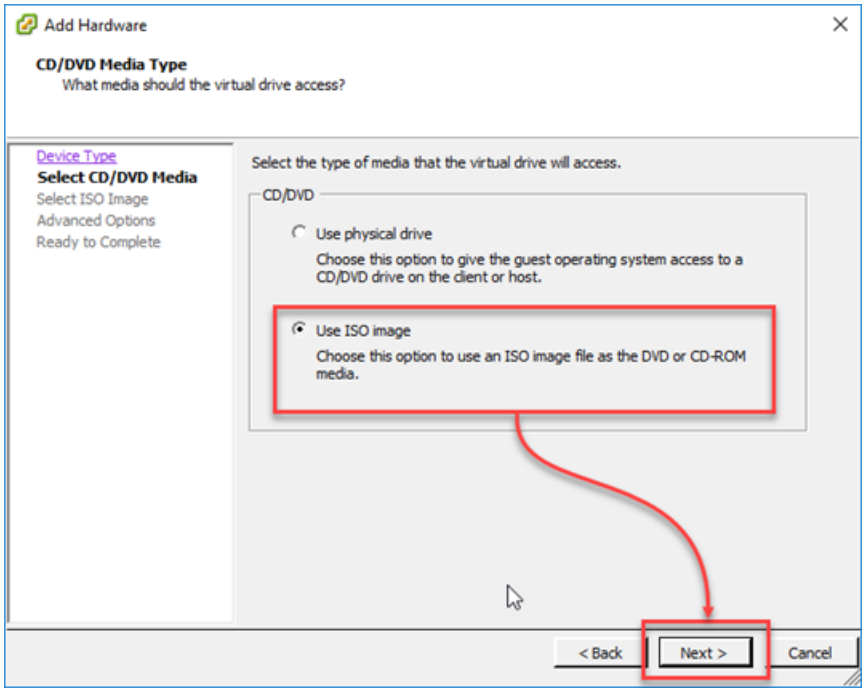
Deploying an Instance

To deploy an instance, copy the ISO Configuration Drive to the ESXi datastore.

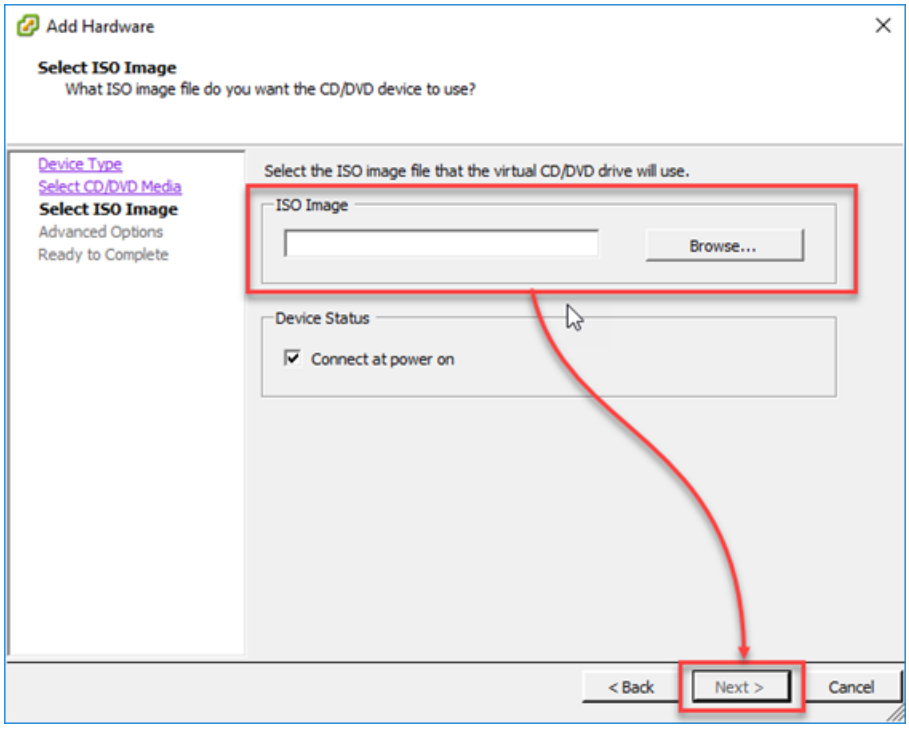
Then attach the generated ISO file (described in the previous step) to the VM and start it.

For VMware ESXi 6.0:

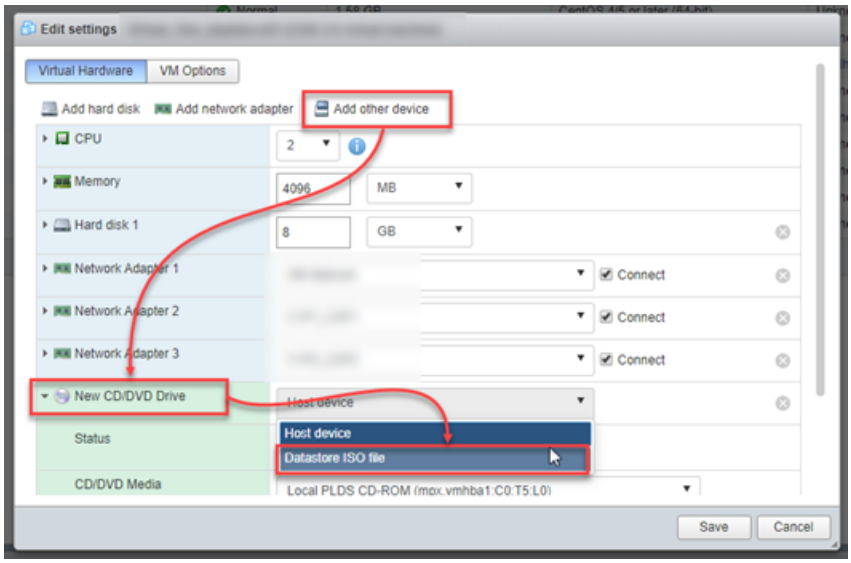




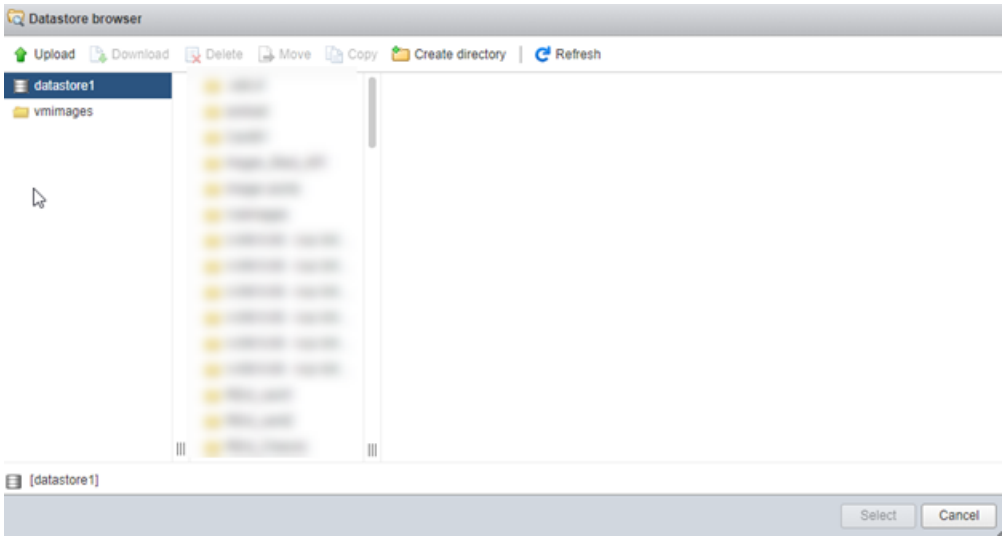
Select your ISO image from the datastore.



For VMware ESXi >= 6.5:



Select your ISO from the datastore:



Click **Select** and then you can start your Instance.

QEMU / KVM

Create Configuration Drive

The simplest method for creating configuration drive is to use cloud-init's no cloud data source; by creating an ISO file system.

The ISO file system is created by the following files:

- meta-data
- user-data (optional)

meta-data file:

The meta-data file is effectively a YAML version as represented below.

```
instance-id: my-instance-id
```

The instance-id key is required. You can also include SSH public keys in this file as shown below.

```
instance-id: my-instance-id
public-keys:
- ssh-rsa AAAAB3NzaC1...
```

user-data file:

The user-data can be any of the various formats supported by cloud-init. For example, it could simply be a shell script.

```
#!/bin/sh
yum -y install some-critical-package
```

Or it could be a cloud-config YAML document:

```
#cloud-config write-files:
- path: /etc/profile.d/gitaliases.sh content:
alias gc="git commit"
alias gcv="git commit --no-verify"
runcmd:
- setenforce 1
```

Configuration Drive

Using the meta-data file and user-data file, you can create the configuration drive as follows:

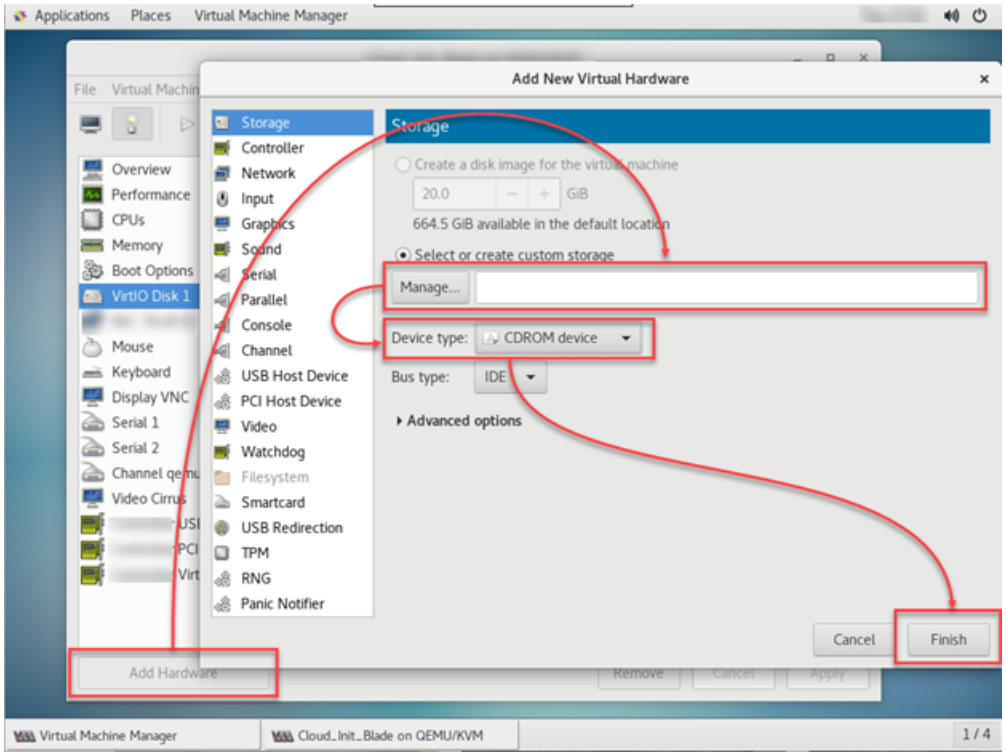
```
genisoimage -o Cloud_Init_Configuration.iso -V cidata -r -J meta-data user-data
```

Deploying an Instance

Copy the Cloud_Init_Configuration.iso ISO file into the same location as your qcow2 image on the QEMU / KVM hypervisor.

When deploying the virtual machine, you have two options:

- GUI



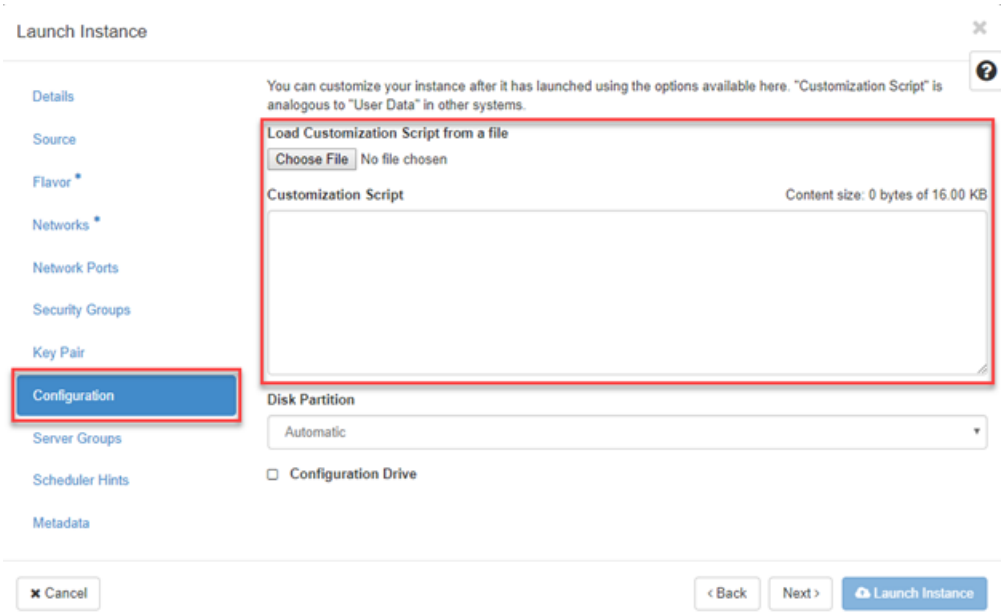
- LIBVIRT / VIRT-MANAGER

```
virt-install --name Controller --ram 8192 --vcpus 8 --network
bridge=br0,model=virt
```

OpenStack

Deployment through the OpenStack User Interface

A portion of configuring Cloud Init from the OpenStack GUI will require using the **Configuration** page of the **Launch Instance** window.



Deployment through OpenStack CLI

User data is a blob of data that the user can specify when they launch an instance. The instance can access this data through the metadata service or config drive. It is commonly used to pass a shell script that the instance runs on boot.

For example, one application that uses user data is the cloud-init system, which is an open-source package from Ubuntu that is available on various Linux distributions and which handles early initialization of a cloud instance.

You can place user data in a local file and pass it through the --user-data <user-data-file> parameter at instance creation.

```
openstack server create --image cloudimage --flavor 1 --user-data mydata.file VM_
INSTANCE
```

Deployment Through Heat Templates

Here is an example of a non-cloud init Heat Template:

```yaml
heat_template_version: 2013-05-23
description: >
  Version: 9.00
    This is the Heat Template used for BPS VE deployment in OpenStack-based environments. It can be used with BPS VE product with version 9.00 or newer.
    The template will create 1 Virtual Networks (named "Test Network") and for Management it will be using an existing user given network name.
    The template will create 1 Virtual Machine  (named "Virtual-Blade-1").
    The Ixia_BreakingPoint_Virtual_Blade_9.00_Heat_Template_Variables.yaml environment file is used for hardcoding the parameters for the resources defined in this Heat Template.
parameters:
    Management_Network:
      type:        string
      label:       BPS Management Network - Name
      description: The name of the MANAGEMENT NETWORK that is already configured in OpenStack
    Test_Network:
      type:        string
      label:       BPS Test Network - Name
      description: The name of the TEST NETWORK to be created
    Test_Sub_Network:
      type:        string
      label:       BPS Test Network - CIDR
      description: The IP Address / Subnet Mask of the TEST NETWORK to be created
    Virtual_Blade_Name:
      type:        string
      label:       BPS Virtual Blade - Instance Name
      description: The name of the BPS Virtual Blade to be created
    Virtual_Blade_Image_Name:
      type:        string
      label:       BPS Virtual Blade - Image Name
      description: The QCOW2 image to be used for the BPS Virtual Blade
    Virtual_Blade_Flavor:
      type:        string
      label:       BPS Virtual Blade - Flavor
      description: The resources required for each BPS VIRTUAL BLADE are 4 vCPU, 8 GB RAM, 14 GB HDD
resources:
  BPS-Test-Network:
    type:                OS::Neutron::Net
    properties:
      name:              {get_param:    Test_Network}
  BPS-Test-Subnet:
     type:               OS::Neutron::Subnet
     properties:
       network_id:       {get_resource: BPS-Test-Network}
       cidr:             {get_param:    Test_Sub_Network}
       gateway_ip:       null
       ip_version:       4
       enable_dhcp:      false
  Test-Port-1:
       type:             OS::Neutron::Port
       properties:
         network_id:     {get_resource: BPS-Test-Network}
         fixed_ips:
           - subnet_id:  {get_resource: BPS-Test-Subnet}
  Test-Port-2:
       type:             OS::Neutron::Port
       properties:
         network_id:     {get_resource: BPS-Test-Network}
         fixed_ips:
           - subnet_id:  {get_resource: BPS-Test-Subnet}
  Virtual-Blade-1:
    type:                OS::Nova::Server
    properties:
      name:              {list_join:    ['-', [{get_param: Virtual_Blade_Name}, '1']]}
      image:             {get_param:    Virtual_Blade_Image_Name}
      flavor:            {get_param:    Virtual_Blade_Flavor}
      networks:
        - network:       {get_param:    Management_Network }
        - port:          {get_resource: Test-Port-1}
        - port:          {get_resource: Test-Port-2}
```

The part which involves cloud-init is placed at the end of the Heat Template. In a scenario where you edit a file on the disk, the result would look like the example below.

```yaml
heat_template_version: 2013-05-23
description: >
  Version: 9.00
    This is the Heat Template used for BPS VE deployment in OpenStack-based environments. It can be used with BPS VE product with version 9.00 or newer.
    The template will create 1 Virtual Networks (named "Test Network") and for Management it will be using an existing user given network name.
    The template will create 1 Virtual Machine  (named "Virtual-Blade-1").
    The Ixia_BreakingPoint_Virtual_Blade_9.00_Heat_Template_Variables.yaml environment file is used for hardcoding the parameters for the resources defined in this Heat Template.
parameters:
    Management_Network:
      type:        string
      label:       BPS Management Network - Name
      description: The name of the MANAGEMENT NETWORK that is already configured in OpenStack
    Test_Network:
      type:        string
      label:       BPS Test Network - Name
      description: The name of the TEST NETWORK to be created
    Test_Sub_Network:
      type:        string
      label:       BPS Test Network - CIDR
      description: The IP Address / Subnet Mask of the TEST NETWORK to be created
    Virtual_Blade_Name:
      type:        string
      label:       BPS Virtual Blade - Instance Name
      description: The name of the BPS Virtual Blade to be created
    Virtual_Blade_Image_Name:
      type:        string
      label:       BPS Virtual Blade - Image Name
      description: The QCOW2 image to be used for the BPS Virtual Blade
    Virtual_Blade_Flavor:
      type:        string
      label:       BPS Virtual Blade - Flavor
      description: The resources required for each BPS VIRTUAL BLADE are 4 vCPU, 8 GB RAM, 14 GB HDD
resources:
  BPS-Test-Network:
    type:               OS::Neutron::Net
    properties:
      name:             {get_param:    Test_Network}
  BPS-Test-Subnet:
     type:              OS::Neutron::Subnet
     properties:
       network_id:      {get_resource: BPS-Test-Network}
       cidr:            {get_param:    Test_Sub_Network}
       gateway_ip:      null
       ip_version:      4
       enable_dhcp:     false
  Test-Port-1:
       type:            OS::Neutron::Port
       properties:
         network_id:    {get_resource: BPS-Test-Network}
         fixed_ips:
           - subnet_id: {get_resource: BPS-Test-Subnet}
  Test-Port-2:
       type:            OS::Neutron::Port
       properties:
         network_id:    {get_resource: BPS-Test-Network}
         fixed_ips:
           - subnet_id: {get_resource: BPS-Test-Subnet}
  boot_config:
    type: OS::Heat::CloudConfig
    properties:
      cloud_config:
        users:
       - name: demo
         passwd: ixia123
         lock_passwd: false
  Virtual-Blade-1:
    type:               OS::Nova::Server
    properties:
      name:             {list_join:    ['-', [{get_param: Virtual_Blade_Name}, '1']]}
      image:            {get_param:    Virtual_Blade_Image_Name}
      flavor:           {get_param:    Virtual_Blade_Flavor}
      user_data_format: SOFTWARE_CONFIG
      user_data:        {get_resource: boot_config}
      networks:
       - network:       {get_param:    Management_Network }
       - port:          {get_resource: Test-Port-1}
       - port:          {get_resource: Test-Port-2}
```

Amazon AWS

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances using the command line tools), or as base64-encoded text (for API calls).

User Data and Shell Scripts

If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch. Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.

User data shell scripts must start with the #! characters and the path to the interpreter you want to read the script (commonly **/bin/bash**). For a great introduction on shell scripting, see the BASH Programming HOW-TO at the Linux Documentation Project (tldp.org).

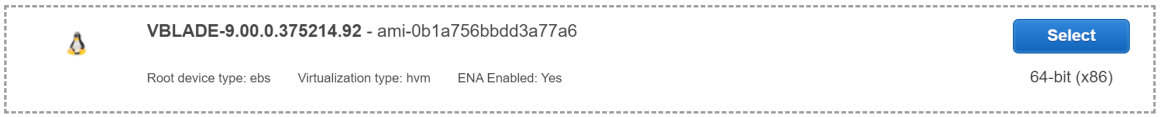
Scripts entered as user data are executed as the **root** user, so do not use the sudo command in the script. Remember that any files you create will be owned by **root**; if you need non-root users to have file access, you should modify the permissions accordingly in the script. Also, because the script is not run interactively, you cannot include commands that require user feedback (such as **yum update** without the `-y` flag).

The cloud-init output log file (`/var/log/cloud-init-output.log`) captures console output so it is easy to debug your scripts following a launch if the instance does not behave the way you intended.

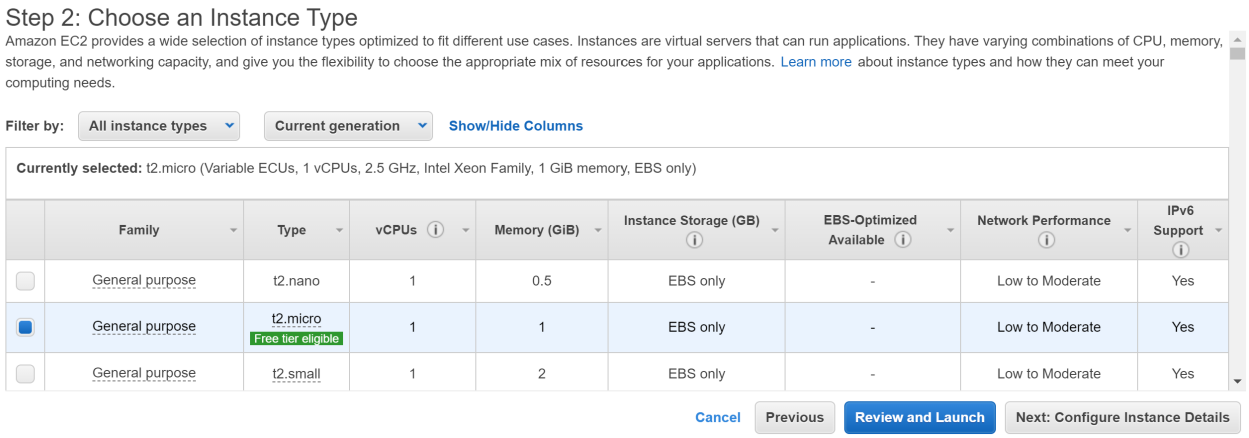
When a user data script is processed, it is copied to and executed from `/var/lib/cloud/instances/instance-id/`. The script is not deleted after it is run. Be sure to delete the user data scripts from `/var/lib/cloud/instances/instance-id/` before you create an AMI from the instance. Otherwise, the script will exist in this directory on any instance launched from the AMI.

UI Guidance

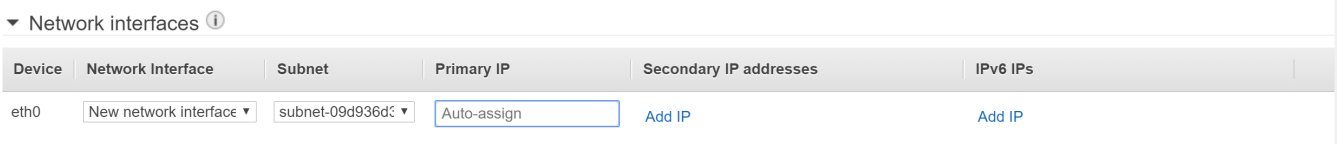
1. Select your AMI ID.



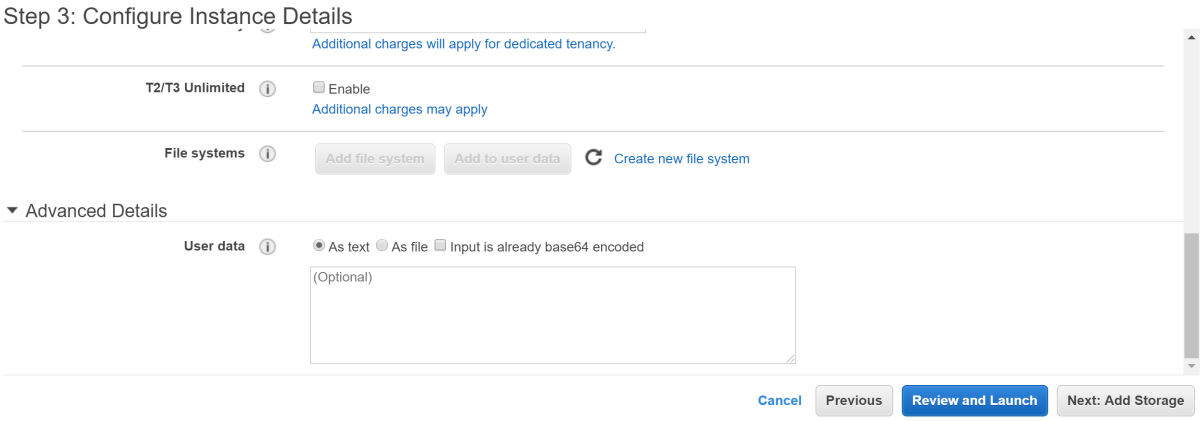
2. Select your instance size and then select **Next**.



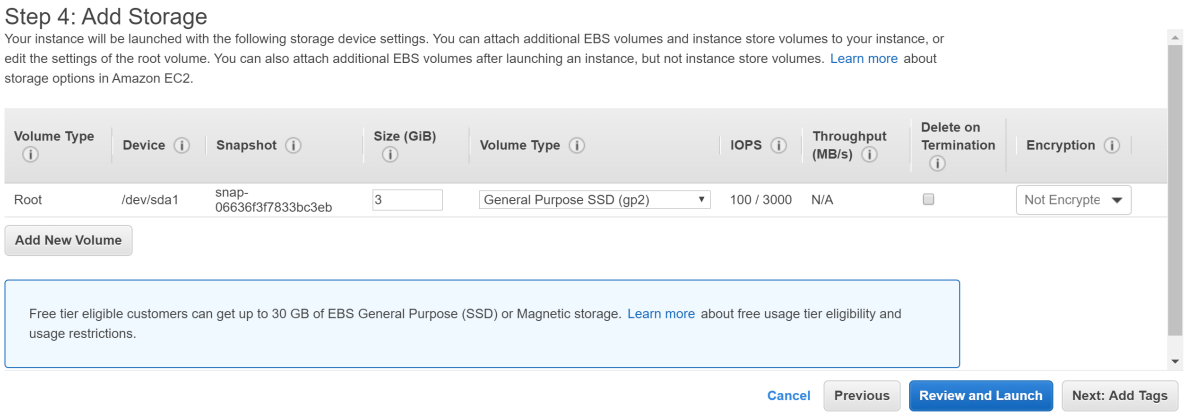
3. In the **Configure Instance Details** > **Network Interfaces** section, add your network interfaces.



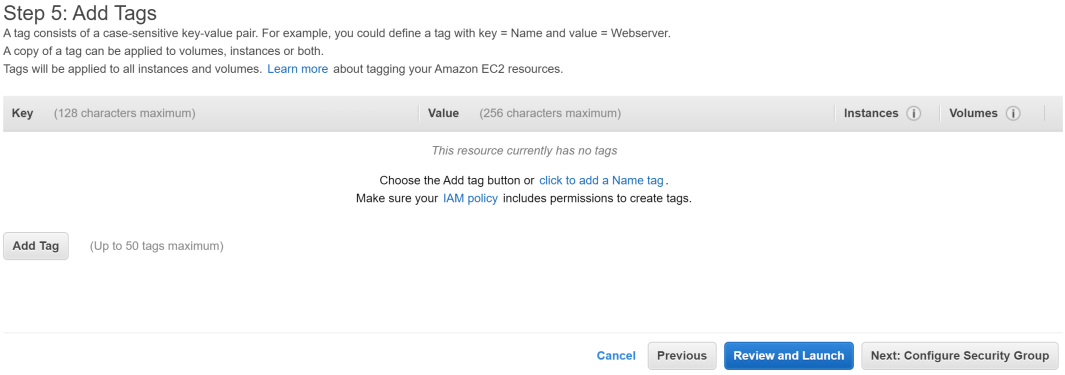
4. In the **Configure Instance Details** > **Advanced Details** section, add your user data and then select **Next**.



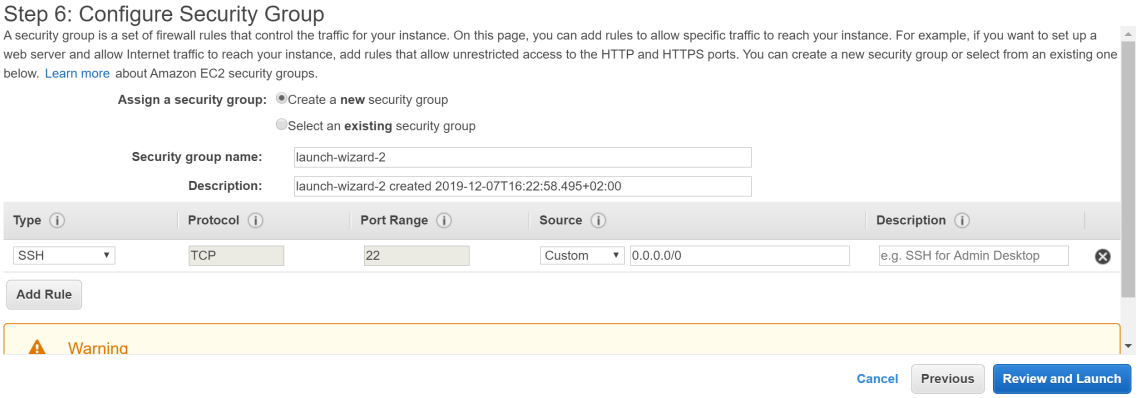
5. Select your storage.



6. Add your tags.



7. Configure your Security Group.



8. Launch your instance.

This page intentionally left blank.

CHAPTER 11 Mellanox Support on BPS VE

This chapter describes Mellanox support on BPS VE.

Ixia currently supports two guest drivers from the Mellanox families for Ethernet adapters:

- mlx4 for ConnectX-3 and ConnectX-3 Pro boards
- mlx5 for ConnectX-4 and ConnectX-4 LX boards

Ixia supports Mellanox on the following platforms:

- VMware ESXi,
- KVM CentOs/ Ubuntu
- OpenStack over CentOs/ Ubuntu
- Microsoft Azure

For more details about speed, driver versions on guest and host, types of supported cards, etc., please see, Supported Platforms on page 176.

Mellanox Driver Installation and Configuration for VMware ESXi

Please note the following prerequisites:

- SR-IOV is enabled in BIOS
- intel_iommu=on and iommu=pt are added to /boot/grub/grub.conf

Please note the following limitations:

- On ESXi 6.0 there can be only 6 PCI devices configured in SR-IOV mode
- There is a known limitation in vSphere ESXi regarding VLANs which causes VLAN tests to fail. Limitations exist for ixgbe adapters as well.
- Before running a DDOS test, a vBlade's compatibility needs to be upgraded to the latest version (**Actions** -> **Upgrade VM Compatibility**) and also note that SR-IOV cards need to be added as network adapters, not PCI devices (this is only available under latest VM 6.7 compatibility mode)

For mlx4 SR-IOV and mlx4 PCI-PassThrough

Note: Connect X3 is not supported in SR-IOV mode on VMware ESXi 6.7

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download the Mellanox dependencies package driver.

```
cd /var/log/vmware
wget http://www.mellanox.com/downloads/Software/MLNX-OFED-ESX-2.4.0.0-10EM-
600.0.0.2494585.zip
```

3. Install the Mellanox dependencies package driver:

```
esxcli software vib install -d MLNX-OFED-ESX-2.4.0.0-10EM-600.0.0.2494585.zip

Installation Result:
Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
Reboot Required: true
VIBs Installed: MEL_bootbank_net-mlx-compat_2.4.0.0-1OEM.600.0.0.2494585, MEL_
bootbank_net-mlx4-core_2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-mlx4-en_
2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-mlx4-ib_2.4.0.0-
1OEM.600.0.0.2494585
VIBs Removed:
VIBs Skipped: MEL_bootbank_net-ib-core_2.4.0.0-1OEM.600.0.0.2494585, MEL_
bootbank_net-ib-ipoib_2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-ib-mad_
2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-ib-sa_2.4.0.0-1OEM.600.0.0.2494585
```

4. Reboot the VM and verify that the driver is loaded.

```
esxcli system module list | grep mlx4

mlx4_core true true
mlx4_en true true
mlx4_ib true true
```

For mlx5 SR-IOV

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download the Mellanox dependencies package driver.

```
cd /var/log/vmware
wget http://www.mellanox.com/downloads/Software/MLNX-OFED-ESX-2.4.0.0-10EM-
600.0.0.2494585.zip
```

3. Install the Mellanox dependencies package driver.

```
esxcli software vib install -d MLNX-OFED-ESX-2.4.0.0-10EM-600.0.0.2494585.zip

Installation Result
Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
Reboot Required: true
```

```
VIBs Installed: MEL_bootbank_net-mlx-compat_2.4.0.0-1OEM.600.0.0.2494585, MEL_
bootbank_net-mlx4-core_2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-mlx4-en_
2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-mlx4-ib_2.4.0.0-
1OEM.600.0.0.2494585
VIBs Removed:
VIBs Skipped: MEL_bootbank_net-ib-core_2.4.0.0-1OEM.600.0.0.2494585, MEL_
bootbank_net-ib-ipoib_2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-ib-mad_
2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-ib-sa_2.4.0.0-1OEM.600.0.0.2494585
```

4. Reboot.
5. Download (locally to your workstation) the Mellanox ConnectX-4(mlx5) from the VMware site.

```
MLNX-NATIVE-ESX-ConnectX-4-5_4.15.10.3-10EM-600.0.0.2768847-6159323.zip
```

6. Extract the zip package and look for the offline bundle.

```
MLNX-NATIVE-ESX-ConnectX-4-5_4.15.10.3-10EM-600.0.0.2768847-offline_bundle-
6159323.zip
```

7. Copy the offline bundle to the following directory of the ESXi server: /var/log/vmware.
8. Install the driver and verify that it is loaded.

```
cd /var/log/vmware
esxcli software vib install -d MLNX-NATIVE-ESX-ConnectX-4-5_4.15.10.3-10EM-
600.0.0.2768847
-offline_bundle-6159323.zip
esxcli system module list | grep mlx5
nmlx5_core true true
```

9. Download and install the MFT tool from Mellanox site.

```
cd /var/log/vmware
wget http://www.mellanox.com/downloads/MFT/vmware_6.0_native/mft-4.7.0.42-10EM-
600.0.0.2768847.x86_64.vib
esxcli software vib install -v mft-4.7.0.42-10EM-600.0.0.2768847.x86_64.vib

Installation Result
Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
Reboot Required: true
VIBs Installed: Mellanox_bootbank_mft_4.7.0.42-0
VIBs Removed:
VIBs Skipped:
```

10. Download and install the NMST tool from the Mellanox site.

```
cd /var/log/vmware
wget http://www.mellanox.com/downloads/MFT/vmware_6.0_native/nmst-4.8.0.26-
1OEM.600.0.0.2768847.x86_64.vib
esxcli software vib install -v nmst-4.8.0.26-1OEM.600.0.0.2768847.x86_64.vib

Installation Result
```

```
Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
Reboot Required: true
VIBs Installed: ...
VIBs Removed:
VIBs Skipped:
```

11. Reboot.
12. List the MFT devices.

```
/opt/mellanox/bin/mst status
MST devices:
------------
mt4117_pciconf0
mt4115_pciconf1
```

13. Query the selected NIC for the actual configuration.

```
/opt/mellanox/bin/mlxconfig -d mt4117_pciconf0 q
```

14. Enable SRIOV and activate the desired number of VFs.

```
/opt/mellanox/bin/mlxconfig -d mt4117_pciconf0 set SRIOV_EN=1 NUM_OF_VFS=8
Device #1:
```

```
Device type: ConnectX4LX
PCI device: mt4117_pciconf0
Configurations: Next Boot New
SRIOV_EN True(0) True(1)
NUM_OF_VFS 0 8
Apply new Configuration? ? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.
```

15. Enable the VF on the Mellanox driver.

```
esxcfg-module -s "max_vfs=8,8" nmlx5_core
```

16. Reboot and verify that the new VF's are created.

```
lspci | grep Mellanox
```

For mlx5 PCI-PassThrough

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download Mellanox dependencies package driver.

```
cd /var/log/vmware
wget http://www.mellanox.com/downloads/Software/MLNX-OFED-ESX-2.4.0.0-10EM-
600.0.0.2494585.zip
```

3. Install Mellanox dependencies package driver.

esxcli software vib install -d MLNX-OFED-ESX-2.4.0.0-10EM-600.0.0.2494585.zip

```
Installation Result

Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
Reboot Required: true
VIBs Installed: MEL_bootbank_net-mlx-compat_2.4.0.0-1OEM.600.0.0.2494585, MEL_
bootbank_net-mlx4-core_2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-mlx4-en_
2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-mlx4-ib_2.4.0.0-
1OEM.600.0.0.2494585
VIBs Removed:
VIBs Skipped: MEL_bootbank_net-ib-core_2.4.0.0-1OEM.600.0.0.2494585, MEL_
bootbank_net-ib-ipoib_2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-ib-mad_
2.4.0.0-1OEM.600.0.0.2494585, MEL_bootbank_net-ib-sa_2.4.0.0-1OEM.600.0.0.2494585
```

4. Reboot.
5. Download (locally to your workstation) the Mellanox ConnectX-4(mlx5) from the VMware site.

```
MLNX-NATIVE-ESX-ConnectX-4-5_4.15.10.3-10EM-600.0.0.2768847-6159323.zip
```

6. Extract the zip package and look for the offline bundle.

```
MLNX-NATIVE-ESX-ConnectX-4-5_4.15.10.3-10EM-600.0.0.2768847-offline_bundle-
6159323.zip
```

7. Copy the offline bundle to the following directory of the ESXi server: /var/log/vmware.
8. Install the driver and verify that it is loaded.

```
cd /var/log/vmware
esxcli software vib install -d MLNX-NATIVE-ESX-ConnectX-4-5_4.15.10.3-10EM-
600.0.0.2768847
-offline_bundle-6159323.zip
esxcli system module list | grep mlx5
nmlx5_core true true
```

Mellanox Driver Installation and Configuration for KVM

Please note the following prerequisites.

KVM installed on hypervisors:

- For CentOS:
  - yum install kvm
  - yum install virt-manager libvirt libvirt-python python-virtinst
- For Ubuntu:
  - apt-get install kvm libvirt-bin virt-manager uml-utilities -y
  - SRIOV is enabled in BIOS
  - intel_iommu=on and iommu=pt are added to /boot/grub/grub.conf

Please note the following limitations:

Only one port will be available for mlx4 PCI-Passthrough.

For mlx4 SR-IOV

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download OFED mlx4 EN driver from Mellanox site and Accept the Eula.

For CentOS:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-rhel7.4-x86_64.tgz

For Ubuntu:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-ubuntu16.04-x86_64.tgz

3. Untar the mlx5 driverarchive and then install it.

```
tar -xvf downloaded_driver.tgz
./mlnxofedinstall
```

4. Insert the new driver modules to activate the driver and then verify its version.

```
/etc/init.d/opensmd restart
ethtool -i NICname
```

5. Download the MFT tool from the Mellanox site and then untar, install and start the MFT tool.

```
wget http://www.mellanox.com/downloads/MFT/mft-4.7.0-42-x86_64-rpm.tgz
tar -xvf mft-4.7.0-42-x86_64-rpm.tgz
```

For CentOS:

```
./install.sh
-I- In order to start mst, please run "mst start".
```

If a message similar to the following is received, "...There are missing packages that are required for installation of MFT..", please install the missing packages indicated by the installer. For example, "yum install gcc rpm-build kernel-devel-3.10.0-693.5.2.el7.x86_64".

For Ubuntu:

```
./install.sh
-I- In order to start mst, please run "mst start".

mst start
Starting MST (Mellanox Software Tools) driver set
Loading MST PCI module - Success
Loading MST PCI configuration module - Success
Create devices
Unloading MST PCI module (unused) - Success
```

6. Verify the MST device and the PCI number for the NIC that you wish to activate for SR-IOV.

```
mst status
MST modules:
------------
MST PCI module loaded
MST PCI configuration module loaded

MST devices:
------------
/dev/mst/mt4103_pciconf0 - PCI configuration cycles access.
domain:bus:dev.fn=0000:04:00.0 addr.reg=88 data.reg=92
Chip revision is: 00
/dev/mst/mt4103_pci_cr0 - PCI direct access.
domain:bus:dev.fn=0000:04:00.0 bar=0x92400000 size=0x100000
Chip revision is: 00
```

7. Query the selected NIC for actual configuration.

```
mlxconfig -d /dev/mst/mt4103_pciconf0 q
```

8. Enable SR-IOV and activate the desired number of VFs.

```
mlxconfig -d /dev/mst/mt4103_pciconf0 set SRIOV_EN=1 NUM_OF_VFS=16
```

9. Activate the number of VFs that you will need.

Create (or edit) the /etc/modprobe.d/mlx4_core.conf by using the following command:

```
options mlx4_core num_vfs=16,16 port_type_array=1,1 probe_vf=0
```

10. Reboot.
11. Verify that the new VFs have been created.

```
lspci | grep Mellanox
```

```
04:00.0 Ethernet controller: Mellanox Technologies MT27520 Family [ConnectX-3
Pro]
04:00.1 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:00.2 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:00.3 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:00.4 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:00.5 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:00.6 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:00.7 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.0 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.1 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.2 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.3 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.4 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.5 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.6 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:01.7 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
04:02.0 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
[ConnectX-3/ConnectX-3 Pro Virtual Function]
```

For mlx4 PCI-PassThrough

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download the OFED mlx4 EN driver from Mellanox site and accept the Eula.

For CentOS:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-rhel7.4-x86_64.tgz

• For Ubuntu:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-ubuntu16.04-x86_64.tgz

3. Untar the mlx5 driverarchive and then install it:

```
tar -xvf downloaded_driver.tgz
./mlnxofedinstall
```

4. Insert new driver modules to activate the new driver and then verify its version:

```
/etc/init.d/opensmd restart
ethtool -i NICname
```

For mlx5 SR-IOV

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download the OFED mlx4 EN driver from Mellanox site and Accept the Eula:

For CentOS:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-rhel7.4-x86_64.tgz

For Ubuntu:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-ubuntu16.04-x86_64.tgz

3. Untar the mlx5 driver archive and then install it.

```
tar -xvf downloaded_driver.tgz
./mlnxofedinstall
```

4. Insert new driver modules to activate the new driver and then verify its version:

```
/etc/init.d/opensmd restart
ethtool -i NICname
```

5. Download MFT tool from Mellanox site, untar, install and start it:

```
wget http://www.mellanox.com/downloads/MFT/mft-4.7.0-42-x86_64-rpm.tgz
tar -xvf mft-4.7.0-42-x86_64-rpm.tgz
```

CentOs Install:

```
./install.sh
-I- In order to start mst, please run "mst start".
```

If a message similar to the following is received, "...There are missing packages that are required for installation of MFT..", please install the missing packages indicated by the installer. For example, "yum install gcc rpm-build kernel-devel-3.10.0-693.5.2.el7.x86_64".

Ubuntu Install:

```
./install.sh
-I- In order to start mst, please run "mst start".

mst start
Starting MST (Mellanox Software Tools) driver set
Loading MST PCI module - Success
Loading MST PCI configuration module - Success
Create devices
Unloading MST PCI module (unused) - Success
```

6. Verify the MST device and PCI numberfor the NIC that you wish to activate for SR-IOV:

```
mst status
MST modules:
------------
MST PCI module is not loaded
MST PCI configuration module loaded

MST devices:
------------
/dev/mst/mt4115_pciconf0 - PCI configuration cycles access.
domain:bus:dev.fn=0000:06:00.0 addr.reg=88 data.reg=92
Chip revision is: 00
/dev/mst/mt4117_pciconf0 - PCI configuration cycles access.
domain:bus:dev.fn=0000:05:00.0 addr.reg=88 data.reg=92
Chip revision is: 00
```

7. Query the selected NIC for the actual configuration.

```
mlxconfig -d /dev/mst/mt4117_pciconf0 q
```

8. Enable SR-IOV and activate the desired number of VFs.

```
mlxconfig -d /dev/mst/mt4117_pciconf0 set SRIOV_EN=1 NUM_OF_VFS=16
```

9. Reboot to enable the configuration.
10. Verify the total number of VFs.

```
cat /sys/class/net/enp5s0f1/device/sriov_totalvfs
16
```

Note: This parameter should be aligned with the number of configured VF's indicated in the command shown above: "mlxconfig -d /dev/mst/mt4117_pciconf0 set SRIOV_EN=1 NUM_OF_VFS=16". If you do not see this parameter, it means that the intel_iommu=on was not added to the grub file, as mentioned in the prerequisites.

11. Check the number of VFs per port and then activate the number of VFs that are needed.

```
cat /sys/class/net/enp5s0f0/device/sriov_numvfs
0
cat /sys/class/net/enp5s0f1/device/sriov_numvfs
0
echo 8 > /sys/class/net/enp5s0f0/device/sriov_numvfs
echo 8 > /sys/class/net/enp5s0f1/device/sriov_numvfs
```

12. Verify that the activation is successful and that the new VFs have been created:

```
cat /sys/class/net/enp5s0f0/device/sriov_numvfs
8
cat /sys/class/net/enp5s0f1/device/sriov_numvfs
8
lspci | grep Mellanox
05:00.0 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx]
05:00.1 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx]
05:00.2 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:00.3 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:00.4 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:00.5 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:00.6 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:00.7 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:01.0 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:01.1 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:02.2 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:02.3 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:02.4 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:02.5 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:02.6 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:02.7 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:03.0 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
05:03.1 Ethernet controller: Mellanox Technologies MT27710 Family [ConnectX-4 Lx
Virtual Function]
```

For mlx5 PCI-PassThrough

1. Verify that the cards are listed.

```
lspci | grep Mellanox
```

2. Download the OFED mlx4 EN driver from Mellanox site and accept the Eula.

For CentOS:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-rhel7.4-x86_64.tgz

For Ubuntu:

http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-4.1-1.0.2.0&mname=MLNX_OFED_LINUX-4.1-1.0.2.0-ubuntu16.04-x86_64.tgz

3. Untar the mlx5 driver archive and then install it.

```
tar -xvf downloaded_driver.tgz
./mlnxofedinstall
```

4. Insert new driver modules to activate the new driver and then verify its version:

```
/etc/init.d/opensmd restart
ethtool -i NICname
```

Mellanox Driver Installation and Configuration for OpenStack Stein

Follow the steps for Mellanox Driver Installation and Configuration for KVM.

For more information, access:

- https://docs.openstack.org/neutron/stein/admin/config-sriov.html
- https://community.mellanox.com/s/article/howto-configure-sr-iov-for-connectx-4-connectx-5-with-kvm--ethernet-x
- https://community.mellanox.com/s/article/howto-configure-sr-iov-vfs-on-different-connectx-3-ports

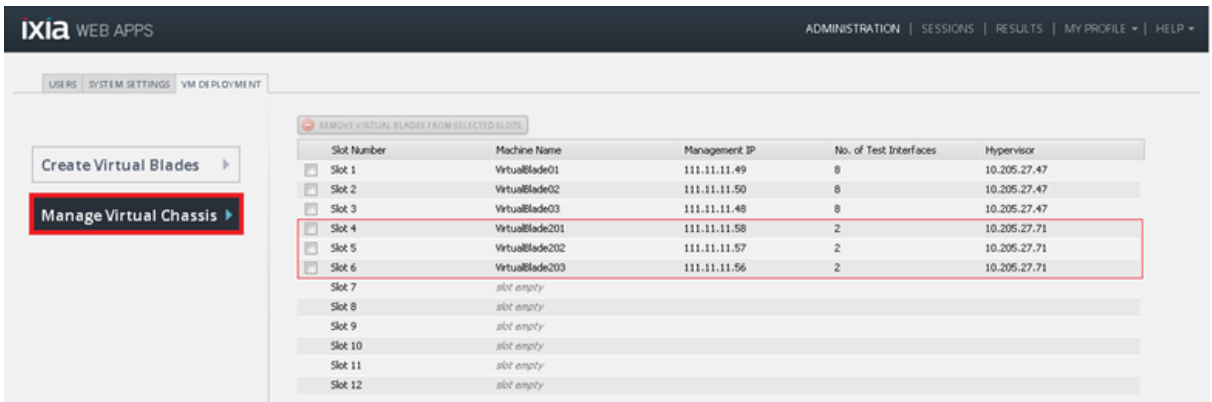
This page intentionally left blank.

CHAPTER 12 Managing vBlades

This section describes the procedures for discovering, deleting and unassigning vBlades.

Discover vBlades

After successfully deploying the vBlades (NP-VM), you can view them in the **Manage Virtual Chassis** tab, which is also known as the Discovery window and BPS Virtual Chassis window.



Virtual Chassis Field Descriptions

Field	Description
Slot Number	Indicates the slot number of the vBlades in a virtual chassis, which ranges from 1 to 12. A system controller can control a maximum of 12 vBlades.
Machine Name	The name of the virtual load module as shown in the image above.
Management IP	The IP of the virtual machine, through which you can manage the vBlades.
No. of Test Interfaces	The number of vPorts on the vBlades.
Hypervisor	The IP of the hypervisor where VMs are deployed.

vBlade Deletion and Assignment Rules

Note the differences between vBlades that are manually deployed and vBlades that are deployed automatically (using the BPS VE UI):

- Deletion will not be possible for vBlades that are assigned manually. The **Delete** check box on the **Manage Virtual Chassis** tab will not be visible for manually deployed vBlades.
- In the **Manage Virtual Chassis** table, the **Machine Name** and **Hypervisor** fields will indicate "unavailable" because the user is not required to provide this information when vBlades are manually deployed.
- All vBlades can be unassigned, irrespective of the way they were deployed.
  - Note that unassignment will only break the connection between the vController and the vBlade.
  - Unassigned vBlades can be assigned and then managed by other vController.

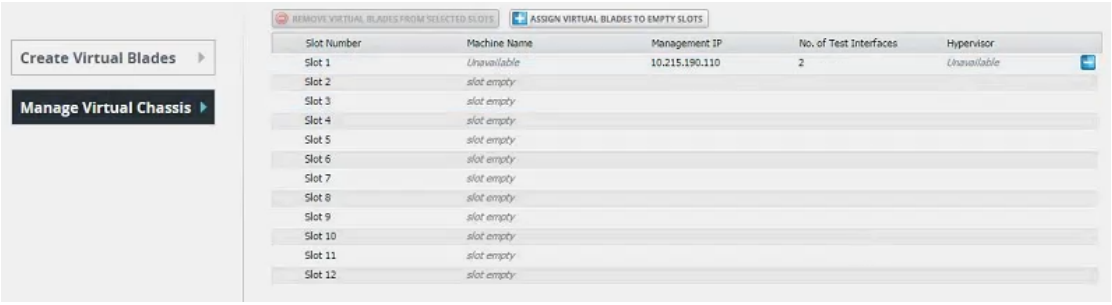
Unassign/Assign a vBlade

Note: To ensure proper vBlade operation, Ixia recommends that vBlades are in the powered ON state before they are unassigned.

To assign or unassign a vBlade:

1. Select **Manage Virtual Chassis**.
2. On the **Assign Virtual Blades To Empty Slots** tab. Select the plus (assign) or minus (unassign) icon that is displayed at the far right side of a slot's row (as shown in the image below).

* **Management IP** = The management IP of the vBlade instance



Note: For BPS on AWS - When manually deploying the vBlade instance, you can attach one more network interface to your instance during launch (in addition to the management interface). After you've launched your instance, you can attach more network interfaces using the EC2 console. Please make sure that after you attach more interfaces, you reboot the vBlade instance (using the EC2 console) in order for the changes to take effect.

Note: Unassigning a vBlade will only break the connection between the controller and the vBlade. The vBlade will not be removed or powered off.

Delete a vBlade

To delete a vBlade, perform the following tasks:

1. Select **Manage Virtual Chassis**.
2. Select **Remove Virtual Blades from Selected Slots**.
3. Select the slots you want to delete vBlades from.
4. Select **Apply**.

This page intentionally left blank.

CHAPTER 13 Licensing

The licensing utility helps in the license management of BreakingPoint Systems by allowing for the activation, deactivation and synchronization of licenses.

Ixia's license management utility allows you to:

- Centralize and monitor your software usage.
- Maintain an accurate license inventory.
- Smoothly transfer licenses across different hosts and teams.

Activation Codes for the purchased Ixia products are sent via email message when you purchase a BreakingPoint Virtual Edition license. Enter this Activation Code in then License Utility to active the license.

Licensing can be performed in the following scenarios:

- From the same VM Controller on which the software was installed; in a scenario where the internet is available on the VM Controller.
- From any other computer connected to internet in a scenario where internet is unavailable on the VM Controller. This option pertains to offline registration mode.

The computer (used for performing the licensing process) must be connected to the internet.

Before activating a license, you must have the e-mail message you received from Ixia which has the activation code. The email contains the following:

- Activation Code: A unique number for the license.
- Quantity: The number of licenses.
- Effective Date: The date from which the license can be activated.
- Expiration Date: The date on which the licenses will expire.

Floating Licenses

Ixia provides Subscription and Perpetual floating licenses for BPS VE.

This type of license is stored on a license server and allows a set number of workstations to use product software features. The workstations using this license must be connected to the license server and the server must be up and running. Additional users for the product features are denied once the set number of licenses is completely being used by the current users.

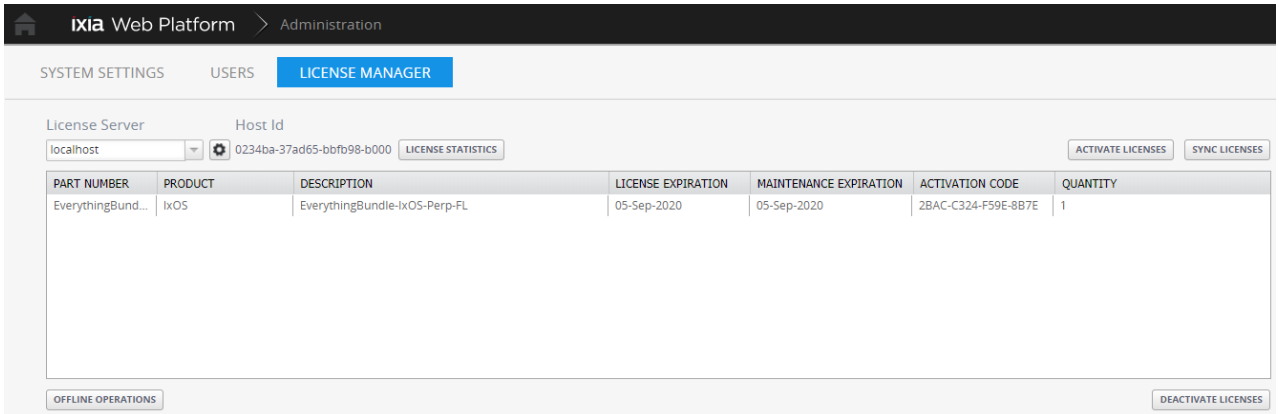
Licensing Utility

The Licensing utility is a one-stop solution which helps to activate, deactivate, sync and view the current licenses that are checked out.

Note: In earlier versions of BPS, BPS VE licensing was accessed in the BPS US at: **BPS Session** > **Control Center** > **Administration** > **Licensing**. BPS VE licensing is now accessed from the Ixia Web Platform as described below.

1. Enter the vController IP address in the URL field of your HTML browser.
2. Log into BreakingPoint (username: admin, password: admin).
3. At the Ixia Web Platform, select the **BreakingPoint** icon.
4. At the BreakingPoint Dashboard, select the **Gear** icon > **Administration** > **System**.
5. Select the **License Manager** tab.

The following figure displays the License Manager user interface.



The following table provides information about the displayed fields and available options:

Field or Button	Description
License Server field	Specify a license server's IP address or hostname. The default value is **localhost**. Localhost points to the computer where BreakingPoint is installed. Select a remote computer's hostname or IP address to view, activate, deactivate and sync licenses on it.
Host ID field	Displays the unique ID of the computer where the License Server is installed.
License Statistics button	Select this link to view details about the type and quantity of licenses that are available.
Activate Licenses button	Select this button to activate a license. Specify the **Activation Code** and **Quantity** of licenses you want to

Field or Button	Description
	activate. The quantity of licenses issued, effective date and expiration date are also mentioned in the email.
Sync Licenses button	If licenses are renewed in the back-end, select **Sync** in utility to reflect the changes.
Part Number field	Displays the product part number.
Product field	Displays the product name.
Description field	Displays a description of the product.
License Expiration field	The date on which the license expires for **Subscription** or **Evaluation** licenses or **Perpetual** for a permanent license.
Maintenance Expiration field	The date on which the product maintenance and support license expires.
Activation Code field	The code that activates the license for BreakingPoint. Refer to the email you received to know the activation code needed to install and use the application.
Quantity field	Displays the number of licenses for the product.
Offline Operations button	Provides licensing operations when the system is offline or not connected to the internet. For example, license activation, deactivation and sync.
Deactivate Licenses button	Select this button to deactivate the selected license. Specify the **Quantity** of licenses you want to deactivate.

Activating Licenses

Before Starting Activation

Ensure the following information is available before starting the license activation process:

Activation code for the license: An email is sent with the Activation Code when you purchase Ixia software. Enter the Activation Code in the **VM License LS+** window to activate the license.

An example e-mail message with the Activation code underlined is shown here:

```
Dear Ixia QA representative,
Thank you for your recent Ixia software purchase. This document contains
important information for activating your software products. Please retain this
information for future reference.
Organization: Ixia QA
Ixia Sales Order#: IxiaQA-RES0HB7X
```

This document provides the right to activate the following product(s) under Entitlement IxiaQA-RES0HB7X:

Product	939-9600, BreakingPoint, Virtual Edition (VE) FLOATING Subscription License
Quantity	100
Activation Code	AA3B-C6CF-3780-3044
Effective Date	2015-01-27
Maintenance Expiration Date	2015-02-26

As a registered customer, you can access software, release notes, and installation instructions from the Ixia website:
http://www.ixiacom.com/support/downloads_and_updates/index.php
If you do not currently have a username and password for the Ixia website, you can request one: http://www.ixiacom.com/support/pwrequest.php
Ixia Technical Support is available to licensed customers who have active software maintenance for their applicable software products. To obtain technical support, go to the support section of Ixia web site:
http://www.ixiacom.com/support
Alternatively, you can contact Ixia Technical Support directly:
support@ixiacom.com
Domestic: (877) FOR-IXIA
International: +1-818-871-1800 (press 1)
Sincerely,
Ixia Order Fulfillment

Activate License

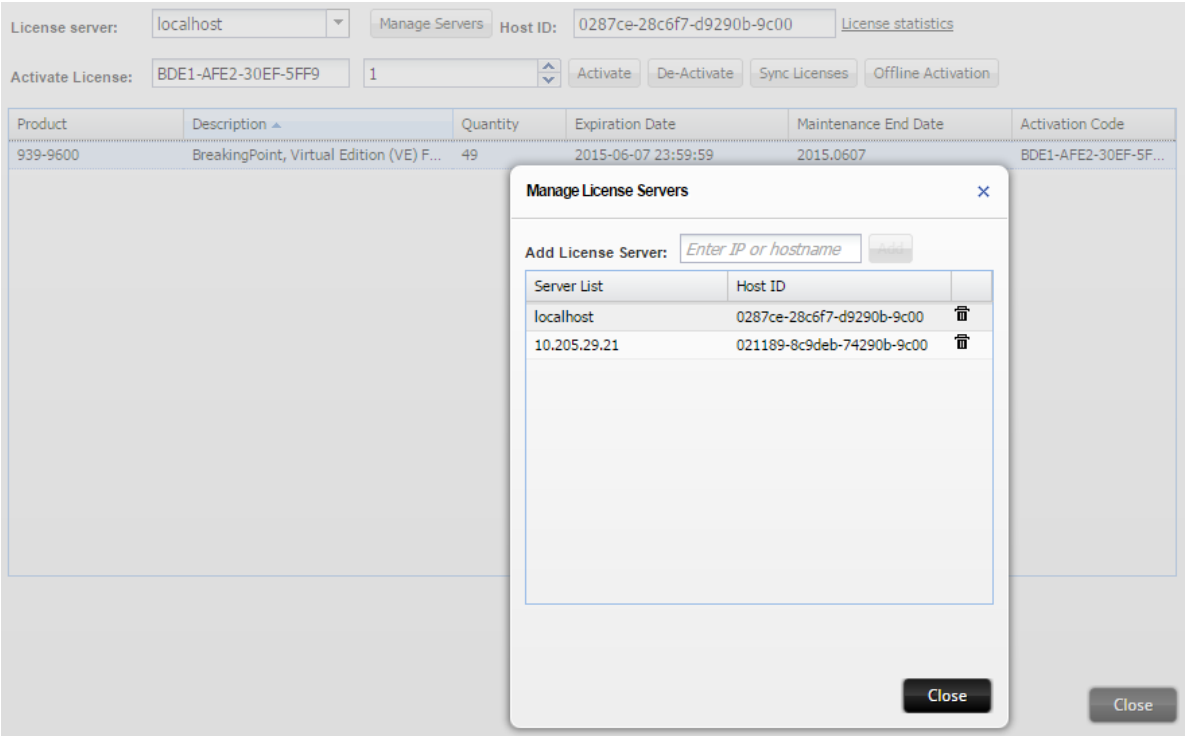
Ensure that vController is connected to internet and that the necessary information discussed previously in Before Starting Activation on the previous page is available.

To activate a license, perform the following tasks:

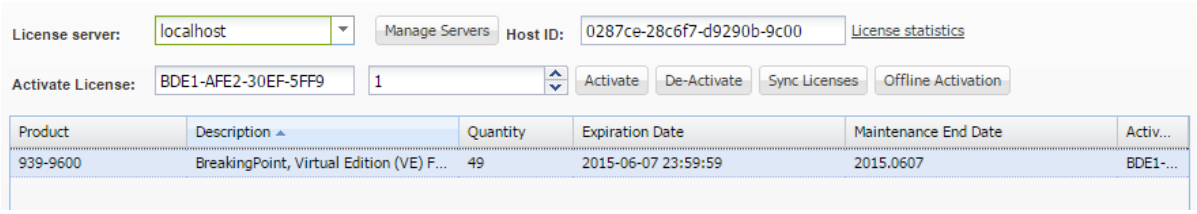
1. Connect to the management IP of vController using a web browser.
2. Go to **BPS Session** > **Control Center** > **Administration** > **Licensing**.

   The **VM Licenses** window opens.
3. In the **License server** box, select the license server IP or Localhost.

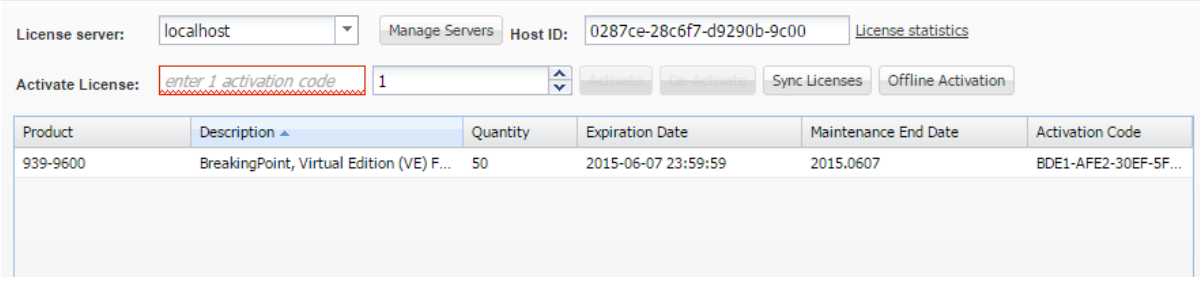
   Note: If you want to add a new license server, select the **Manage Servers** button and provide server details in the **Manage License Servers** dialog box.



4. In the **Activate License** text box, enter the Activation Code and the license quantity as depicted in the following image.



5. Select **Activate**. The activated license is now available in the **VM Licenses** window.



10G Subscription and Perpetual Licenses

This section of the installation guide describes BPS VE licensing that allows a single user to run tests with a TPUT (throughput) between 1Gbps to 10Gbps (maximum).

One unit of this license will allow a single user to execute a test consisting of the following:

- 10Gbps TPUT or 20,000,000 (20 million) CC
- Up to 2 security components

During license checkout, the four license types will be checked out in sequence shown based on the algorithm described in detail below.

- 10G-Subs (Subscription)
- 10G-Perp (Perpetual)
- 1G- Subs
- 1G-Perp

Note: Subscription license types get higher preference than perpetual license types.

License Checkout Algorithm

For each of the license types, based on the sequential order (that is, 10G-Subs, 10G-Perp, 1G-Subs, 1G-Perp), BPS VE will check with each license server for availability of license count.

1. License count is decided by the expression **Floor** (Remaining-license-count / (Multiplicative-factor for the test component considered).
2. License type of immediate preceding value (10G-*) in the sequence mentioned will be considered if a lower valued license type (1G-*) is not available. In that case, license count is 1. The surplus lower valued licenses will be released.

License Checkout Examples

Case 1

For this example, consider a premises that has 2 license servers. The different types of BPS VE licenses counts are shown in the following table:

License Servers	10G-Subs	10G-Perp	1G-Subs	1G-Perp
LicSvr1	2	1	12	2
LicSvr2	10	0	0	0

A user needs to run a 41Gbps TPUT test. The License Checkout sequence will be as described below:

Test Type - non security TPUT. Multiplicative factors are 10 and 1 respectively for 10G-* and 1G-*.

License Checked	Remaining License	License	License	Remaining

out	Count	Requested	Granted	
2 x 10G-Subs from LicSvr1.	41	Floor(41/10) = 4	2	41 – (2 * 10) = 21
2 x 10G-Subs from LicSvr2.	21	Floor(21/10) = 2	2	21 – (2 * 10) = 1
1 x 1G-Subs from LicSvr1.	1	Floor(1/1) = 1	1	1 – (1 * 1) = 0

Case 2

For this example, consider the license count available in the servers is as shown below:

License Servers	10G-Subs	10G-Perp	1G-Subs	1G-Perp
LicSvr1	1	0	0	0
LicSvr2	10	0	0	0

A user needs to run a test with 5 security components. Multiplicative factors are 2 and 1 respectively.

License Checked out	Remaining License Count	License Requested	License Granted	Remaining
1 x 10G-Subs from LicSvr1.	5	Floor(5/2) = 2	1	5 – (1 * 2) = 3
1 x 10G-Subs from LicSvr2.	3	Floor(3/2) = 1	1	3 – (1 * 2) = 1
With 1 pending unit and no 1G-* license available, the algorithm will now look for the license type of the immediately preceding value (10G-*).				
1 x 10G-Subs from LicSvr2.	1	1	1	NA

Case 3

For this example, consider the license count available in the servers is as shown below:

License Servers	10G-Subs	10G-Perp	1G-Subs	1G-Perp

LicSvr1	2	0	1	0
LicSvr2	0	0	1	3

The user needs to run a test with TPUT of 17Gbps.

License Checked out	**Remaining License Count**	**License Requested**	**License Granted**	**Remaining**
1 x 10G-Subs from LicSvr1.	17	Floor(17/10) = 1	1	17 – (1*10) = 7
1 x 1G-Subs from LicSvr1.	7	Floor(7/1) = 7	1	7 – (1 * 1) = 6
1 x 1G-Subs from LicSvr2.	6	Floor(6/1) = 6	1	6 – (1 * 1) = 5
3 x 1G-Perp from LicSvr2	5	Floor(5/1) = 5	1	5 – (3 * 1) = 2
With 2 pending unit and no 1G-* license available, the algorithm will now look for the license type of the immediately preceding value (10G-*).				
1 x 10G-Subs from LicSvr2.	2	1	1	Surplus = 10 – 2 = 8
Release lower valued licenses up to surplus number.				
Release 2x1G-Subs				
Release 3x1G-Subs				

De-Activating Licenses

Introduction

A license, once activated, is said to be assigned to the license server specified during activation process. It may only be served to various applications on various workstations from this license server.

A license can be deactivated, including all of its features, at any time.

Before starting the deactivation process, ensure that the following information is available:

1. **Activation Code** for the license to be deactivated.
2. **Workstation name**: This is the name of the vController that currently uses the licensed software.
3. **License Server Hostname/IP**: The license server where the licenses are currently being registered to.

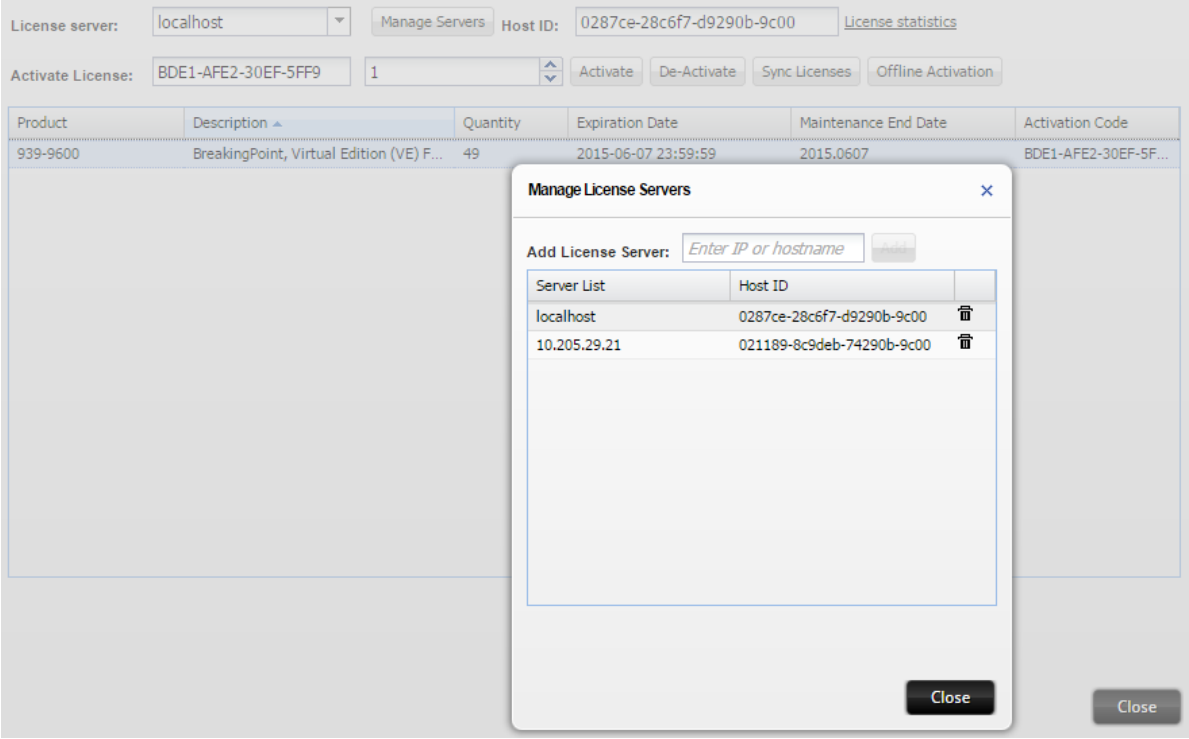
An example of the Ixia activation e-mail message, with the activation number is provided in Before Starting Activation on page 152.

License Deactivation

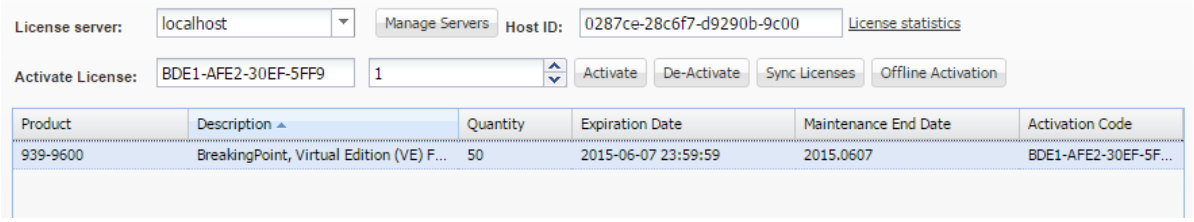
To deactivate a license, perform the following tasks:

1. Connect to the management IP of the vController using a web browser.
2. Go to **BPS Session** > **Control Center** > **Administration** > **Licensing**.
   The **VM Licenses** window opens.
3. In the **License server** box, select the license server IP or Localhost.

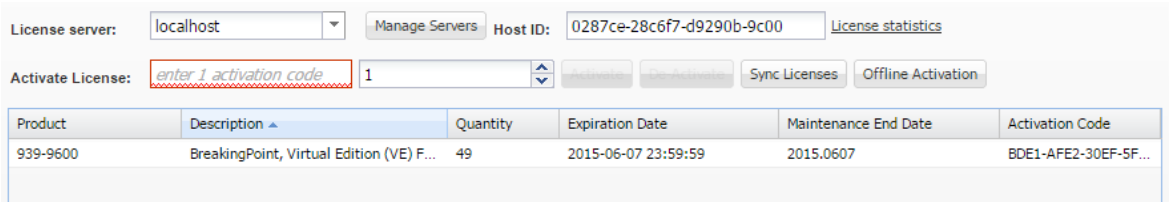
   Note: If you want to add a new license server, select the **Manage Servers** button and provide server details in the **Manage License Servers** dialog box.



4. In the **Activate License** text box, enter the Activation Code and the license quantity that you want to deactivate as depicted in the following image.



5. Select **Deactivate**. The activated license is now removed from the corresponding license server window.



Overview of Offline Activation/Deactivation

Offline activation/deactivation of licenses is required when the BreakingPoint Virtual Edition is deployed in a network that cannot access the internet. As a solution, you can generate the license file from a computer with internet and then transfer the file to the vController running as license server. The license file when imported, activates/deactivates the license.

For both activation and deactivation, it is required to generate the license file from the Fulfillment Router (FR) page.

Offline Activation

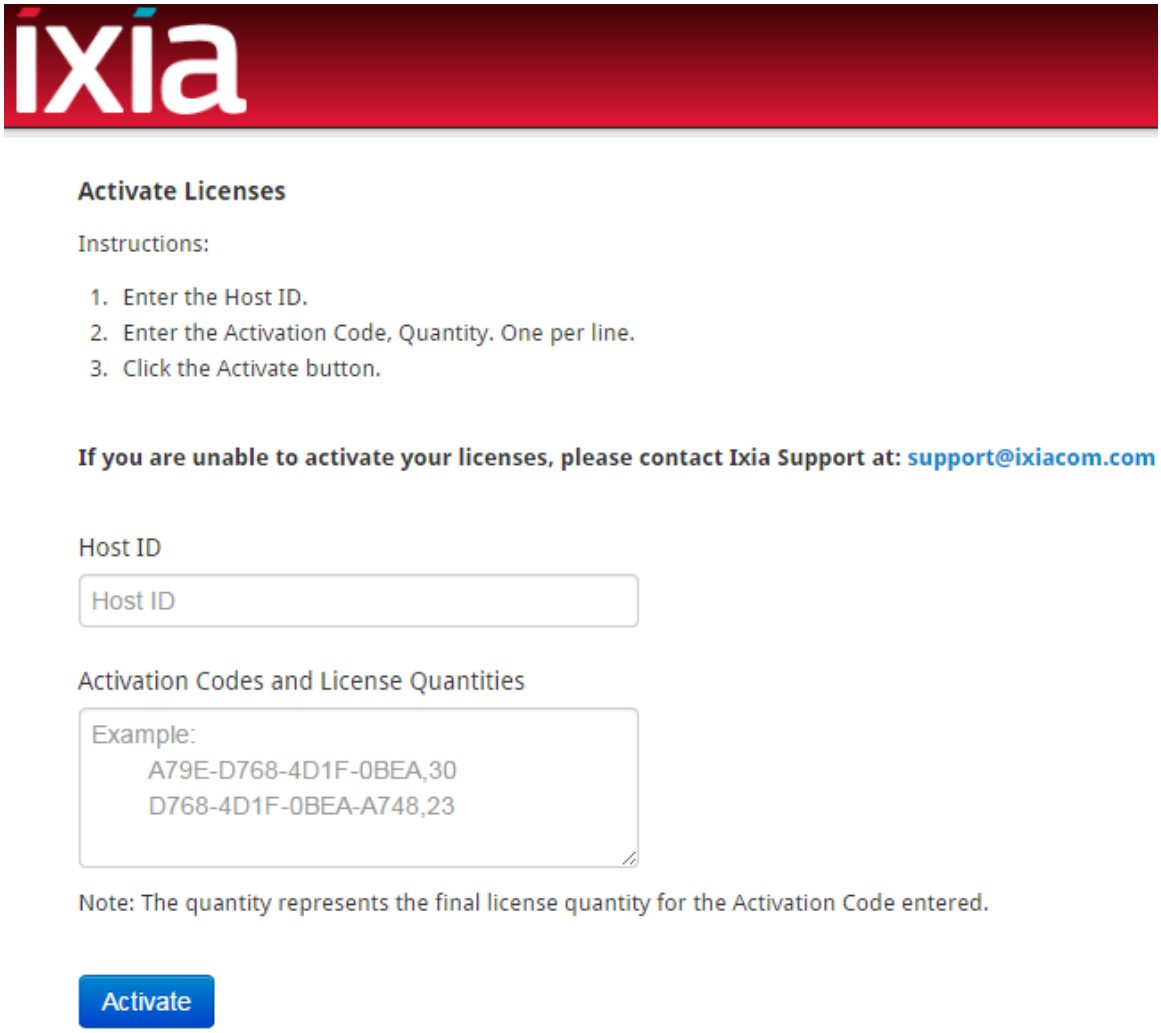
Ensure network connectivity and that the necessary information discussed in Before Starting Activation on page 152 is available. The steps for offline activation process are as follows:

- Step 1: Generate the license file from a computer with internet connection below
- Step 2: Import the License File on page 161

Step 1: Generate the license file from a computer with internet connection

To generate the license file, perform the following tasks:

1. Go to Fulfillment Router (FR) page at: https://fulfillment-prod.ixiacom.com/activation



2. In the **Host ID** text box, enter the Host ID of the vController where the licenses are going to be installed.
   a. Using a web browser, connect to the BreakingPoint vController IP address.
   b. Select **BPS Session** > **Control Center** > **Administration** > **Licensing**.
      The **VM Licenses** window opens.
   c. Select the required License Server.
   d. Get the Host ID from Host ID field.
3. In the **Activation Codes and License Quantities** text box, enter the activation codes as specified in the e-mail and quantity of licenses you want to activate.
   - Here, the **Quantity** represents the final license quantity that you want to activate. For example, if an **Activation Code** with six quantities is already registered in the license server, and when you specify the **Activation Codes and License Quantities** as seven for the same **Activation Code**, then it means the effective quantity is seven and not 13.

- You can perform offline activation for multiple activation codes at once. The syntax is:
  <ActCode1>, <FinalQty1><NEWLINE>
  <ActCode2>, <FinalQty2><NEWLINE>
  ....

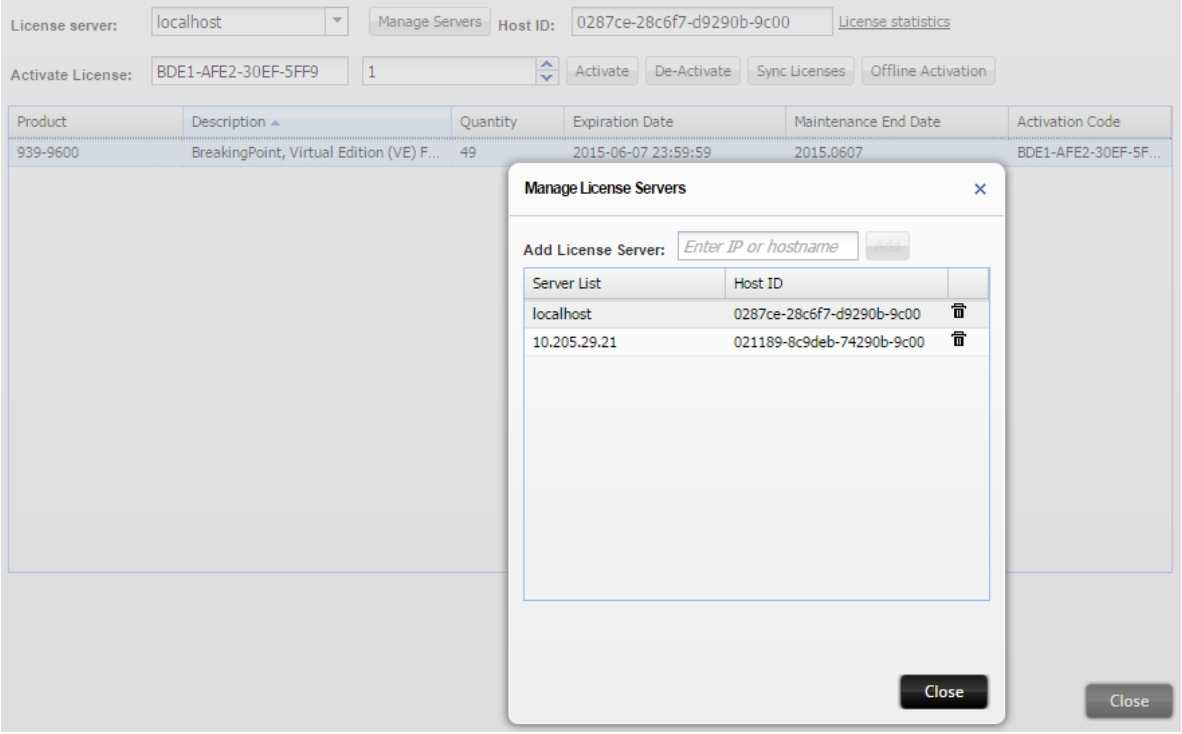
4. Select **Activate**.
   The system generates the license file in `.bin` format, prompting you to open or save it.
5. Save the license file in the required location and transfer it to the vController where the licenses are going to be installed.

Step 2: Import the License File

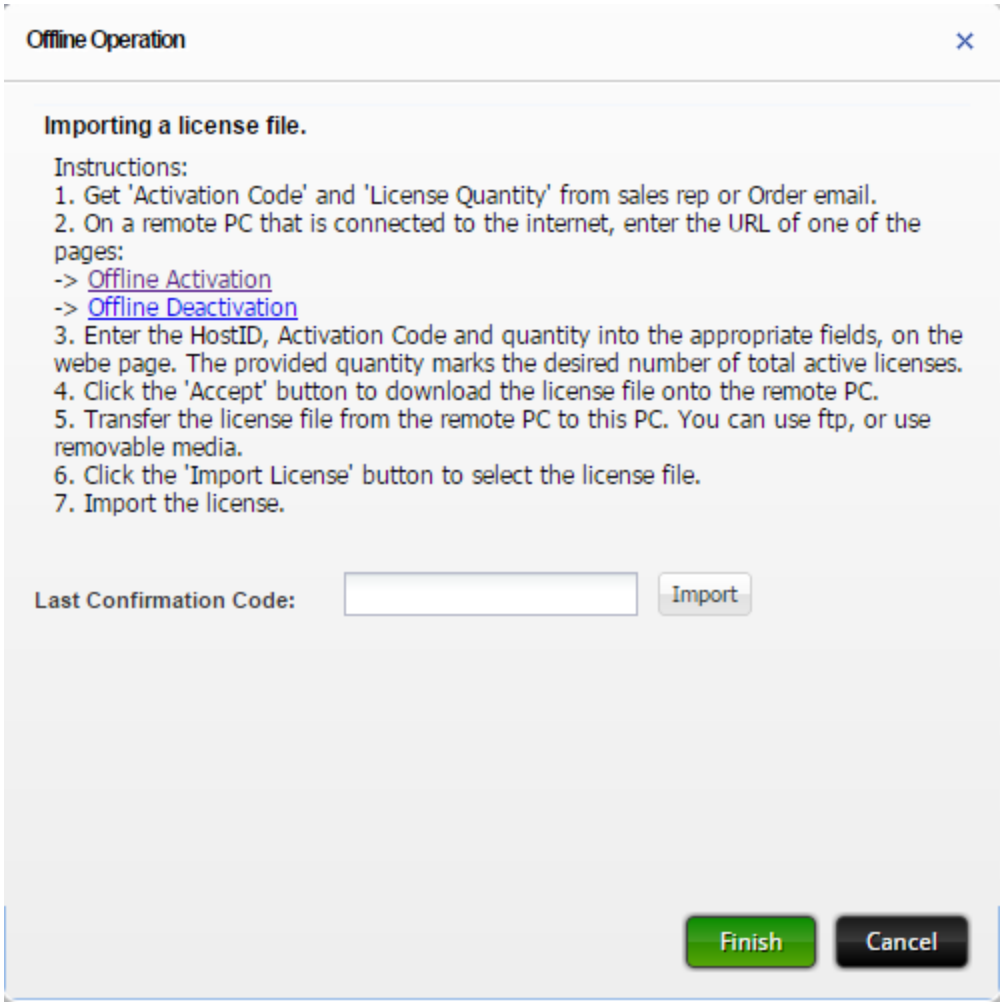
To import the license file, perform the following tasks:

1. Connect to the management IP of the vController.
2. Go to **BPS Session** > **Control Center** > **Administration** > **Licensing**.
   The **VM Licenses** window opens.
3. In the **License server** box, select the license server IP or Localhost.

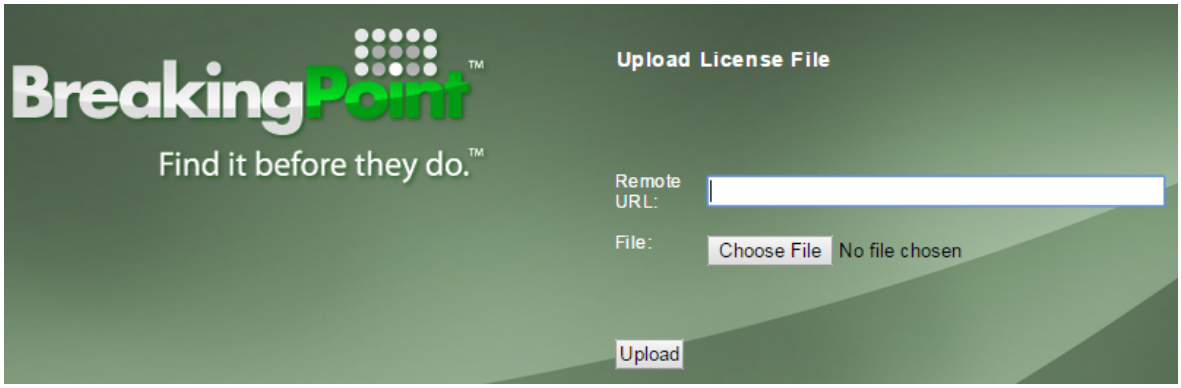
   Note: If you want to add a new license server, select the **Manage Servers** button and provide server details in the **Manage License Servers** dialog box.



4. Select **Offline Activation**.

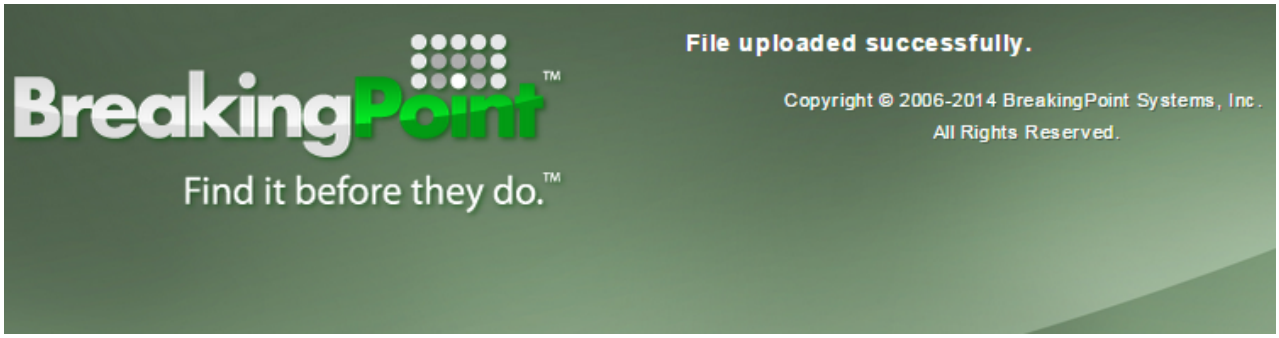


5. In the **Offline Operation** dialog box, select **Import**.
   The **BreakingPoint Systems** window appears asking you to **Upload License File**.



6. Select **Choose File** and open the license file intended for import.

7. Select **Upload** to complete the import.
   On successful upload, the following message appears.



8. In the **Offline Operation** dialog box, select **Finish** to complete the activation process.
   The license is now available for use on the relevant license server.

Offline Deactivation

Before starting the deactivation process, ensure that the following information is available:

- Host ID of the computer
- Activation Code for the license to be deactivated

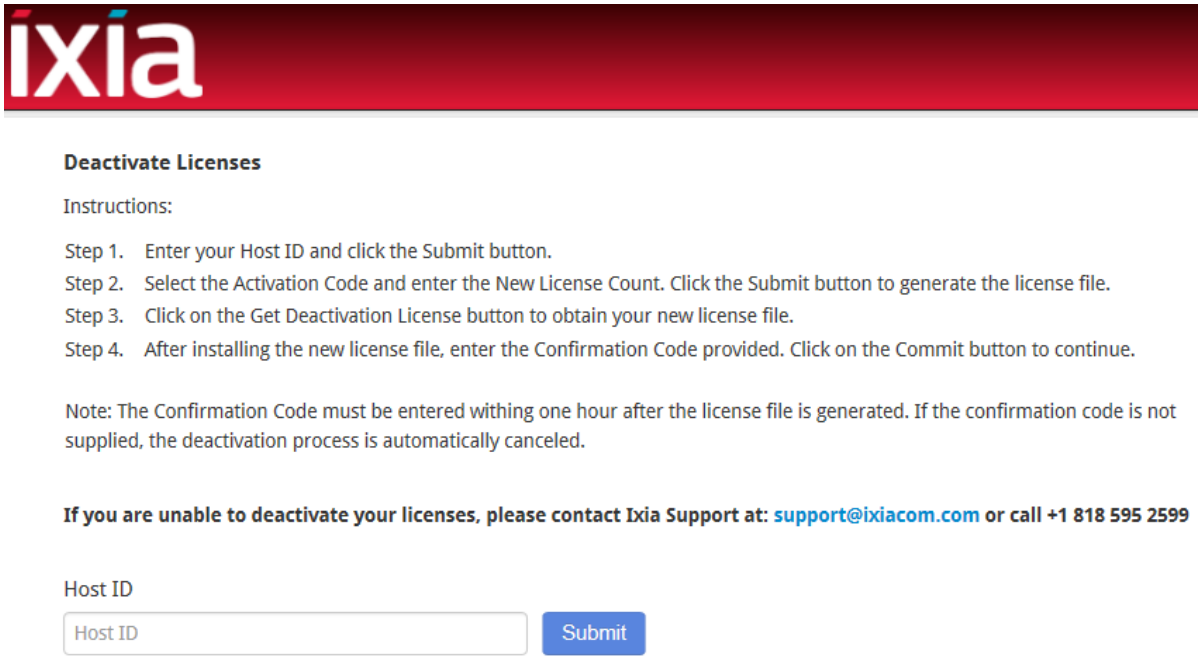
The steps for offline deactivation process are as follows:

- Step 1: Generate License File below
- Step 2: Import License File on page 166
- Step 3: Submit Confirmation Code on page 169

Step 1: Generate License File

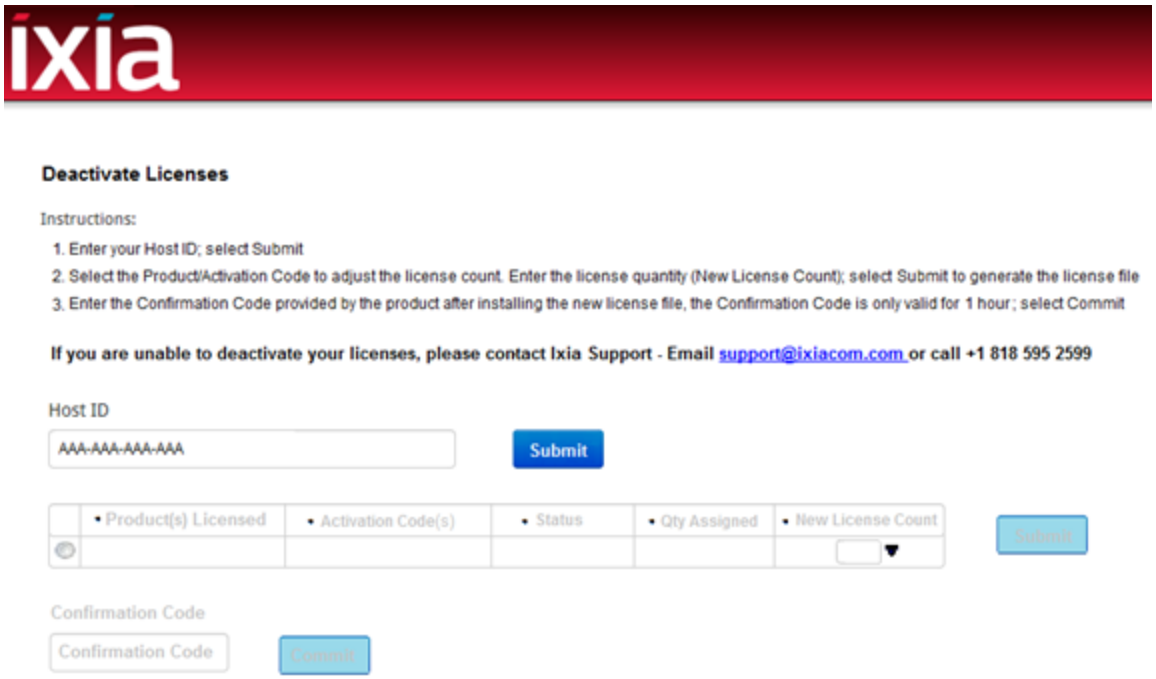
To generate the license file, perform the following tasks:

1. Go to the Fulfillment Router (FR) page at: https://fulfillment-prod.ixiacom.com/deactivation.



2. In the **Host ID** text box, enter the Host ID of the vController where the licenses are going to be installed.
3. Select **Submit**.

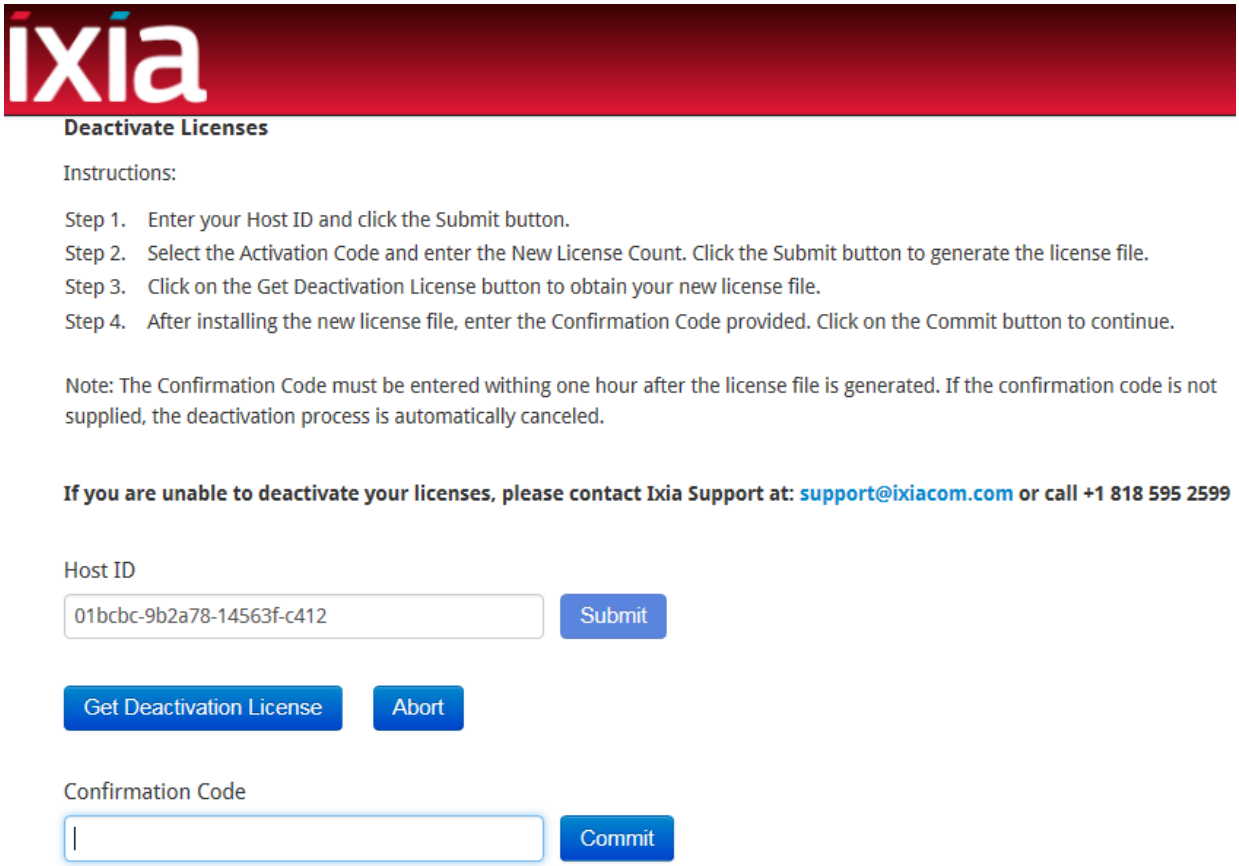
   The system lists all the licenses activated for the specified host.



4. Specify a new value in the **New License Count** list for the selected license. The system updates the license quantity to this new value. Selecting zero, completely deactivates the license.

   Note: At a time, you can perform deactivation for a single activation code only.

5. Select **Submit**.
6. Select **Get Deactivation License** to generate the license file.



7. Save the license file in the required location and transfer it to the vController where the licenses are going to be installed.

At this point, you must enter the **Confirmation Code**, and then select **Commit** to complete the deactivation. **Confirmation Code** is available after importing the license file as explained in Step 2: Import License File on the facing page. The validity of the confirmation code is 48 hours and you have to submit the confirmation code within the time frame to complete the deactivation process.

After generating the license file, FR maintains the state of Host ID for 48 hours. It means, during this period, server cannot perform additional activation/deactivation in the FR for that Host ID, until you either submit the confirmation code or abort the deactivation process.

You can perform the following actions in **Deactivate Licenses** window:

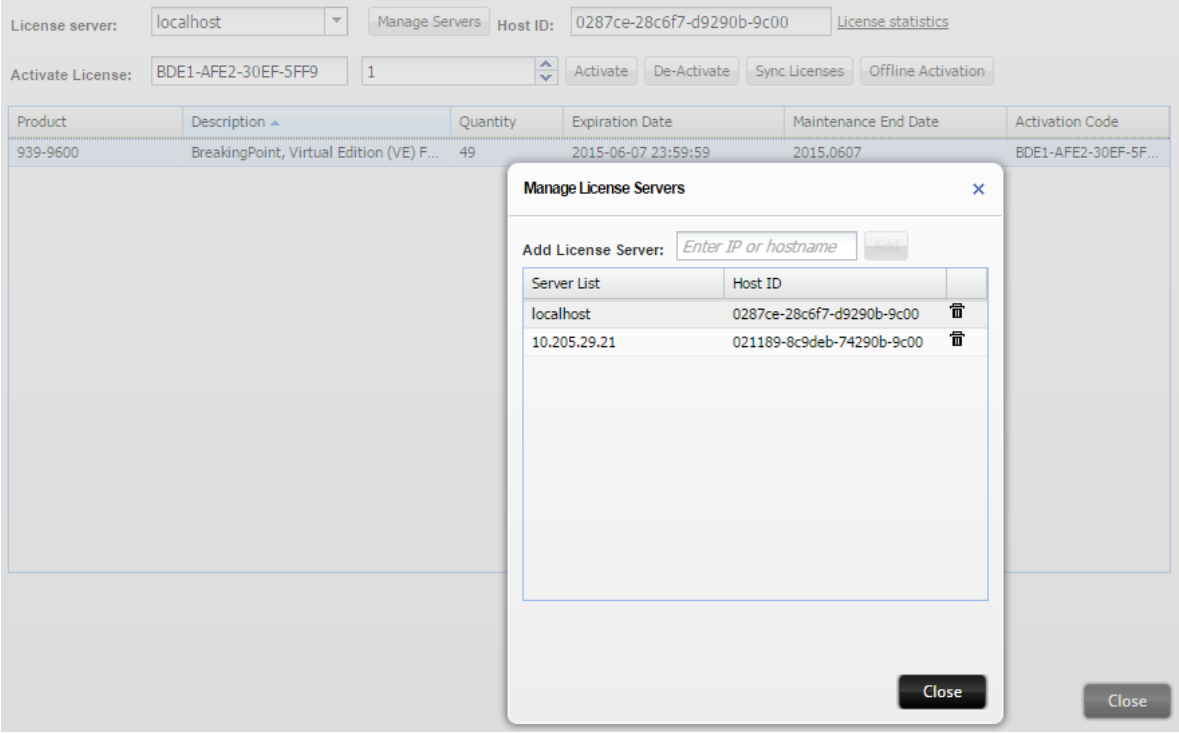
- **Abort** - Cancel the offline deactivation process. The licensed quantities are retained as before.
- **Get Deactivation License** - Generate the deactivation license file that must be imported to the computer installed with BreakingPoint. In case the file is lost, select again to regenerate the license file.
- **Commit** - Submit the confirmation code. Until the confirmation code is committed, the deactivation process in not complete.

Step 2: Import License File

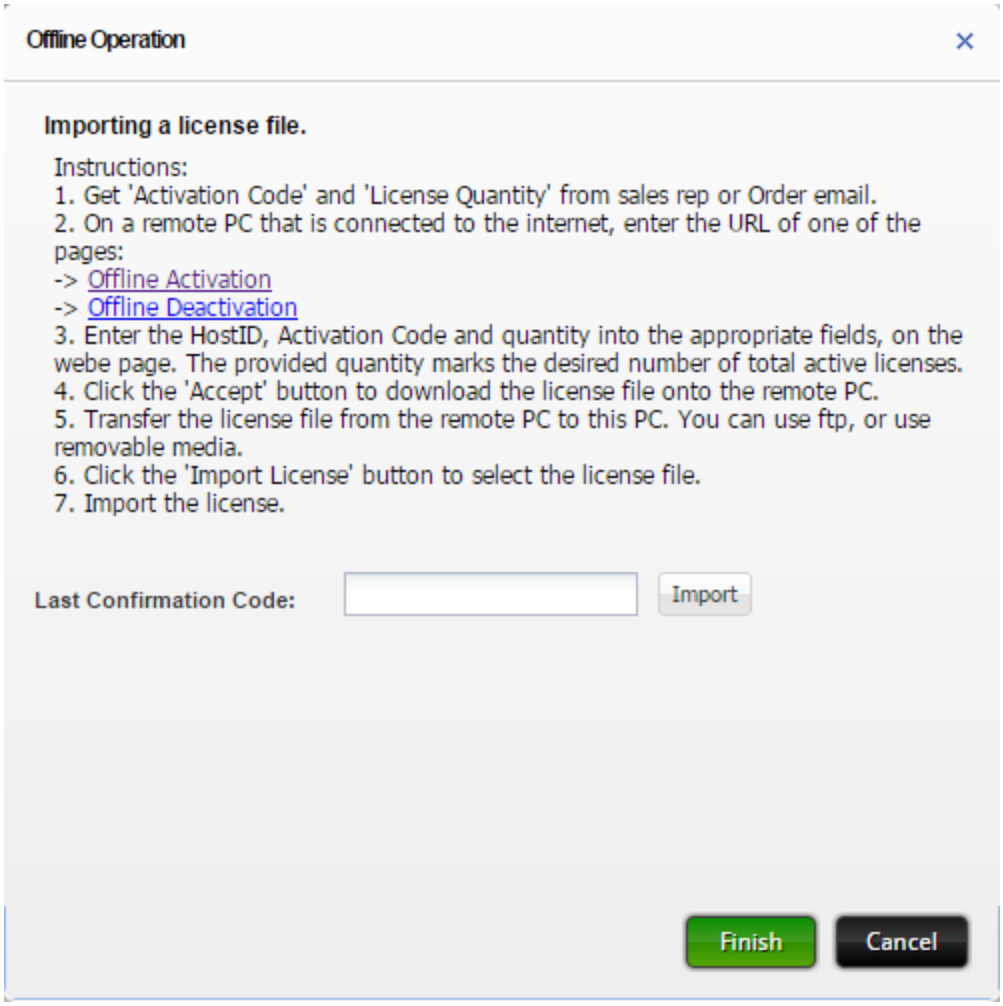
1. Connect to the management IP of the vController using a web browser.
2. In the computer installed with BreakingPoint, select **BPS Session** > **Control Center** > **Administration** > **Licensing**

   The **VM Licenses** window opens.
3. In the **License server** box, select the license server IP or Localhost.

   Note: If you want to add a new license server, select the **Manage Servers** button and provide server details in the **Manage License Servers** dialog box.

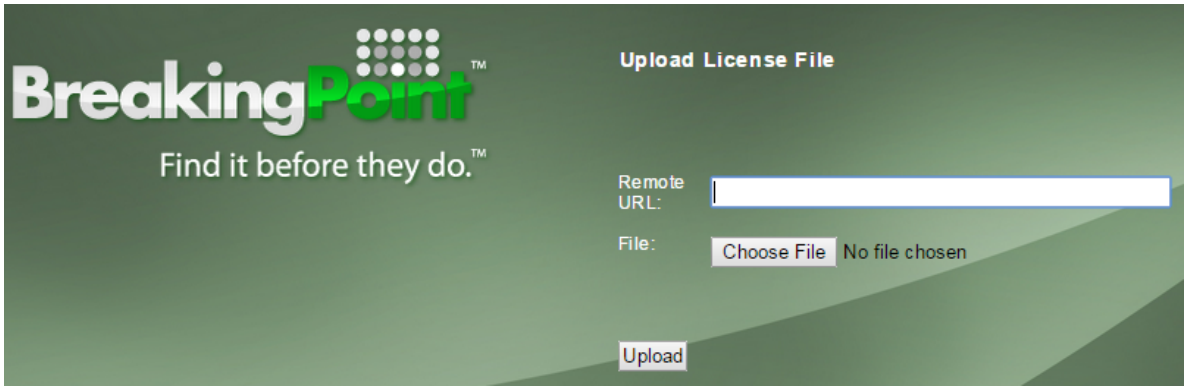


4. Select **Offline Activation**. The **Offline Operation** dialog box opens.



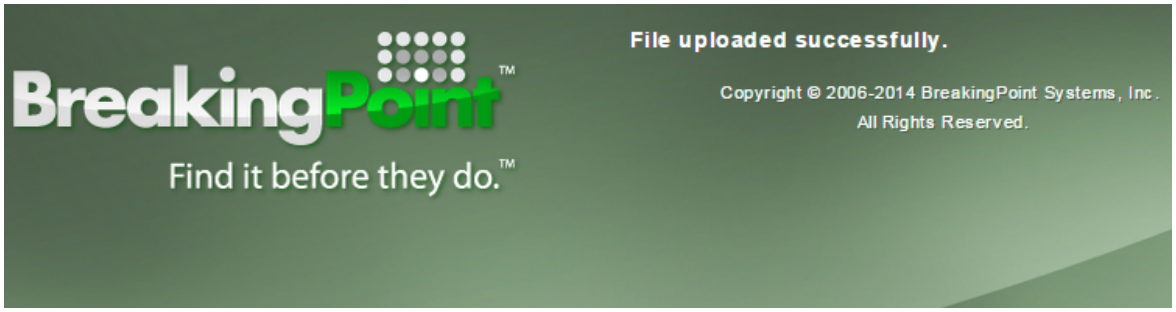
5. Select **Import**.

   The **BreakingPoint Systems** window appears asking you to **Upload License File**.

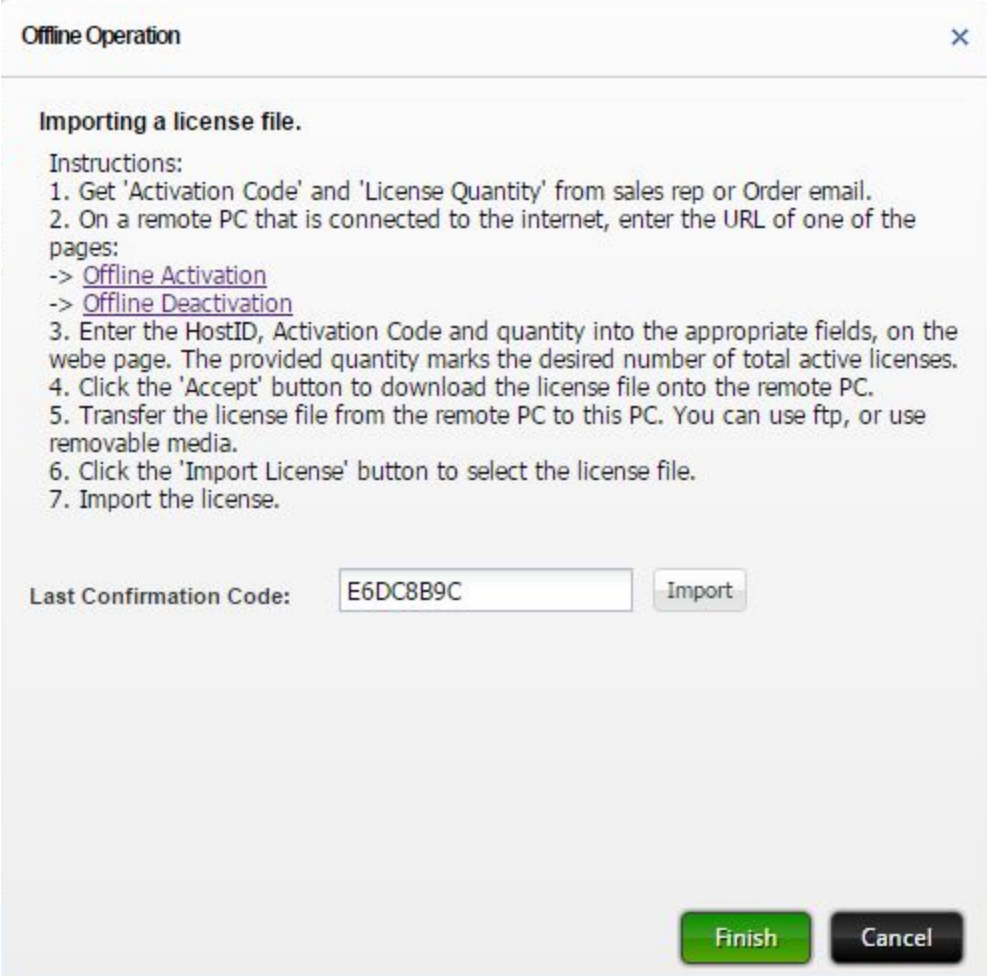


6. Select **Choose File** and open the license file intended for import.

7. Select Upload to complete the import.
   On successful upload, the following message appears.



8. In the **Offline Operation** dialog box, Select **Finish**.
   The system generates the **Confirmation Code** as depicted in the following image. You have to submit this code in the deactivation window. Make a note of this code.



> **Note:** In case you lose the **Confirmation Code**, select the **Offline Activation** button again. The **Offline Operation** dialog box displays the **Last Confirmation Code** for the **Last Imported File**.

Step 3: Submit Confirmation Code

1. Go to step 6 in Step 1: Generate License File on page 163.
2. Enter the **Confirmation Code**.
3. Select **Commit**.

The license is now deactivated.

CHAPTER 14 Troubleshooting

This chapter provides recommended solutions for issues you may encounter while deploying or using BreakingPoint Virtual Edition.

Unable to Track Modified IPs

After the deployment of the System Controller and Virtual Blades, the IP addresses for these components are stored in the vController and displayed at the console. These IP addresses allow the components to recognize each other and populate slot information in the **Manage Virtual Chassis** and **Device Status** areas of the user interface.

If the IP addresses of the vBlades change for any reason (for example, due to new IP addresses being issued from DHCP) the vController will not be aware of the new IP addresses. This will result in the BPS Chassis View indicating that ports are not available.

Solution

Perform the following tasks to resolve the problem:

1. Go to **VM Deployment** > **Manage Virtual Chassis**. Delete one of the slots. This task empties the slot in the Manage Controller.
2. Delete the virtual machine from vSphere. This Virtual Machine (VM) should not be used for any other purpose.
3. Install the Virtual Blades again from the **VM Deployment**. New IP addresses for the Virtual Machine (VM) are added in the **Manage Virtual Chassis** and **Device Status** areas of the user interface.

Virtual Blades Not Available

In a scenario where the IP address of the System Controller has changed, the vBlades will not be available in the **Manage Virtual Chassis** area of the user interface. Note that NIC1 of the vController (Refer to Network Topology Diagram) is used for System Controller and vBlade communications.

Solution

Perform the following tasks to resolve this problem:

1. Go to **Manage Virtual Chassis** and delete all Virtual Blades from the vSphere.
2. Deploy VM again so that new entries are created in the vController and recognized in **Manage Virtual Chassis** and **Device Status**.

Cannot Connect to a Hypervisor from the BPS VE User Interface

In a scenario where you cannot connect to a Hypervisor from the BreakingPoint Virtual Edition user interface, try making the following modifications on the Hypervisor to resolve the issue.

Solution

1. sudo vi /etc/ssh/sshd_config
2. Modify line "PermitRootLogin without-password" with "PermitRootLogin yes"
3. sudo service ssh restart

Permission Denied/Temp Error Occurs at Power Up

While trying to deploy vBlades from the BreakingPoint Virtual Edition UI, you may receive the following error, "permission denied /temp".

Solution

Make the following modifications on the Hypervisor to resolve the issue.

- UBUNTU Setup

1. Add " /tmp/* rw," in the file /etc/apparmor.d/abstractions/libvirt-qemu to grant write permission on /tmp
2. Restart AppArmor: #/etc/init.d/apparmor restart

- CENTOS Setup

SELinux needs to be disabled on the host machine.

1. Set SELINUX=permissive in file /etc/sysconfig/selinux and Save
2. Reboot the system

BP VE User Interface Not Performing as Expected

The user interface has become unresponsive or is not performing as expected.

Solution

Make the following operating system modifications at the host.

1. Export PATH variable - export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
2. Execute command: apt-get update
3. Add following entries to /etc/sysctl.conf:
   net.bridge.-nf-call-ip6tables = 0
   net.bridge.-nf-call-iptables = 0
   net.bridge.-nf-call-arptables = 0

net.bridge.-nf-filter-vlan-tagged = 0

4. Execute command: sysctl –p
5. Recreate bridges
6. Set txqueuelen for vnet1 & vnet2 to 12000
7. Select Model as **Nehalem** under the processor configuration section and select **Copy Host CPU Configuration**
8. Delete unwanted devices
9. Before running the test ensure that: vhost_net module loaded using command: lsmod | grep vhost
10. Turn off the firewall using the command: ufw disable

Permission Denied Error Occurs While Trying to Deploy vController

A "permission denied" error may be observed in the console or Virtual Machine Manager at the host while trying to deploy the vController.

Solution

- Enable root access for QEMU guests:
  - Edit file /etc/libvirt/qemu.conf and uncomment Line (1)User = "root" and (2)group = "root"
- Restart libvirt daemon:
  - #/etc/init.d/libvirt-bin restart
  - #/etc/init.d/libvirtd restart

Restart Connection Interruption During KVM vBlade Deployment

Please be aware that during vBlade deployment from the BPS user interface in the KVM setup, a restart connection interruption may occur in the Virtual Machine Manager on the host machine due to the Libvirt service.

vBlade Memory Errors

When the system has 64MB or less of free memory, a vBlade will generate low memory error messages in 120 second intervals.

Solution

In a scenario where the system becomes unstable due to low memory, try the following steps to resolve the issue. For best results, perform these steps in order.

1. Reduce "Maximum Simultaneous Super Flows".
2. If running a multicomponent test, reduce the number of components.
3. Reduce the number of vBlade NICs that are used.
4. Reduce the number of IP addresses if "Per-host Stats" is enabled.

vController Memory Errors

When the system has 64MB or less of free memory, a System Controller will generate low memory error messages in 120 second intervals.

Note: There should be a balance between the System Controller and the number of supported vBlades based on the resources provided to the System Controller.

CHAPTER 15 Upgrade the BPS VE Software

In order to upgrade BreakingPoint VE software, you must download the appropriate update file from either of the following sites (which will require a password for access):

https://strikecenter.ixiacom.com/bps/osupdates

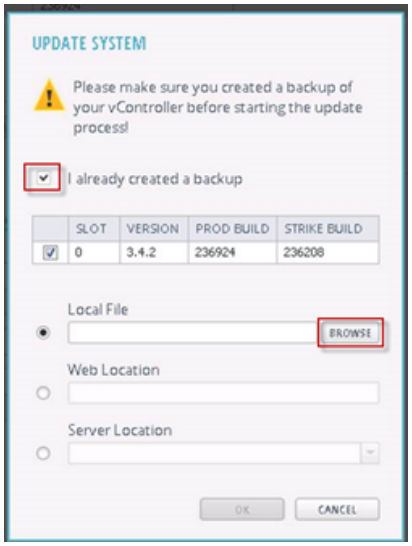
http://www.ixiacom.com/downloads-updates (select BreakingPoint Virtual Edition)

You will also need to obtain the applicable release notes from the website. The release notes describe new features, resolved issues and known issues that may affect the BPS VE installation, upgrade and operation.

Note: You must have BreakingPoint VE controller version 3.4.2 or higher to perform this upgrade.

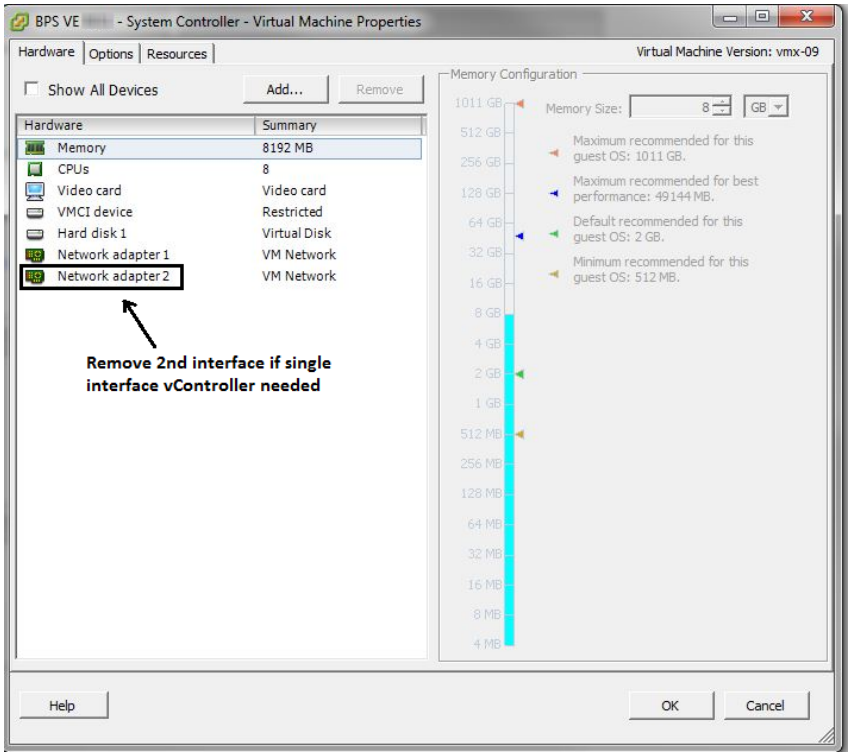
To upgrade BPS VE:

1. Download the BreakingPoint Virtual Edition VM update file.
2. Log in to the Ixia BreakingPoint VE System.
3. Navigate to **ADMINISTRATION -> SYSTEM SETTINGS -> UPDATES**.
4. Select **UPDATE SYSTEM** and then see the image below.
   a. After you have created a backup of your vController, select the, **I already created a backup**, option.
   b. Browse to the location of the BreakingPoint VE update file and select **OK** to start the update.



5. The BreakingPoint VE update will take 15-20 minutes to complete.
6. To verify that the update has been installed, see the version information in the Installed Applications section of the **UPDATES** tab.

Note: After upgrading the BPS VE vController from 8.01 (or earlier releases) to release 8.10, the vController will continue to display 2 interfaces. To operate using a **Single Interface vController**, access the Virtual Machine Properties and delete the 2nd interface (Network adapter2) as shown in the image below. **Do not delete the 1st interface.**



APPENDIX A Supported Platforms

Certified and compatible platform versions

It is always recommended to use the latest versions of the virtualization platforms. BPS VE works best in the virtualization platforms that are qualified by Ixia. However, if you have not upgraded to the latest virtualization platforms, use the compatible versions mentioned in the table below. The versions mentioned in the Compatible column are not qualified by Ixia. In rare cases, you may experience errors when you deploy BPS VE in environments that are not qualified by Ixia. In these scenarios, the proposed solution is to use the recommended version of the virtualization platform.

Category	Qualified	Compatible
Hypervisor and Host	VMware vSphere ESXi 6.X	KVM over CentOS 6.X
	KVM over CentOS 7.X	KVM over Ubuntu 14.04 LTS
	KVM over Ubuntu 16.04 LTS 64-bit	KVM over RHEL 6.X
	KVM over Ubuntu 18.04 LTS 64-bit	KVM over RHEL 7.X
Management and Orchestration	OpenStack Stein (vanilla distribution)2	Other OpenStack-based platforms (vanilla distributions)2
		OtherOpenStack-based platforms
		(vendor-specific distributions)2
Network Connection and vNIC Driver	Virtual Switch	VMXNET3 (VMware)

Category	Qualified	Compatible
		VIRTIO (KVM)
	PCI PassThrough	Intel 10G ixgbe (all platforms)
		Intel 25G i40e (all platforms)
		Intel 40G i40e (all platforms)
		Mellanox 10G mlx4 (all platforms)
		Mellanox 25G mlx5 (all platforms)
		Mellanox 40G mlx5 (all platforms)
	SR-IOV	Intel 10G ixgbevf (all platforms)
		Intel 25G i40evf (all platforms)
		Intel 40G i40evf (all platforms)
		Mellanox 10G mlx4 (all platforms)
		Mellanox 25G mlx5 (all platforms)
		Mellanox 40G mlx5 (all platforms)
Virtual Switch Model	Virtual Standard Switch (only on VMware)	Linux Bridges (only on OpenStack) 2
	Linux Bridges (only on KVM)	
	Open Virtual Switch versions 2.2/2.4/2.6 (only on KVM) OpenVirtual Switch (only on OpenStack)	

Certified/compatible boards

The following table lists the boards that are certified/compatible on various host servers.

Card	Vendor	Speed (Gbps)	Driver Version on Guest				Delivered As
				VMware ESXi 6.0	KVM CentOS / OpenStack	KVM Ubuntu / OpenStack	
X520	Intel	10	ixgbe 5.1.3 ixgbevf 4.1.2	ixgbe 4.5.2	ixgbe 5.1.3 / kernel 3.10.0-514.26.2.el7.x86_64	ixgbe 5.1.3 / kernel 4.4.0-62-generic	Certified
X540	Intel	10	ixgbe 5.1.3 ixgbevf 4.1.2	ixgbe 4.5.2	ixgbe 5.1.3 / kernel 3.10.0-514.26.2.el7.x86_64	ixgbe 5.1.3 / kernel 4.4.0-62-generic	Certified
X550	Intel	10	ixgbe 5.1.3 ixgbevf 4.1.2	ixgbe 4.5.2	ixgbe 5.1.3 / kernel 3.10.0-514.26.2.el7.x86_64	ixgbe 5.1.3 / kernel 4.4.0-62-generic	Certified
X552	Intel	10	ixgbe 5.1.3 ixgbevf 4.1.2	ixgbe 4.5.2	ixgbe 5.1.3 / kernel 3.10.0-514.26.2.el7.x86_64	ixgbe 5.1.3 / kernel 4.4.0-62-generic	Compatible
X557	Intel	10	ixgbe 5.1.3 ixgbevf 4.1.2	ixgbe 4.5.2	ixgbe 5.1.3 / kernel 3.10.0-514.26.2.el7.x86_64	ixgbe 5.1.3 / kernel 4.4.0-62-generic	Compatible
X710	Intel	10	i40e 2.0.26 i40evf 2.0.30	i40e 2.0.6	i40e 2.0.26 / kernel 3.10.0-514.26.2.el7.x86_64	i40e 2.0.26 / kernel 4.4.0-62-generic	Certified
XL710	Intel	40	i40e 2.0.26	i40e 2.0.6	i40e 2.0.26 / kernel 3.10.0-	i40e 2.0.26 / kernel	Certified

Card	Vendor	Speed (Gbps)	Driver Version on Guest				Delivered As
				VMware ESXi 6.0	KVM CentOS / OpenStack	KVM Ubuntu / OpenStack	
			i40evf 2.0.30		514.26.2.el7.x86_64	4.4.0-62-generic	
XXV710	Intel	25	i40e 2.0.26 i40evf 2.0.30	i40e 2.0.6	i40e 2.0.26 / kernel 3.10.0-514.26.2.el7.x86_64	i40e 2.0.26 / kernel 4.4.0-62-generic	Certified

APPENDIX B Open Port Requirements for BPS VE

The following ports may need to be included in the security exception list to allow the respective BPS interfaces to pass through firewalls.

Interface between client UI browser (or TCL) and vController (System Controller):

- 80
- 443
- 843
- 1099
- 8880
- 8881

Interface between vController (System Controller) and vBlade (Network processor)

- 8887
- 8889 - 8939
- 8943 - 8945

Interface between vController (System Controller) and an external License Server

- 4501
- 4502
- 27002
- 47392

APPENDIX C Console Commands

This sections provides and overview of the commands that can be from the console of the vController Virtual Machine (VM). For a complete list of console commands, run the **help** command as described below.

You can access the console from your VMware or KVM user interface or SSH.

The following login is required:

user: netadmin

password: netadmin

Welcome Screen

After logging in, a Welcome screen similar to the one shown below will display.

```
~>
~> ssh netadmin@10.216.110.231
netadmin@10.216.110.231's password:
Last login: Mon Sep  7 19:06:18 2015 from
+-------------------------------------------------------+
|  eth0          : 10.216.110.231                       |
|  ctrl0         : 10.216.110.73                        |
|  Machine       : x86_64                               |
|  Kernel        : 2.6.32-504.1.3.el6.x86_64            |
|  System        : CentOS release 6.6 (Final)           |
+-------------------------------------------------------+
netadmin:~$
netadmin:~$
netadmin:~$
```

help

Enter "?" or **help** at the console to see a list of all console commands as shown in the image below.

```
netadmin:~$
netadmin:~$ ?
clear   help     lpath   lsudo   restartservice   showdate
exit    history  ls      pwd     setip            showip
netadmin:~$
netadmin:~$ help
clear   help     lpath   lsudo   restartservice   showdate
exit    history  ls      pwd     setip            showip
netadmin:~$
netadmin:~$
```

For help with the parameters of a specific command, enter the command followed by "-h". For example, **restartservice -h**.

restartservice

See the example below.

```
netadmin:~$
netadmin:~$ restartservice -h
usage: restartservice [-h] -s SERVICE

Restarts the service specified.

optional arguments:
  -h, --help  show this help message and exit
  -s SERVICE  Service, e.g. network
netadmin:~$
netadmin:~$
```

Showdate

See the example below.

```
netadmin:~$
netadmin:~$ showdate -h
usage: showdate [-h]

Prints the system date and time.

optional arguments:
  -h, --help  show this help message and exit
netadmin:~$
netadmin:~$ showdate
Mon Sep  7 19:20:05 PDT 2015
netadmin:~$
netadmin:~$
netadmin:~$
```

Showip

See the example below.

```
netadmin:~$
netadmin:~$ showip -h
usage: showip [-h]

Displays the status of the currently active interfaces.

optional arguments:
  -h, --help  show this help message and exit
netadmin:~$
netadmin:~$ showip
 eth0   : 10.216.110.231
 ctrl0  : 10.216.110.73
netadmin:~$
netadmin:~$
```

Setip

See the example below.

```
metadmin:~$ setip -h
usage: setip [-h] -iface IFACE [-dhcp] [-ip IP] [-mask MASK] [-gw GW]

Sets the IPv4 address for the specified interface.

optional arguments:
  -h, --help    show this help message and exit
  -iface IFACE  Interface, e.g. eth0/ctrl0.
  -dhcp         DHCP/Static, if dhcp, following parameters are ignored.
  -ip IP        IP Address, e.g. 192.168.10.15
  -mask MASK    Netmask, e.g. 24; within range 1 to 31
  -gw GW        [Optional] Gateway, e.g. 192.168.10.1
metadmin:~$
metadmin:~$ setip -iface ctrl0 -ip 192.168.10.15 -mask 24 -gw 192.168.10.1_
```

Note: The interface names that can be used with this command are:

- For vController with a single management interface: ctrl0.
- For vController with 2 management interfaces: eth0 (for external management) and ctrl0 (for internal management).
- For vBlade: eth0.

INDEX

Ixia, a Keysight Business
1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1738