Installation Manual

AudioCodes Mediant™ Family of Session Border Controllers (SBC)

Mediant Virtual Edition (VE) SBC

Deployment on VMware, KVM or Hyper-V

Version 7.4

Table of Contents

List of Tables

List of Figures

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: January-30-2023

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our website at https://www.audiocodes.com/services-support/maintenance-and-support.

Stay in the Loop with AudioCodes

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Related Documentation

Manual Name
SBC-Gateway Series Release Notes for Latest Release (LR) Versions
Mediant Software SBC User's Manual
Mediant Virtual Edition SBC for Microsoft Azure Installation Manual
Mediant Virtual Edition SBC for Amazon AWS Installation Manual
Mediant Virtual Edition SBC for Google Cloud Installation Manual
Stack Manager for Mediant VE-CE User's Manual

Document Revision Record

LTRT	Description
10868	Initial document release for Version 7.4. 20 GB required instead of 10 GB, VMware ESXi 6.5 or later; Windows Server 2019
10877	Hyper-V prerequisites
10894	Typos
10899	Intel X710 NIC; i40e Linux driver

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our website at https://online.audiocodes.com/documentation-feedback.

1 Introduction

AudioCodes' Mediant Virtual Edition (VE) Session Border Controller (SBC), hereafter referred to as *Mediant VE SBC*, is a software product installed and hosted in a virtual machine environment, enabling connectivity and security between enterprises' and Service Providers' VoIP networks.

Mediant VE SBC provides perimeter defense as a way of protecting companies from malicious VoIP attacks; voice and signaling mediation and normalization for allowing the connection of any IP PBX to any Service Provider; and service assurance for service quality and manageability.

The product also offers call "survivability", ensuring service continuity to enterprises served by a centralized SIP-based IP-Centrex server or branch offices of distributed enterprises. Survivability functionality enables internal office communication between SIP clients in the case of disconnection from the centralized SIP IP-Centrex server or IP-PBX.

The product features full management through its Web and CLI management interfaces.

The product enables customers to significantly cut costs due to reduced hardware, power and cooling requirements.

Note: The scope of this document does not fully cover security aspects for deploying the product in your environment. Security measures should be done in accordance with your organization's security policies. For basic security guidelines, see *Recommended Security Guidelines*.

1.1 Mediant VE SBC Offered Flavors

AudioCodes offers several orderable Mediant VE SBC flavors, which are based on the following:

- Maximum SBC call session capacity (with and without transcoding)
- Hypervisor type
- Number of virtual CPUs (vCPU)
- DRAM memory

For a detailed description of the offered flavors, please refer to the *Release Notes*, which can be downloaded from AudioCodes website at https://www.audiocodes.com/library/technical-documents.

1.2 Product Package

The product is delivered as a virtual appliance that can be deployed on VMware® vSphere ESXi™ Version 5.x or later Hypervisor, Linux KVM (Kernel-based Virtual Machine), OpenStack, Microsoft Hyper-V Server, or Amazon Elastic Compute Cloud (Amazon EC2). Different images are provided for each virtual environment.

Customers can choose how to obtain the product package:

- Downloadable file containing the virtual appliance image
- DVD containing the virtual appliance image

2 Installation Prerequisites

Installation prerequisites depend on your ordered Mediant VE SBC flavor (see Section 1.1 on page 9).

2.1 Host Server

The physical server on which Mediant VE SBC is to be installed must meet the following specifications:

Table 2-1: Host Server (Hypervisor) Specifications

Resource	Specifications
Processor type	64-bit Xeon Intel® CPU with support for hardware virtualization (Intel VT-x) enabled with Advanced Vector Extensions (AVX) and AES-NI support (Sandy-Bridge architecture or newer). Capacity figures depend on CPU speed. The capacity figures described in *Release Notes* are based on the reference CPU speed as described in Section 3.12 on page 55.
Number of CPU Cores	CPU cores are required for the hypervisor in addition to the cores required for the SBC virtual machine (refer to the *Release Notes*). The number of cores required for the hypervisor depends on the hypervisor's specific requirements. **Note:** • If hyper-threading is enabled, the number of virtual CPUs for the SBC virtual machine, as specified in the *Release Notes*, refers to cores and not threads. • The SBC's virtual machine cores must not be shared with other virtual machines.
Memory	Refer to the *Release Notes* for memory required for the SBC virtual machine. Additional memory is required for the hypervisor (depends on the hypervisor's specific requirements).
Disk Space	Each SBC virtual machine requires at least 20 GB. Additional storage is required for the hypervisor (depends on the hypervisor's specific requirements).
Network Interfaces	2 or more. For SR-IOV support, 10-GbE NICs with Intel chipset (such as Intel 82599 and X710) must be installed.

The host server should have one of the following hypervisors installed on it:

- **VMWare:** VMware ESXi Version 6.5 or later. For instructions on installing VMware vSphere ESXi, see www.vmware.com.
- **KVM:** Linux version 2.6.32 or later, with KVM/QEMU. For instructions on installing KVM, refer to your Linux distribution's documentation.
- **OpenStack:** Release Juno or later. For instructions on installing OpenStack, see https://docs.openstack.org.
- **Hyper-V:** Microsoft Server 2016 or later (Windows Server 2019 is recommended). For instructions on installing Microsoft Hyper-V, see the *Hyper-V Getting Started Guide* at http://technet.microsoft.com.

Note:

- The VMware vSphere ESXi / Linux KVM / Microsoft Hyper-V are 'bare-metal' hypervisors installed directly on the physical server.
- Mediant VE SBC does not support VMware Workstation and nested virtualization solutions.

2.2 Mediant VE SBC Virtual Machine

The number of virtual CPUs and memory required for the SBC virtual machine is specified in the *Release Notes*.

Notes:

- Transcoding functionality is software-based. There is no need for dedicated hardware except for adding more vCPUs. The transcoding capacity is linear with the number of vCPUs allocated for transcoding.
- Enabling transcoding functionality requires new License Key.

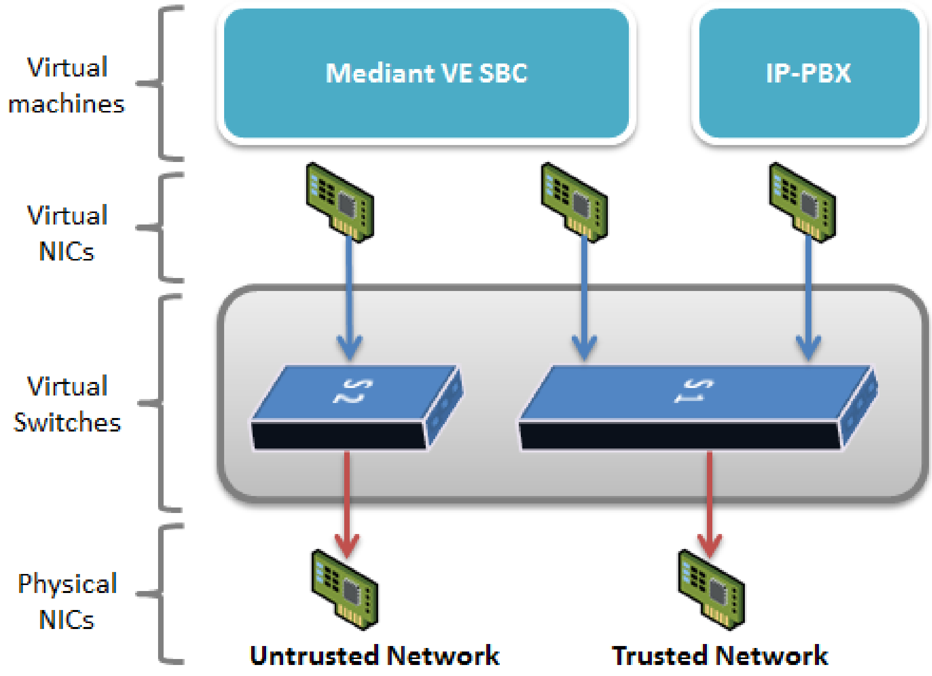
Table 2-2: Mediant VE SBC Virtual Machine Specifications

Resource	Specifications
Disk space	At least 20 GB
Virtual Network Interfaces	Two vNICs are recommended (for trusted / untrusted traffic), an additional vNIC is recommended for HA configurations

2.3 Virtual Networking Configuration

The virtual machine hypervisor should be pre-configured with two virtual networks designated for trusted and untrusted network traffic correspondingly.

Figure 2-1: Virtual Networking Configuration



Note: It is recommended that the physical NICs used by the Mediant SBC VE virtual machine do not share traffic with other applications such as other virtual machines or the hypervisor itself.

This page is intentionally left blank.

3 Installing the Mediant VE SBC

3.1 Configuring the Server's BIOS

For optimal performance, the server's BIOS settings should be configured as described in the procedure below:

➢ **To optimize the server BIOS settings:**

1. Enter the server BIOS (see your server’s documentation for details).
2. Set 'Power Management' to **Maximum Performance** (usually under 'Power' options).
3. Enable 'Intel Turbo Boost' (usually under the 'Processor' options).
4. Enable 'Intel Virtualization Technology VT-d' and 'VT-x' (usually under 'Processor' options).
5. Disable 'Hyper-Threading' (usually under 'Processor' options):
   - **VMWare Hypervisor:** you can keep 'Hyper-Threading' enabled.
   - **KVM Hypervisor:** you can keep 'Hyper-Threading' enabled and configure the virtual machine as described in Section 3.4.9.
   - **OpenStack:** you can keep 'Hyper-Threading' enabled and configure the virtual machine as described in Section 3.4.3.4.
6. For enabling SR-IOV:
   - The SR-IOV support must be enabled in the BIOS (under BIOS 'Advanced Options').
   - The VT-d support must be enabled in the BIOS (under BIOS 'CPU setting').

3.2 Installing Mediant VE SBC on VMware vSphere ESXi

This section describes how to install Mediant VE SBC on VMware vSphere ESXi 6.5 and later, using the VMware vSphere client. The installation process might differ for other hypervisor versions and installation methods (e.g., vCenter).

➢ **To install Mediant VE SBC on VMware:**

1. Deploy the OVF Template (see Section 3.2.1).
2. Adjust the deployed virtual machine (see Section 3.2.2)
3. Start the Mediant VE SBC (see Section 3.2.3)
4. Reconfigure the default IP address to match your network settings (see Section 3.7).

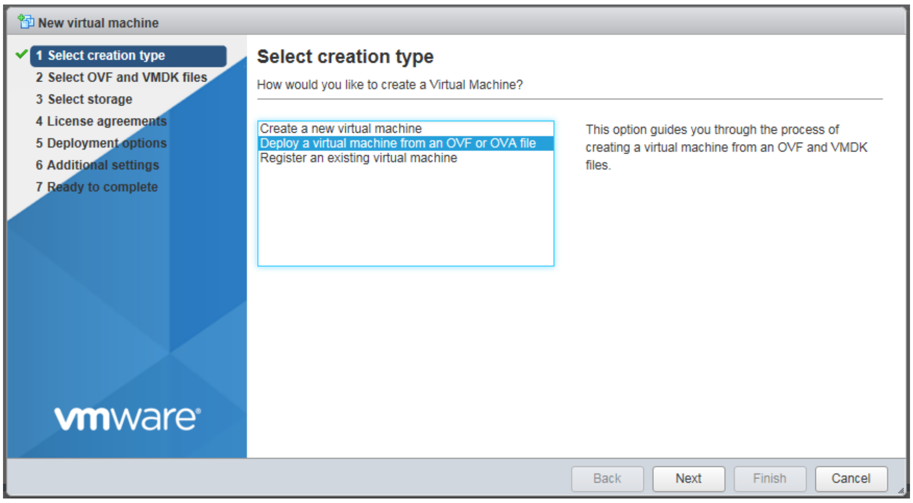
3.2.1 Deploying the OVF Template File

Mediant VE SBC is distributed in the form of an Open Virtualization Format (OVF) file which you must deploy.

➢ **To deploy the OVF file:**

1. Log into the VMware Web client.
2. In the Navigator pane, click **Virtual Machines**, and then in the right pane, click **Create / Register VM**; the 'New virtual machine' wizard appears. From the drop-down list, select **Deploy a virtual machine from an OVF or OVA file**:

Figure 3-1: Deploying the OVF Template – Stage 1 Select Creation Type



New virtual machine

Select creation type

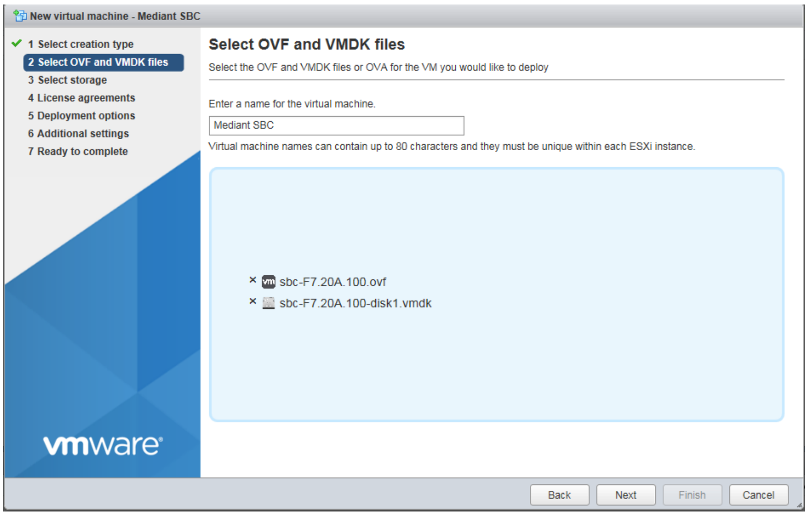
How would you like to create a Virtual Machine?

Create a new virtual machine
Deploy a virtual machine from an OVF or OVA file
Register an existing virtual machine

This option guides you through the process of creating a virtual machine from an OVF and VMDK files.

3. Click **Next**.

Figure 3-2: Deploying the OVF Template – Stage 2 Select OVF and VMDK Files



Select OVF and VMDK files

Select the OVF and VMDK files or OVA for the VM you would like to deploy

Enter a name for the virtual machine.

Mediant SBC

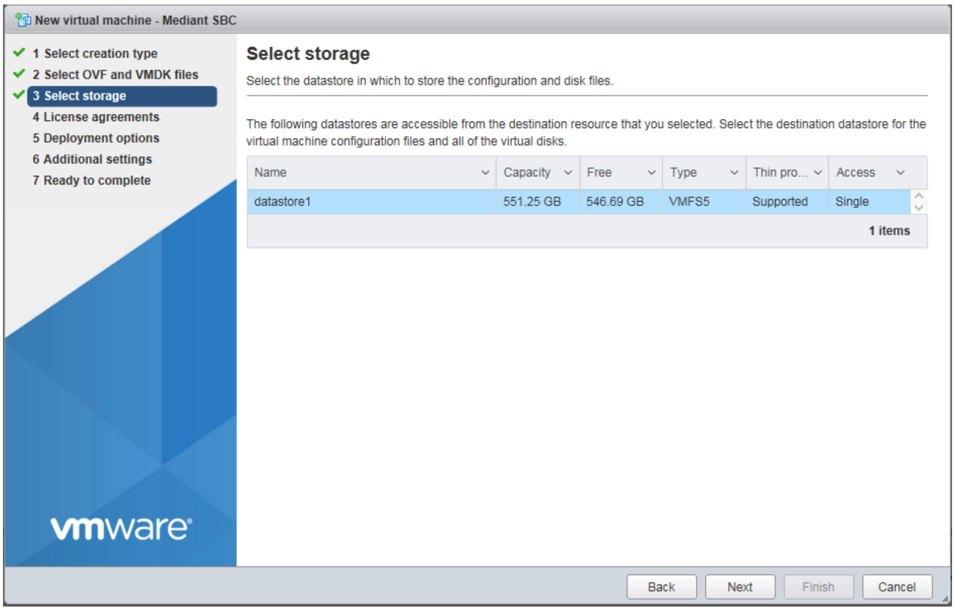
Virtual machine names can contain up to 80 characters and they must be unique within each ESXi instance.

× sbc-F7.20A.100.ovf

× sbc-F7.20A.100-disk1.vmdk

4. Enter a name for the virtual machine, and then select or drag-and-drop the OVF template file supplied by AudioCodes.
5. Click **Next**.

Figure 3-3: Deploying the OVF Template – Stage 3 Select Storage



Select storage

Select the datastore in which to store the configuration and disk files.

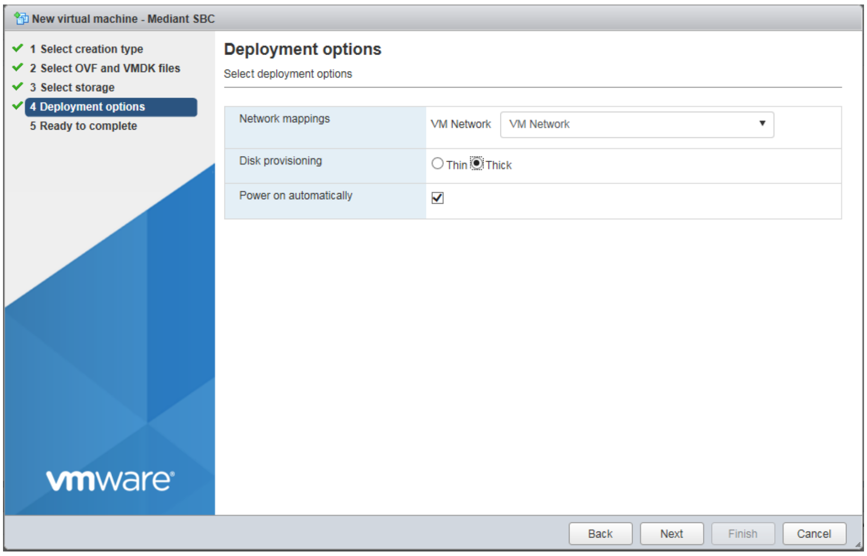
The following datastores are accessible from the destination resource that you selected. Select the destination datastore for the virtual machine configuration files and all of the virtual disks.

Name	Capacity	Free	Type	Thin pro...	Access
datastore1	551.25 GB	546.69 GB	VMFS5	Supported	Single

1 items

6. Verify the OVF details, and then click **Next**.

Figure 3-4: Deploying the OVF Template – Stage 4 Deployment Options



New virtual machine - Mediant SBC

- ✓ 1 Select creation type
- ✓ 2 Select OVF and VMDK files
- ✓ 3 Select storage
- ✓ 4 Deployment options
- 5 Ready to complete

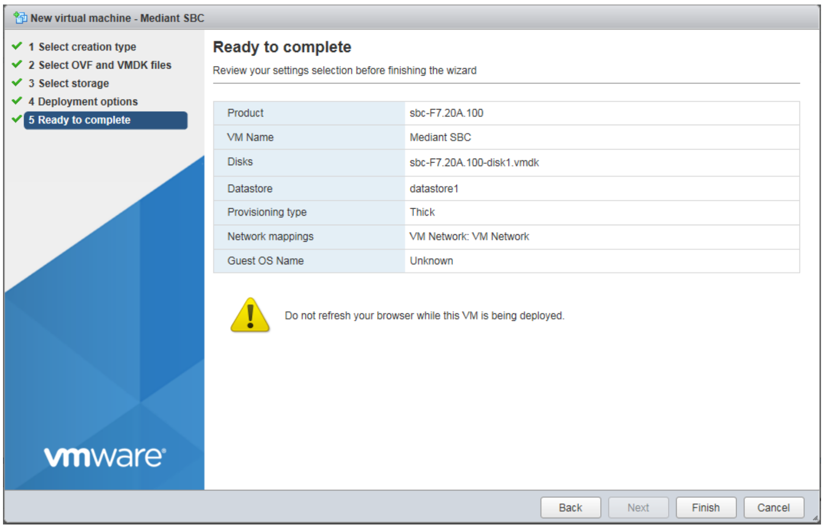
Deployment options

Select deployment options

Network mappings	VM Network	VM Network ▾
Disk provisioning	○ Thin ● Thick	
Power on automatically	☑	

7. Do the following:
   a. Select the destination network(s) to which two of the SBC virtual Network Interface Cards will be connected. Note that the destination network(s) name(s) depend on VMware host configuration. The OVF template provides the virtual machine with two NICs of type VMXNET3. After installing the SBC virtual machine, you can change the number of network connections and/or their type (see Appendix A, Configuring the Network, on page 61).
   b. Select the **Thick** option for disk provisioning.
8. Click **Next**.

Figure 3-5: Deploying the OVF Template – Stage 5 Ready to Complete



New virtual machine - Mediant SBC

- ✓ 1 Select creation type
- ✓ 2 Select OVF and VMDK files
- ✓ 3 Select storage
- ✓ 4 Deployment options
- ✓ 5 Ready to complete

Ready to complete

Review your settings selection before finishing the wizard

Product	sbc-F7.20A.100
VM Name	Mediant SBC
Disks	sbc-F7.20A.100-disk1.vmdk
Datastore	datastore1
Provisioning type	Thick
Network mappings	VM Network: VM Network
Guest OS Name	Unknown

Do not refresh your browser while this VM is being deployed.

9. Click **Finish** and wait for the deployment process to complete.
10. Adjust the deployed virtual machine as described in Section 3.2.2.

3.2.2 Adjusting the Virtual Machine to Chosen Mediant VE SBC

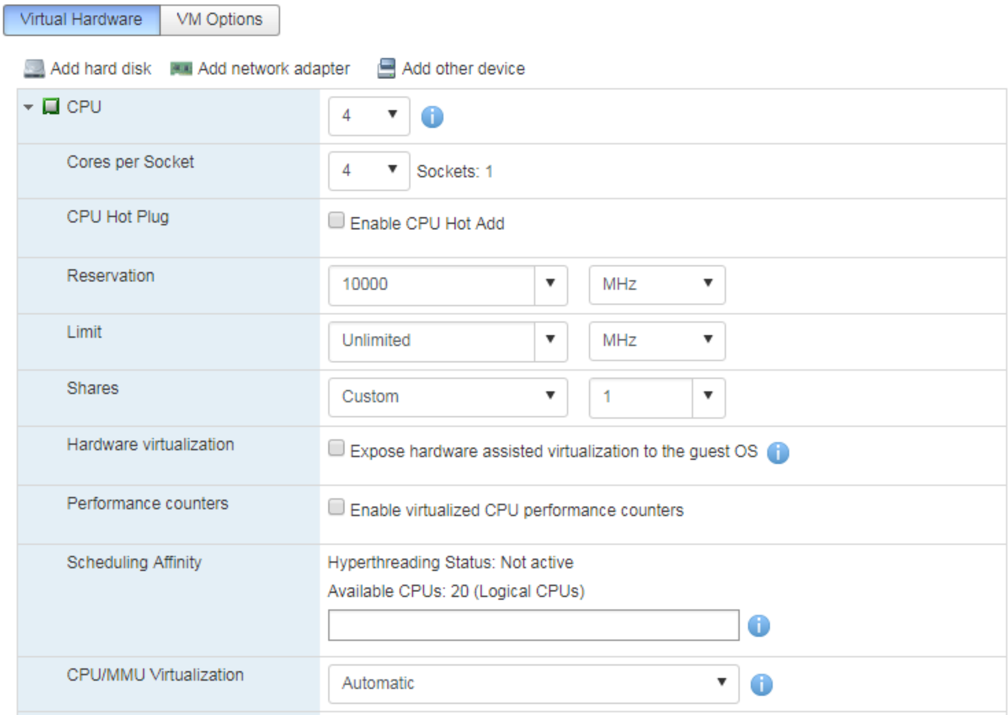
This section shows how to adjust the deployed virtual machine for the chosen Mediant VE SBC flavor.

➢ To adjust deployed virtual machine for Mediant VE SBC:

1. In the Navigator pane, click **Virtual Machines**, locate the new virtual machine in the list, right-click it, and then choose **Edit settings;** the Edit settings screen opens.
2. Click the **Virtual Hardware** tab:
   a. Expand the **CPU** folder, and do the following:
      - Configure the 'Cores per Socket' parameter to the same value as the 'CPU' parameter. When configuration is set correctly, "Sockets: 1" should be displayed to the right of the value (as shown in the figure below).
      - Configure 'Reservation' of CPU frequency to the maximum MHz value to ensure that full physical CPU cores will be reserved for the Mediant VE SBC virtual machine. For example, for Intel® Xeon™ Gold 6126 with a core frequency of 2.6 GHz, in order to reserve four CPUs, the reservation should be set to the maximum allowed (i.e., ~10 GHz).
      - From the 'Limit' drop-down list, select **Unlimited**.
      - If 'HyperThreading Status' is "Active" and the required number of virtual CPUs is higher than 1, there are two options to configure the VM:
- If the Host is using a CPU **prior to Intel® Xeon® Scalable Processors**, then set the 'Scheduling Affinity' field to ‘0, 2, 4, 6’ or any other sequence of even cores indexes such as ‘2, 4, 6, 8’, as shown in the figure above.
- If the Host is using a CPU **Intel® Xeon® Scalable Processors or later**, then set the 'Scheduling Affinity' field to ‘0-3’ (for 4 vCPUs) or any other successive range starting with even number such as ‘8-11’, according to the number of vCPUs required. In addition, the INI file parameter CPUOverrideHT should be configured to 1.

        If 'HyperThreading Status' is "Not active", leave the 'Scheduling Affinity' field empty.

Figure 3-6: Adjusting Virtual Machine for SBC – Virtual Hardware – CPU

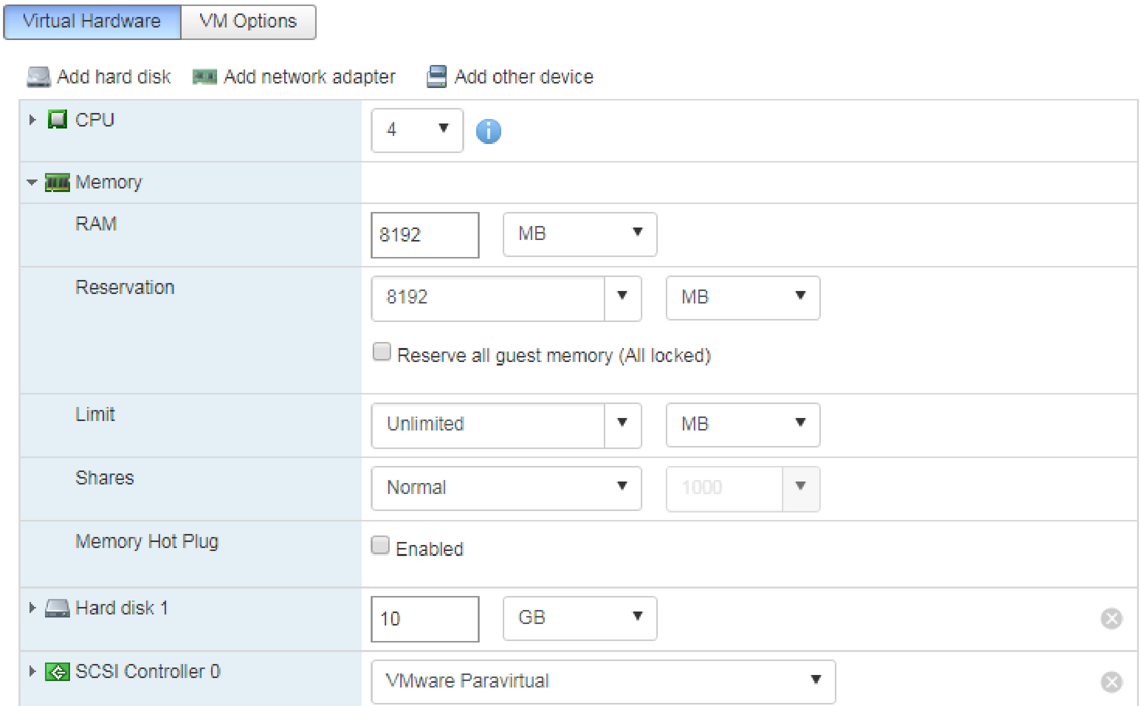


CPU	4 ▾
Cores per Socket	4 ▾ Sockets: 1
CPU Hot Plug	□ Enable CPU Hot Add
Reservation	10000 ▾ MHz ▾
Limit	Unlimited ▾ MHz ▾
Shares	Custom ▾ 1 ▾
Hardware virtualization	□ Expose hardware assisted virtualization to the guest OS
Performance counters	□ Enable virtualized CPU performance counters
Scheduling Affinity	Hyperthreading Status: Not active Available CPUs: 20 (Logical CPUs)
CPU/MMU Virtualization	Automatic ▾

b. Expand the **Memory** folder**:** and adjust the memory reservation according to the chosen configuration flavor.

- Configure 'Reservation' of memory to the maximum value allowed (minimum configuration of the chosen Mediant VE SBC flavor).
- Select the 'Unlimited' check box if it isn't already selected.

Figure 3-7: Adjusting Virtual Machine SBC – Virtual Hardware - Memory



CPU	4	
Memory		
RAM	8192	MB
Reservation	8192 □ Reserve all guest memory (All locked)	MB
Limit	Unlimited	MB
Shares	Normal	1000
Memory Hot Plug	□ Enabled	
Hard disk 1	10	GB
SCSI Controller 0	VMware Paravirtual	

3. Click **Save**.

3.2.3 Starting Mediant VE SBC

This section shows how to start Mediant VE SBC.

➢ To start Mediant VE SBC:

1. In the Navigator pane, click **Virtual Machines**.
2. Locate the new virtual machine in the list, right-click it, and then from the shortcut menu, choose **Power** > **Power on**.
3. Proceed to Section 3.7.

3.3 Installing Mediant VE SBC on KVM

This section shows how to install Mediant VE SBC on KVM. KVM is a virtualization solution integrated into Linux kernel and available on different Linux distributions. In the instructions below, CentOS 7 distribution is used as 'Linux host'. The installation process might differ for other distributions (e.g., Ubuntu).

➢ **To install Mediant VE SBC on KVM:**

1. Adjust Linux host's settings for optimal performance (see Section 3.3.1).
2. Install and configure Open vSwitch on Linux host (see Section 3.4)
3. Deploy the QCOW2 Image (see Section 3.4.8).
4. Reconfigure the default IP address to match your network settings (see Section 3.7).

3.3.1 Adjusting Linux Host's Settings

This section shows how to adjust the Linux host's settings for an optimal Mediant VE SBC deployment.

➢ **To adjust the Linux host's settings:**

1. Configure Linux host to use "virtual host" operation profile.

```
[root@virt-host ~]# tuned-adm profile virtual-host
```

2. Determine the names of all the available network interfaces.

```
[root@virt-host ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
qlen 1000
    link/ether d8:d3:85:12:34:01 brd ff:ff:ff:ff:ff:ff
    inet 10.4.219.60/16 brd 10.4.255.255 scope global enp2s0f0
       valid_lft forever preferred_lft forever
    inet6 fe80::dad3:85ff:feb9:eb50/64 scope link
       valid_lft forever preferred_lft forever
3: eno2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen
1000
    link/ether d8:d3:85:12:34:02 brd ff:ff:ff:ff:ff:ff
4: eno3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen
1000
    link/ether d8:d3:85:12:34:03 brd ff:ff:ff:ff:ff:ff
4: eno4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen
1000
    link/ether d8:d3:85:12:34:04 brd ff:ff:ff:ff:ff:ff
```

3. For each network interface, determine the maximum supported ring size.

```
[root@virt-host ~]# ethtool -g eno1
Ring parameters for eno1:
Pre-set maximums:
RX:             4096
RX Mini:        0
RX Jumbo:       0
TX:             4096
```

```
Current hardware settings:
RX:             256
RX Mini:        0
RX Jumbo:       0
TX:             256
```

4. Configure each network interface to use the maximum supported ring size (adjust the command below based on the previous command’s output).

```
[root@virt-host ~]# echo 'ETHTOOL_OPTS="-G eno1 rx 4096 tx
4096"' >> /etc/sysconfig/network-scripts/ifcfg-eno1
```

5. Restart the network service to apply the new configuration.

```
[root@virt-host ~]# systemctl restart network
```

3.4 Installing Mediant VE SBC on OpenStack

This section describes the installation process of Mediant VE SBC on OpenStack cloud computing platform.

➢ **To install Mediant VE SBC on OpenStack:**

1. Create a flat provider network/s (see Section 3.4.1).
2. Create a Mediant VE SBC image (see Section 3.4.2).
3. Create Mediant VE SBC flavors (see Section 3.4.3).
4. Create a Mediant VE SBC security group (see Section 3.4.4).
5. Deploy the Mediant VE SBC instance (see Section 3.4.5).

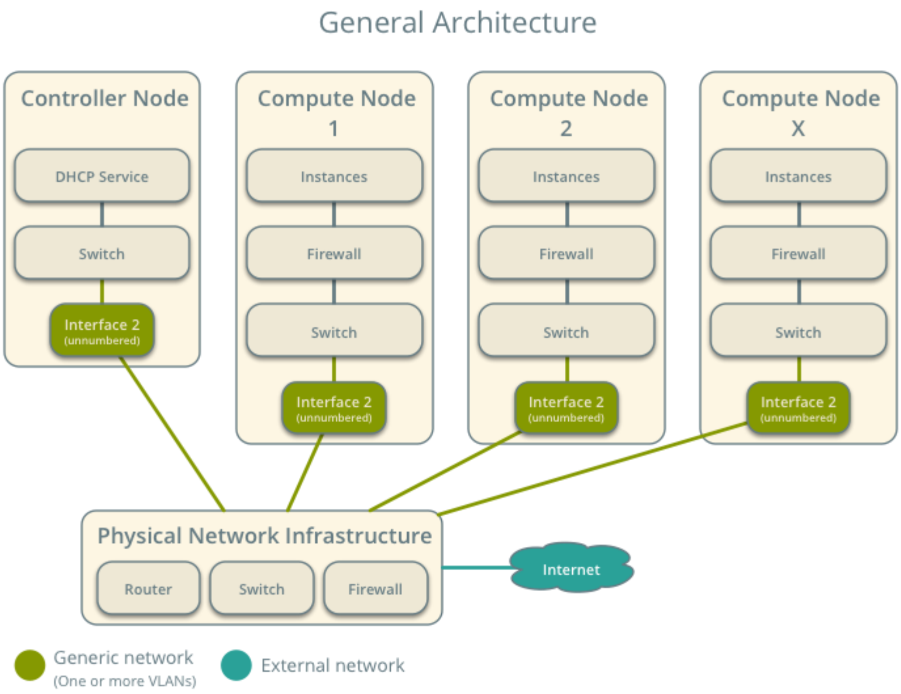
The examples below use OpenStack CLI and adhere to the syntax of the OpenStack Mitaka release. Earlier / future OpenStack releases may have slightly different CLI syntax. It is also possible to perform most of the tasks using Dashboard (horizon) graphical user interface.

Note: For optimal Mediant VE SBC performance, make sure that OpenStack compute nodes on which the Mediant VE SBC will be deployed have the 'Power Management' setting in BIOS configured for **Maximum Performance** (as described in Section 2.1)

3.4.1 Creating a Flat Provider Network

For optimal deployment of Mediant VE SBC, it is recommended to use flat provider networks. Flat provider networks use a physical network infrastructure to handle switching and routing of network traffic. Instances (virtual machines) are connected to the physical network infrastructure through Layer-2 bridging / switching. Use of flat provider networks improve network performance and simplify interworking between OpenStack instances and physical equipment at the datacenter.

Figure 3-8: General Architecture of OpenStack Flat Provider Network



➢ **To create a flat provider network:**

1. Create a flat provider network for the first network interface (**physnet1**). Adjust IP addresses to match your setup and network topology.

```
$ neutron net-create net1 --shared --provider:network_type
flat --provider:physical_network physnet1
$ neutron subnet-create --name net1_subnet --gateway 10.3.0.1
--dns-nameserver 10.1.1.11 --allocation-pool
start=10.3.160.2,end=10.3.160.10 net1 10.3.0.0/16
```

2. Create additional flat provider networks as needed.

3.4.2 Creating a Mediant VE SBC Image

Mediant VE SBC is distributed as a QCOW2 image file. This file must be imported into the OpenStack image repository (glance).

➢ **To create a Mediant VE SBC image:**

1. Copy the Mediant VE SBC QCOW2 image file to the OpenStack controller node.
2. Create a Mediant VE SBC image:

```
$ glance image-create --name sbc-F7.40A.005.505 --disk-format
qcow2 --container-format bare --file sbc-F7.40A.005.505.qcow2
```

3.4.3 Creating Mediant VE SBC Flavors

Create three flavors for different sizes of Mediant VE SBC instances using the commands below. Refer to the *Release Notes* for the detailed sizing and capacity of supported Mediant VE SBC flavors.

➢ **To create Mediant VE SBC flavors:**

1. Create an **sbc.small** flavor (1 CPU, 2-GB RAM):

```
$ nova flavor-create sbc.small auto 2048 20 1
```

2. Create an **sbc.large** flavor (4 CPU, 16-GB RAM):

```
$ nova flavor-create sbc.large auto 16384 20 4
```

3. Create an **sbc.xlarge** flavor (8 CPU, 32-GB RAM):

```
$ nova flavor-create sbc.xlarge auto 32678 20 8
```

3.4.3.1 Binding Mediant VE SBC Instances to Physical CPU Cores

Configure a "dedicated" CPU policy for Mediant VE SBC flavors in order to achieve optimal performance of Mediant VE SBC instances:

```
$ nova flavor-key sbc.small set hw:cpu_policy=dedicated
$ nova flavor-key sbc.large set hw:cpu_policy=dedicated
$ nova flavor-key sbc.xlarge set hw:cpu_policy=dedicated
```

3.4.3.2 Configuring Virtual CPU Topology

Configure **sbc.large** and **sbc.xlarge** flavors to expose single socket CPU topology for optimal performance of Mediant VE SBC instances:

```
$ nova flavor-key sbc.large set hw:cpu_sockets=1
$ nova flavor-key sbc.xlarge set hw:cpu_sockets=1
```

3.4.3.3 Adjusting to Compute Nodes with Multiple CPU Sockets

If OpenStack compute nodes have multiple CPU sockets, configure **sbc.large** and **sbc.xlarge** flavors to run on a single NUMA node in order to achieve optimal performance of Mediant VE SBC instances:

```
$ nova flavor-key sbc.large set hw:numa_nodes=1
$ nova flavor-key sbc.xlarge set hw:numa_nodes=1
```

3.4.3.4 Adjusting to Compute Nodes with Hyper-Threading Enabled

If OpenStack compute nodes have Hyper-Threading enabled, configure Mediant VE SBC flavors to utilize “isolate” thread policy in order to achieve optimal performance of Mediant VE SBC instances:

```
$ nova flavor-key sbc.small set hw:cpu_thread_policy=isolate
$ nova flavor-key sbc.large set hw:cpu_thread_policy=isolate
$ nova flavor-key sbc.xlarge set hw:cpu_thread_policy=isolate
```

3.4.4 Creating a Mediant VE SBC Security Group

Create a security group for Mediant VE SBC instances.

The example below creates a security group that permits the following incoming traffic:

- TCP port 80 (HTTP)
- TCP port 443 (HTTPS)
- TCP port 22 (SSH)
- UDP port 5060 and 5061 (SIP)
- UDP ports 6000-65535 (RTP)

Adjust the rules (e.g. SIP and RTP protocol and ports) to match your deployment topology:

```
$ nova secgroup-create sbc "Mediant VE SBC security group"
$ nova secgroup-add-rule sbc tcp 80 80 0.0.0.0/0
$ nova secgroup-add-rule sbc tcp 443 443 0.0.0.0/0
$ nova secgroup-add-rule sbc tcp 22 22 0.0.0.0/0
$ nova secgroup-add-rule sbc udp 5060 5060 0.0.0.0/0
$ nova secgroup-add-rule sbc udp 6000 65535 0.0.0.0/0
```

3.4.5 Deploying the Mediant VE SBC Instance

Deploy the Mediant VE SBC instance by running the command below. Adjust the flavor type according to the required instance sizing.

```
$ nova boot --image sbc-F7.40A.005.505 --nic net-name=net1 --
security-groups sbc --flavor sbc.small sbc1
```

Mediant VE SBC automatically detects the OpenStack network topology and configures its network interfaces accordingly. This should enable you to connect to the Mediant VE SBC instance’s management interface (Web or CLI) without any further configuration.

The automatic configuration process relies on the DHCP and MediaData services provided by OpenStack and may be extended and customized as per environment / customer needs. Refer to the *Automatic Provisioning of Mediant VE SBC via Cloud-Init* for detailed information.

3.4.6 Using SR-IOV Network Interfaces

SR-IOV is PCI Special Interest Group (PCI-SIG) specification for virtualizing network interfaces, representing each physical resource as a configurable entity (called a PF for Physical Function), and creating multiple virtual interfaces (VFs or Virtual Functions) with limited configurability on top of it. Use of SR-IOV significantly improves network performance by pushing network-related tasks down to the hardware layer, off-loading them from the hypervisor and virtual switch.

Note: SR-IOV support by Mediant VE SBC requires NIC controllers that support the Virtual Function of Linux drivers ixgbe and i40e. Controllers that have been tested and approved by AudioCodes include the Intel 82599 NIC and Intel X710 NIC. For additional NIC controller support, please contact your AudioCodes sales representative.

Refer to the upstream OpenStack documentation or to the documentation of the vendor-specific OpenStack distribution for detailed instructions on how to configure OpenStack to utilize SR-IOV interfaces. For example:

- https://docs.openstack.org/mitaka/networking-guide/config-sriov.html
- https://content.mirantis.com/MOS-7-NFVI-Whitepaper-Landing-Page.html

To launch the Mediant VE SBC instance into the SR-IOV network, you must create a network port with “direct” binding type and specify it during the instance creation. For example:

```
$ neutron port-create --name sriov_port --binding:vnic_type direct
net1
Created a new port:
+-----------------------+-------------------------------------------------------------------------------------+
| Field                 | Value                                                                               |
+-----------------------+-------------------------------------------------------------------------------------+
| admin_state_up        | True                                                                                |
| allowed_address_pairs |                                                                                     |
| binding:host_id       |                                                                                     |
| binding:profile       | {}                                                                                  |
| binding:vif_details   | {}                                                                                  |
| binding:vif_type      | unbound                                                                             |
| binding:vnic_type     | direct                                                                              |
| created_at            | 2017-02-28T09:02:30Z                                                                |
| description           |                                                                                     |
| device_id             |                                                                                     |
| device_owner          |                                                                                     |
| extra_dhcp_opts       |                                                                                     |
| fixed_ips             | {"subnet_id": "3f29fbc8-1939-4a72-a2d6-3a5711775e9e", "ip_address": "10.10.131.47"} |
| id                    | 876bc41a-d754-4bc8-8f4f-3e3a7f77db31                                                |
| mac_address           | fa:16:3e:9d:fd:7b                                                                   |
| name                  | sriov_port                                                                          |
| network_id            | 93aef71e-3771-43d1-9d4b-458658046412                                                |
| project_id            | aa24ff13d26a4b5fb4ab69967fe2377e                                                    |
| revision_number       | 5                                                                                   |
| security_groups       | f30c4212-df0d-4276-92a6-91fdd9fffcbe                                                |
| status                | DOWN                                                                                |
| tenant_id             | aa24ff13d26a4b5fb4ab69967fe2377e                                                    |
| updated_at            | 2017-02-28T09:02:30Z                                                                |
+-----------------------+-------------------------------------------------------------------------------------+

$ port_id=876bc41a-d754-4bc8-8f4f-3e3a7f77db31
$ nova boot --image sbc-F7.40A.005.505 --nic port-id=$port_id --
security-groups sbc --flavor sbc.xlarge sbc1
```

3.4.7 Installing and Configuring Networking on Linux Host

The virtual network interfaces associated with the Mediant VE SBC virtual machine can be either SR-IOV virtual function or Open vSwitch virtual NICs. A virtual network interface using SR-IOV provides superior performance relative to Open vSwitch virtual interface and thus, is more suitable for media network interfaces. It is possible to mix both types of interfaces (SR-IOV and Open vSwitch) and have multiple virtual interfaces of each type.

The following subsections describe the installation and configuration required for each type of virtual network interface.

3.4.7.1 Installing and Configuring Open vSwitch on Linux Host

Open vSwitch is an open-source virtual switch used by the default OpenStack Neutron implementation. It provides better performance for the Mediant VE SBC traffic load than alternative bridging implementations, for example, MacVTap and is therefore, recommended for use even for "pure" KVM deployments without OpenStack.

➢ **To install and configure Open vSwitch on the Linux host:**

1. Install Open vSwitch from the RDO repository:

```
[root@virt-host ~]# yum install centos-release-openstack-vicrotia
[root@virt-host ~]# yum install openvswitch
```

2. Start Open vSwitch:

```
[root@virt-host ~]# systemctl enable openvswitch.service
[root@virt-host ~]# systemctl start openvswitch.service
```

3. Configure bridges and attach them to network interfaces (in the example below, two bridges are configured and eno2 and eno3 interfaces are attached to them; adjust it to match you deployment's requirements).

```
[root@virt-host ~]# ovs-vsctl add-br br-ex1
[root@virt-host ~]# ovs-vsctl add-br br-ex2
[root@virt-host ~]# ovs-vsctl add-port br-ex1 eno2
[root@virt-host ~]# ovs-vsctl add-port br-ex2 eno3
```

4. Verify the Open vSwitch configuration:

```
[root@virt-host ~]# ovs-vsctl show
9f724f32-65e2-4ce9-829f-91a41ab09c40
    Bridge "br-ex1"
        Port "br-ex1"
            Interface "br-ex1"
                type: internal
        Port "eno2"
            Interface "eno2"
    Bridge "br-ex2"
        Port "br-ex2"
            Interface "br-ex2"
                type: internal
        Port "eno3"
            Interface "eno3"
    ovs_version: "2.13.4"
```

5. Make Open vSwitch configuration persistent by editing network configuration scripts:

```
/etc/sysconfig/network-scripts/ifcfg-br-ex1
DEVICE=br-ex1
ONBOOT=yes
```

```
DEVICETYPE=ovs
TYPE=OVSBridge
HOTPLUG=no
USERCTL=no

/etc/sysconfig/network-scripts/ifcfg-br-ex2
DEVICE=br-ex2
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSBridge
HOTPLUG=no
USERCTL=no

/etc/sysconfig/network-scripts/ifcfg-eno2
DEVICE=eno2
ONBOOT=yes
IPV6INIT=no
BOOTPROTO=none
NAME=eno2
DEVICETYPE=ovs
TYPE=OVSPort
OVS_BRIDGE=br-ex1
ETHTOOL_OPTS="-G eno2 rx 4096 tx 4096"

/etc/sysconfig/network-scripts/ifcfg-eno3
DEVICE=eno3
ONBOOT=yes
IPV6INIT=no
BOOTPROTO=none
NAME=eno3
DEVICETYPE=ovs
TYPE=OVSPort
OVS_BRIDGE=br-ex2
ETHTOOL_OPTS="-G eno3 rx 4096 tx 4096"
```

3.4.7.2 Installing and Configuring SR-IOV on Linux Host

SR-IOV (single root input/output virtualization) is a network interface that allows the isolation of the PCI Express resources for manageability and performance reasons. A physical PCI Express resource can be shared on a virtual environment using the SR-IOV specification. The SR-IOV offers different virtual functions to different virtual components (e.g. network adapter) on a physical server machine. A NIC supporting SR-IOV allows the virtual machine to share the NIC resources by accessing through virtual function exposed by the NIC's physical function.

> **Note:** SR-IOV support by Mediant VE SBC requires NIC controllers that support the Virtual Function of Linux drivers ixgbe and i40e. Controllers that have been tested and approved by AudioCodes include the Intel 82599 NIC and Intel X710 NIC. For additional NIC controller support, please contact your AudioCodes sales representative.

➢ **To install and configure SR-IOV on the Linux host:**

1. For supporting SR-IOV, the IOMMU should be enabled on the host's kernel, and selinux must be disabled, by changing the GRUB linux command line

   ```
   [root@virt-host ~]# vi /etc/sysconfig/grub
   ```

   Add to GRUB_CMDLINE_LINUX: intel_iommu=on AND selinux=0, for example:

   ```
   GRUB_TIMEOUT=5
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_TERMINAL_OUTPUT="console"
   GRUB_CMDLINE_LINUX="rd.lvm.lv=centos/root
   rd.lvm.lv=centos/swap crashkernel=auto rhgb quiet
   intel_iommu=on selinux=0"
   GRUB_DISABLE_RECOVERY="true"
   ```

2. Regenerate grub.cfg:

   ```
   [root@virt-host ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

3. Disable SELInux:

   ```
   [root@virt-host ~]# vi /etc/sysconfig/selinux
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #     enforcing - SELinux security policy is enforced.
   #     permissive - SELinux prints warnings instead of
   enforcing.
   #     disabled - No SELinux policy is loaded.
   SELINUX=disabled
   # SELINUXTYPE= can take one of three two values:
   #     targeted - Targeted processes are protected,
   #     minimum - Modification of targeted policy. Only selected
   processes are protected.
   #     mls - Multi Level Security protection.
   SELINUXTYPE=targeted
   ```

4. Reboot the host.

5. Detect the PCI bus location of the NICs supported SR-IOV (e.g., using the **lspci** command) and the name of the associated physical interface names (e.g., using the **ip addr show** command).

6. Define the number of virtual functions to be created upon boot. For example, add the following line to create 4 virtual functions on the SR-IOV interface enp3s0f0:

   ```
   [root@virt-host ~]# vi /etc/rc.d/rc.local
   ```

   Add the following line:

   ```
   echo 4 > /sys/class/net/enp3s0f0/device/sriov_numvfs
   ```

   Save and exit:

   ```
   [root@virt-host ~]# chmod +x /etc/rc.d/rc.local
   ```

   Configure the physical link to be up in boot:

   ```
   [root@virt-host ~]# vi /etc/sysconfig/network-scripts/ifcfg-
   enp3s0f0
   TYPE=Ethernet
   BOOTPROTO=none
   NAME=enp3s0f0
   DEVICE=enp3s0f0
   ONBOOT=yes
   ETHTOOL_OPTS="-G enp3s0f0 rx 4096 tx 4096"
   USERCTL=no
   ```

7. Reboot the host.

8. Check the PCI bus location of the new created virtual function:

   ```
   [root@virt-host ~]# lspci | grep Eth
   ```

   For example, the bus location on the following virtual function is 03.10.0:

   ```
   03:10.0 Ethernet controller: Intel Corporation 82599 Ethernet
   Controller Virtual Function (rev 01)
   ```

9. Create an XML interface for the virtual function:

   ```
   [root@virt-host ~]# vi new-sriov-connection.xml
      <interface type='hostdev' managed='yes'>
         <source>
           <address type='pci' domain='0x0000' bus='0x03'
   slot='0x10' function='0x0'/>
         </source>
       </interface>
   ```

10. Create a virtual machine for the Mediant VE SBC (if not already exists), as described in Section 3.4.8.

11. Shut down the Mediant VE SBC virtual machine.

12. Attach the new virtual function to the virtual machine:

    ```
    [root@virt-host ~]# virsh attach-device sbc-test ./new-sriov-
    connection.xml  --config
    ```

13. Start the virtual machine:

    ```
    [root@virt-host ~]# virsh start sbc-test
    ```

3.4.8 Deploying the QCOW2 Image

Mediant VE SBC is distributed as a QCOW2 image.

➢ **To deploy the image:**

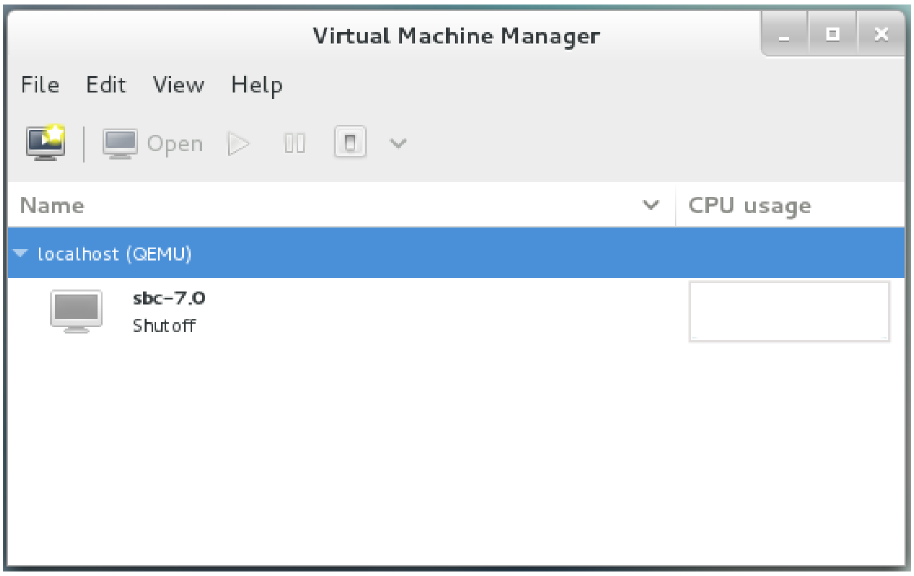
1. Copy QCOW2 image to the standard image repository.

```
[root@virt-host ~]# cp ~admin/sbc-7.00A.004.503.qcow2
/var/lib/libvirt/images/sbc-test.qcow2
[root@virt-host ~]# chown root:root
/var/lib/libvirt/images/sbc-test.qcow2
[root@virt-host ~]# chmod 600 /var/lib/libvirt/images/sbc-
test.qcow2
```

   Remember that KVM will use the image as an actual virtual machine disk, so if you need to create multiple virtual machine instances, create a new copy of the image for each instance.

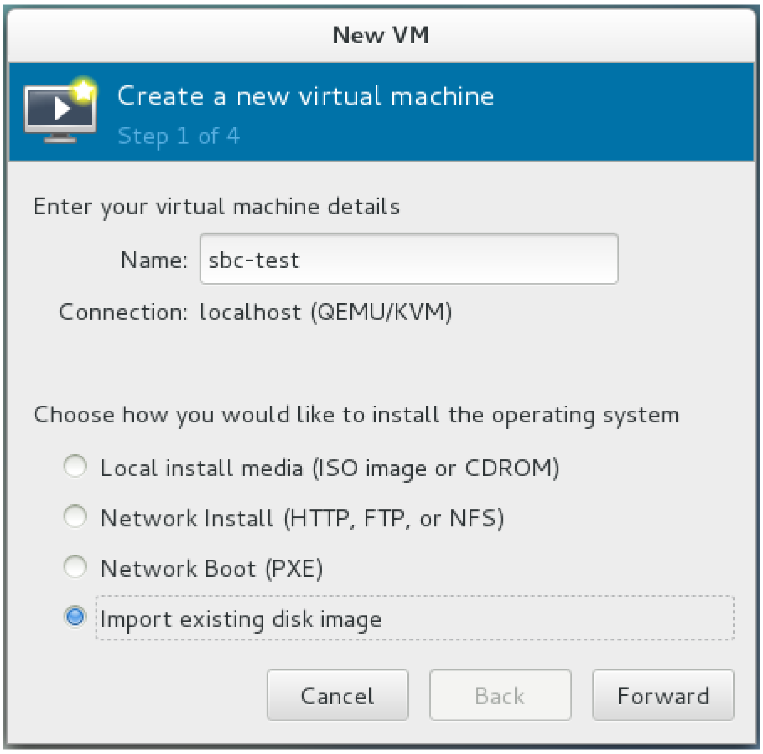
2. Open Virtual Machine Manager:

```
[root@virt-host ~]# virt-manager
```



3. Click the **Create New Virtual Machine** button.

4. Enter the new virtual machine name, select the **Import existing disk image** option, and then click **Forward**.



New VM

Create a new virtual machine
Step 1 of 4

Enter your virtual machine details

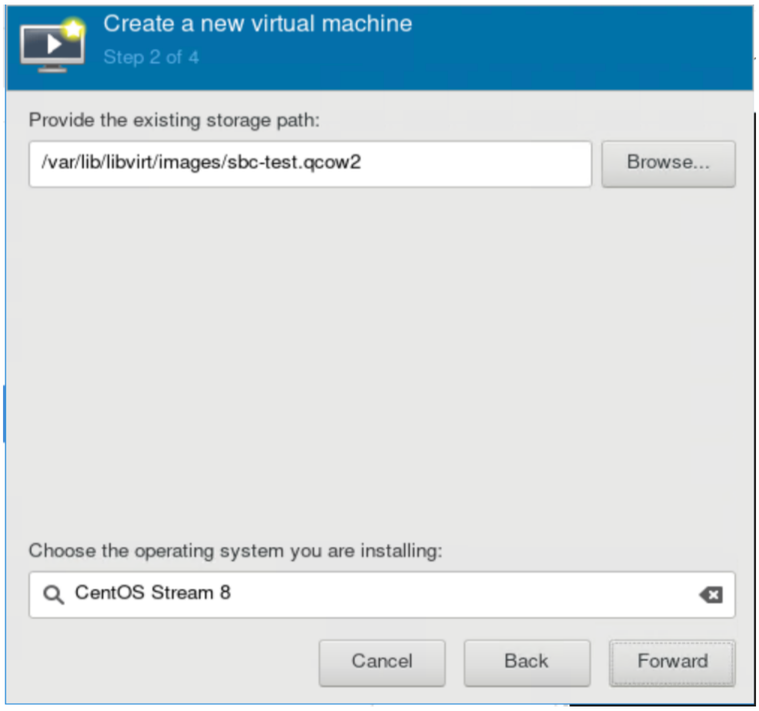
Name: sbc-test

Connection: localhost (QEMU/KVM)

Choose how you would like to install the operating system

○ Local install media (ISO image or CDROM)
○ Network Install (HTTP, FTP, or NFS)
○ Network Boot (PXE)
● Import existing disk image

5. Click **Browse** and select the QCOW2 image. Change 'OS Type' to **Linux** and 'Version' to **Centos Stream 8**. Click **Forward**.



Create a new virtual machine
Step 2 of 4

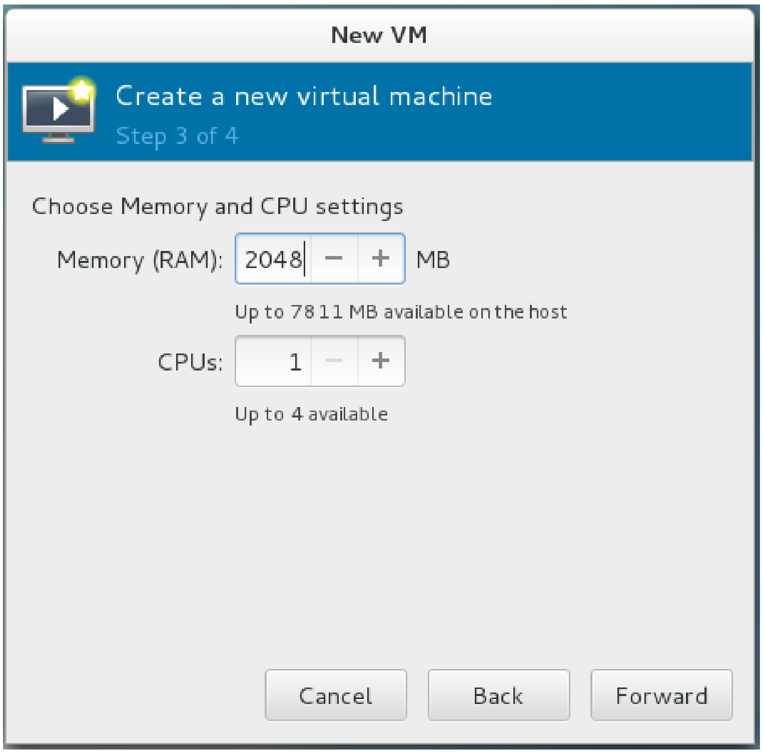
Provide the existing storage path:

/var/lib/libvirt/images/sbc-test.qcow2

Choose the operating system you are installing:

CentOS Stream 8

6. Adjust virtual machine Memory and CPU allocation according to the chosen Mediant VE SBC flavor. Click **Forward**.



Choose Memory and CPU settings

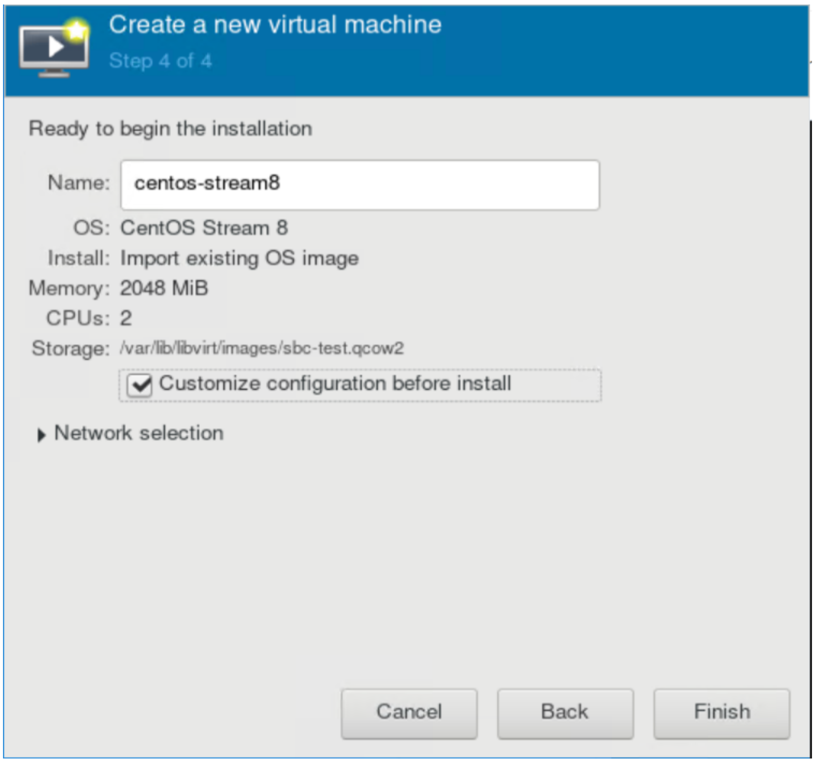
Memory (RAM): 2048 MB

Up to 7811 MB available on the host

CPUs: 1

Up to 4 available

7. Select the **Customize configuration before install** option, and then click **Finish**.



Ready to begin the installation

Name: centos-stream8

OS: CentOS Stream 8

Install: Import existing OS image

Memory: 2048 MiB

CPUs: 2

Storage: /var/lib/libvirt/images/sbc-test.qcow2

☑ Customize configuration before install

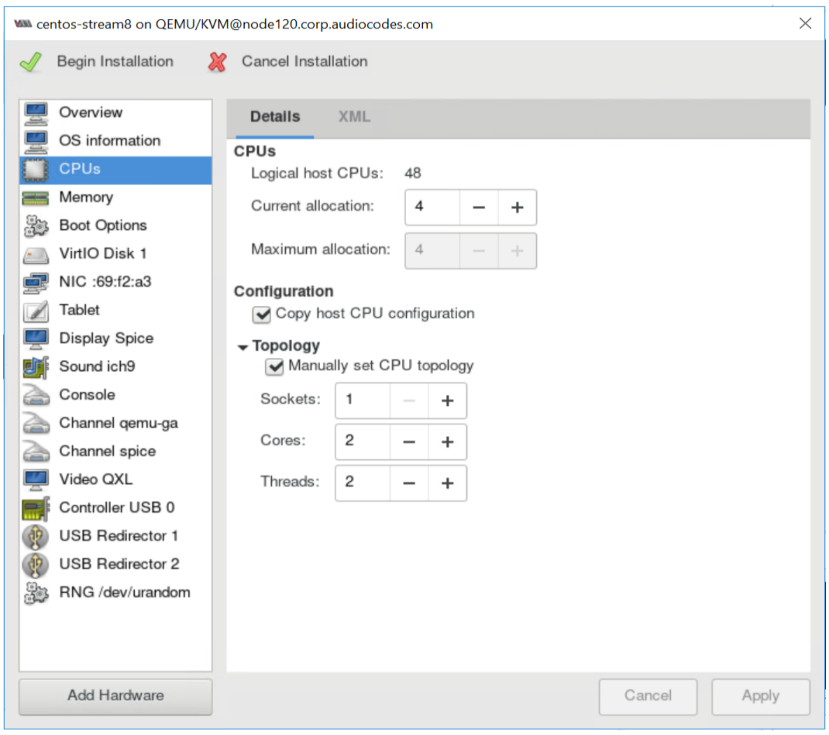
8. In the virtual machine's properties, open the Processor configuration screen. Expand the Configuration section, and then click **Copy host CPU configuration** to enable the Virtual Machine to use a full set of host CPU features.

9. Set the number of CPUs according to the required capacity based on the Capacity table in the Release Notes. The **Current Allocation** should be the same as the **Maximum Allocation**.

10. Click **Topology** and make sure to define the CPU topology according to the chosen configuration from the Capacity table in the Release Notes:

    - If the chosen capacity configuration is not using Hyper-Threading, then set the topology to 1 Socket, 1 Threads, and the number of Cores should be set to the number of allocated CPUs (above).
    - If the chosen capacity configuration is using Hyper-Threading, then set the topology to 1 Socket, 2 Threads, and the number of Cores should be set to the number of CPUs allocated, divided by 2. When using Hyper-Threading topology, make sure the Hyper-Threading Capability is also enabled on the Host server.

    For performance-related settings (including CPU affinity and pinning), see Section 3.12.



centos-stream8 on QEMU/KVM@node120.corp.audiocodes.com

CPUs

Logical host CPUs: 48

Current allocation: 4

Maximum allocation: 4

Configuration

☑ Copy host CPU configuration

Topology

☑ Manually set CPU topology

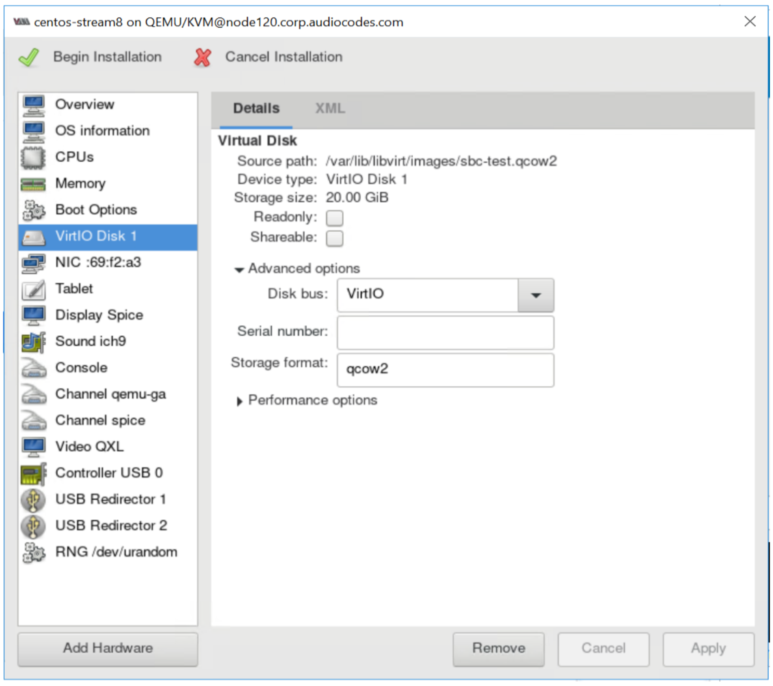
Sockets: 1

Cores: 2

Threads: 2

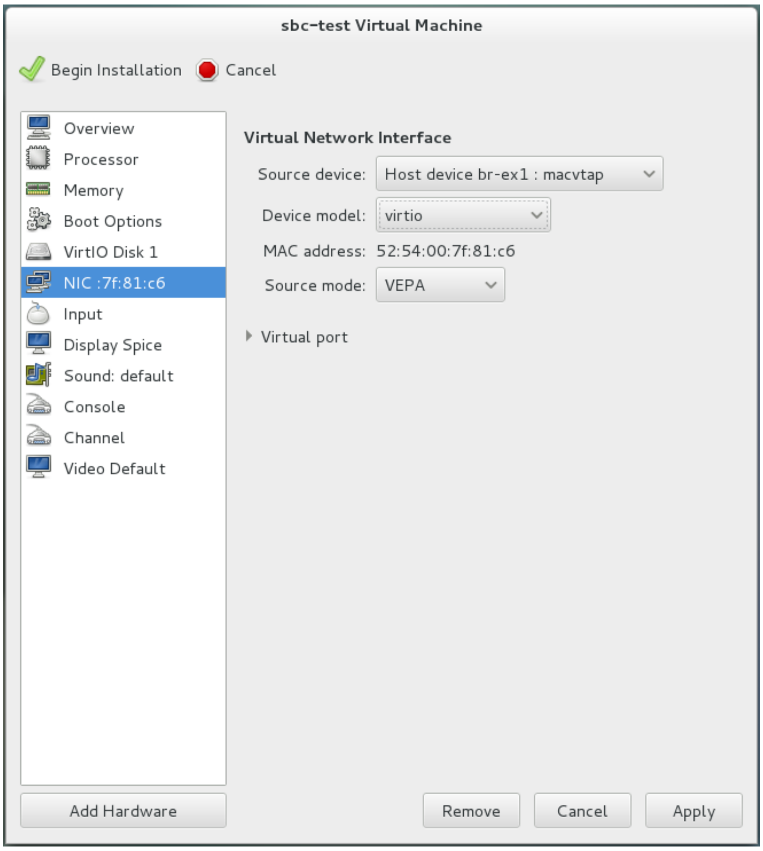
11. Click **Apply**.

12. In the virtual machine's properties, open the VirtIO Disk 1 configuration screen. Expand the 'Advanced options' section, and then from the 'Storage format' dropdown, select **qcow2**. Click **Apply**.



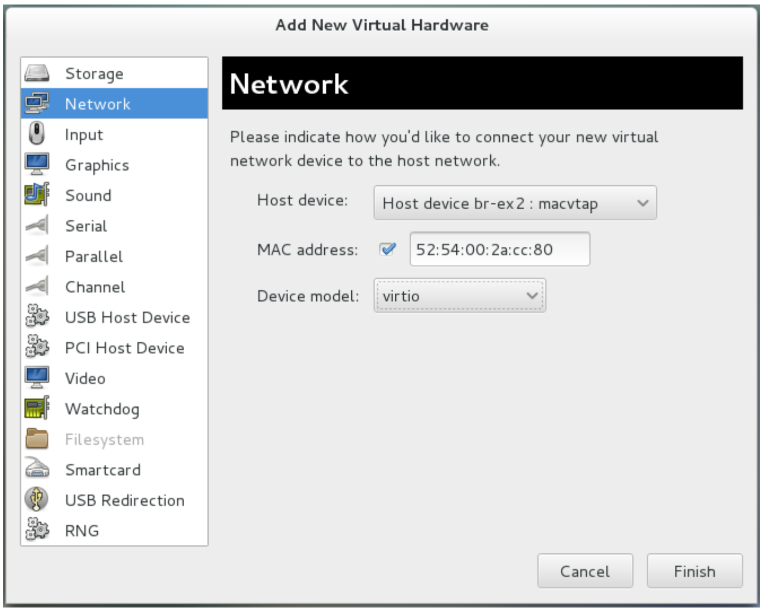
Virtual Disk
Source path: /var/lib/libvirt/images/sbc-test.qcow2
Device type: VirtIO Disk 1
Storage size: 20.00 GiB
Readonly: □
Shareable: □
Advanced options
Disk bus: VirtIO
Serial number:
Storage format: qcow2
Performance options

13. In the virtual machine's properties, open the NIC configuration screen. From the 'Source device' dropdown, select **Host device br-ex1: macvtap**. From the 'Device model' dropdown, select **virtio**. Click **Apply**.



Virtual Network Interface
Source device: Host device br-ex1 : macvtap
Device model: virtio
MAC address: 52:54:00:7f:81:c6
Source mode: VEPA
Virtual port

14. To configure additional Virtual Machine network interfaces, click **Add Hardware** (shown in the figure above), and then configure an additional NIC as shown below.



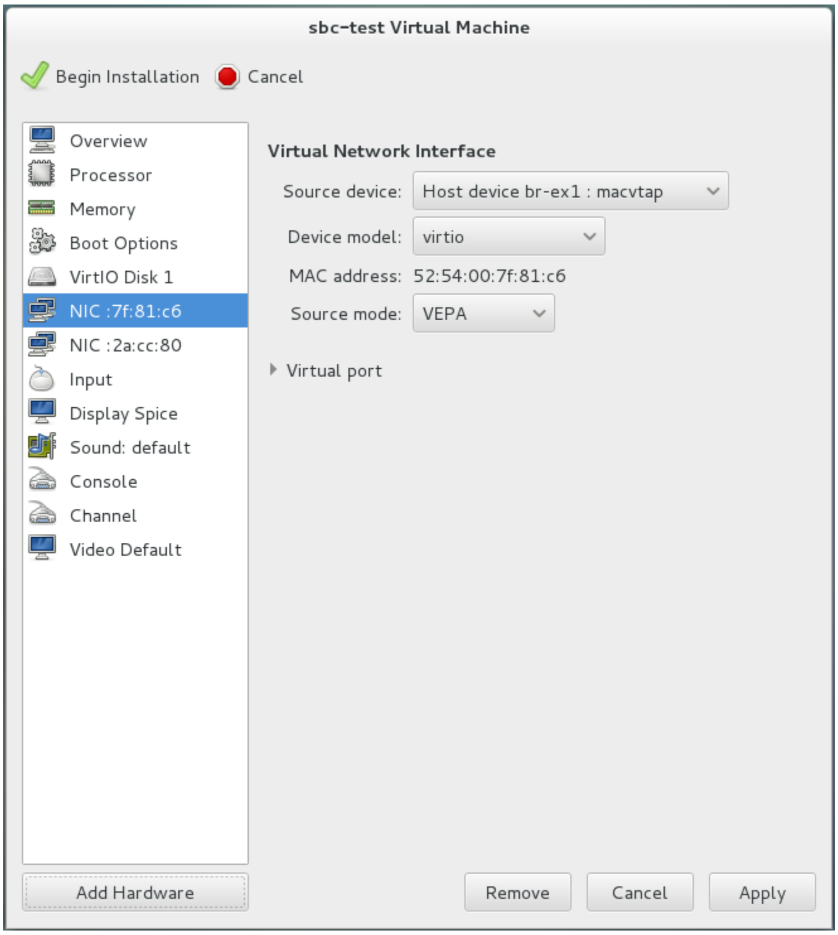
Please indicate how you'd like to connect your new virtual network device to the host network.

Host device: Host device br-ex2 : macvtap

MAC address: ☑ 52:54:00:2a:cc:80

Device model: virtio

15. Click **Begin Installation**.



Virtual Network Interface

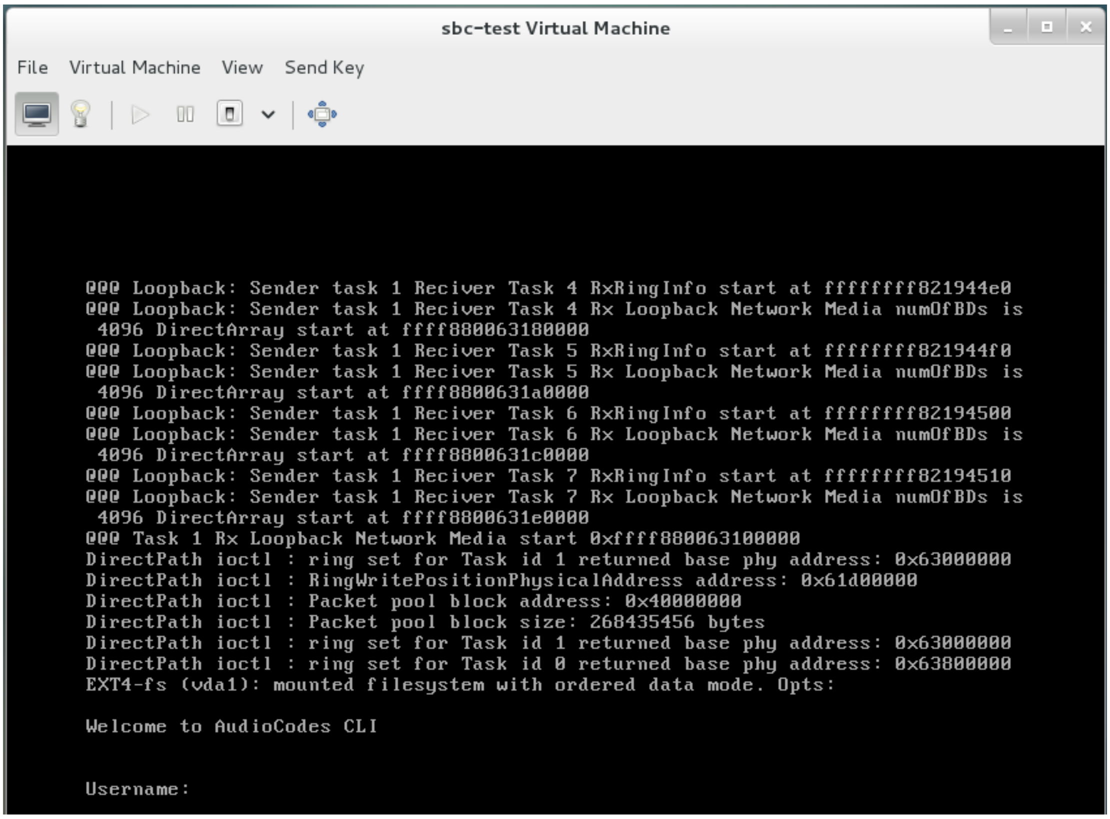
Source device: Host device br-ex1 : macvtap

Device model: virtio

MAC address: 52:54:00:7f:81:c6

Source mode: VEPA

16. Wait until the Mediant VE SBC fully starts and shows the CLI login prompt.



```
@@@ Loopback: Sender task 1 Reciver Task 4 RxRingInfo start at ffffffff821944e0
@@@ Loopback: Sender task 1 Reciver Task 4 Rx Loopback Network Media numOfBDs is
 4096 DirectArray start at ffff880063180000
@@@ Loopback: Sender task 1 Reciver Task 5 RxRingInfo start at ffffffff821944f0
@@@ Loopback: Sender task 1 Reciver Task 5 Rx Loopback Network Media numOfBDs is
 4096 DirectArray start at ffff8800631a0000
@@@ Loopback: Sender task 1 Reciver Task 6 RxRingInfo start at ffffffff82194500
@@@ Loopback: Sender task 1 Reciver Task 6 Rx Loopback Network Media numOfBDs is
 4096 DirectArray start at ffff8800631c0000
@@@ Loopback: Sender task 1 Reciver Task 7 RxRingInfo start at ffffffff82194510
@@@ Loopback: Sender task 1 Reciver Task 7 Rx Loopback Network Media numOfBDs is
 4096 DirectArray start at ffff8800631e0000
@@@ Task 1 Rx Loopback Network Media start 0xffff880063100000
DirectPath ioctl : ring set for Task id 1 returned base phy address: 0x63000000
DirectPath ioctl : RingWritePositionPhysicalAddress address: 0x61d00000
DirectPath ioctl : Packet pool block address: 0x40000000
DirectPath ioctl : Packet pool block size: 268435456 bytes
DirectPath ioctl : ring set for Task id 1 returned base phy address: 0x63000000
DirectPath ioctl : ring set for Task id 0 returned base phy address: 0x63800000
EXT4-fs (vda1): mounted filesystem with ordered data mode. Opts:

Welcome to AudioCodes CLI


Username:
```

17. Shut down the Virtual Machine by clicking **Shutdown** button.

18. If your Linux host (hypervisor) has more than one CPU installed, configure domain emulator pinning for optimal performance. Use the same values generated by **Generate from host NUMA configuration** in Step 8.

```
[root@virt-host ~]# virsh emulatorpin sbc-test 0-9 --config
```

19. Open the advanced Virtual Machine configuration editor.

```
[root@virt-host ~]# virsh edit sbc-test
```

20. Change the network interfaces configuration to look like this:

```
    <interface type='bridge'>
      <mac address='52:54:00:7f:81:c6'/>
      <source bridge='br-ex1'/>
      <virtualport type='openvswitch'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00'
      slot='0x03' function='0x0'/>
    </interface>
    <interface type='bridge'>
      <mac address='52:54:00:2a:cc:80'/>
      <source bridge='br-ex2'/>
      <virtualport type='openvswitch'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00'
      slot='0x04' function='0x0'/>
    </interface>
```

21. Save the new configuration and exit.

22. Dump the updated Virtual Machine configuration and verify that **<parameters interfaceid=…/>** was added under the **<virtualport>** tag.

```
[root@virt-host ~]# virsh dump sbc-test
...
    <interface type='bridge'>
      <mac address='52:54:00:7f:81:c6'/>
      <source bridge='br-ex1'/>
      <virtualport type='openvswitch'>
          <parameters interfaceid='74b6858e-8012-4caa-85c7-
b64902a19605'/>
      </virtualport>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00'
slot='0x03' function='0x0'/>
    </interface>
...
```

23. Start the Virtual Machine by clicking the **Start** ▷ button; the Virtual Machine will now have full network connectivity.

3.4.9 Configuring Virtual Machine to Operate with Hyper-Threading

The Mediant VE SBC virtual machine can utilize Hyper-Threading to achieve better performance, by using fewer physical cores. To support this, the Hyper-Threading capability must be enabled on the server and the correct mapping should be done between the logical cores of the server and the virtual CPUs of the Mediant VE SBC.

Note: This section is not relevant for host servers where Hyper-Threading is disabled.

The following steps should be done to determine how to achieve maximum performance from your server.

1. Choose the required capacity configuration from the Capacity table in the Release Notes. The capacity configuration specifies whether Hyper-Threading is utilized and the number of vCPUs required.

   For example, when the Capacity table specifies that Hyper-Threading is utilized and 8 vCPUs are required, it means that when using Xeon-SP (Skylake or Cascade), only 4 physical cores can be used to achieve similar capacity (depending only on the CPU speed). Alternatively, if Hyper-Threading is not utilized by the chosen configuration or an older CPU is used, then the recommended mapping should utilize 8 physical cores (16 logical cores).

2. View the CPU topology of your host server:

```
[root@virt-host ~]# for i in `seq 0 15`; do echo -n "physical-core $i is using HT cores "; cat /sys/devices/system/cpu/cpu$i/topology/thread_siblings_list; done
```

3. The output displays which logical cores reside on which physical core, for example:

```
physical-core 0 is using HT cores 0,16
physical-core 1 is using HT cores 1,17
physical-core 2 is using HT cores 2,18
physical-core 3 is using HT cores 3,19
physical-core 4 is using HT cores 4,20
physical-core 5 is using HT cores 5,21
physical-core 6 is using HT cores 6,22
physical-core 7 is using HT cores 7,23
physical-core 8 is using HT cores 8,24
physical-core 9 is using HT cores 9,25
physical-core 10 is using HT cores 10,26
physical-core 11 is using HT cores 11,27
physical-core 12 is using HT cores 12,28
physical-core 13 is using HT cores 13,29
physical-core 14 is using HT cores 14,30
physical-core 15 is using HT cores 15,31
```

If the chosen configuration utilizes Hyper-Threading and the host CPU is Xeon-SP, then continue with Section 3.4.9.1. Otherwise, continue with Section 3.4.9.2.

3.4.9.1 Configuring Virtual Machine to Utilize Hyper-Threading

When Hyper-Threading is required, you must map each logical core to each vCPU of the Mediant VE SBC virtual machine. For example, if Mediant VE SBC uses eight vCPUs, you can configure the virtual machine to allocate only four physical cores of the host (eight hyperthreaded cores).

> **Note:** When the CPU topology of the Mediant VE is defined with hyper-threading (threads=2), every two cascading vCPUs represent two threads of the same 'physical' core.

1. Shut down the virtual machine, by clicking the **Shutdown** ▣ button.
2. Open the advanced Virtual Machine configuration editor:

```
[root@virt-host ~]# virsh edit sbc-test
```

3. If you want to map 4 physical cores #1-#4 to the virtual machine (vcpu 0-8), and the CPU topology is as in the example above, then change the CPU mapping so that it appears as follows:

```
…
  <vcpu placement='static'>8</vcpu>
  <cputune>
    <vcpupin vcpu='0' cpuset='1'/>
    <vcpupin vcpu='1' cpuset='17'/>
    <vcpupin vcpu='2' cpuset='2'/>
    <vcpupin vcpu='3' cpuset='18'/>
    <vcpupin vcpu='4' cpuset='3'/>
    <vcpupin vcpu='5' cpuset='19'/>
    <vcpupin vcpu='6' cpuset='4'/>
    <vcpupin vcpu='7' cpuset='20'/>

  </cputune>
```

4. Save the new configuration, and then exit.
5. Start the virtual machine, by clicking the **Start** ▷ button; the virtual machine now operates with four physical cores.

3.4.9.2 Configuring Virtual Machine to Ignore Hyper-Threading

When Hyper-Threading is not required, you must map both logical cores of the same physical core to each vCPU of the Mediant VE SBC virtual machine. For example, if Mediant VE SBC uses four vCPUs, you must configure the virtual machine to allocate every pair of hyperthreaded cores to a certain vCPU so that a total of four physical cores (eight hyperthreaded cores) are allocated to the virtual machine.

1. Shut down the virtual machine, by clicking the **Shutdown** ▯ button.
2. Open the advanced Virtual Machine configuration editor:

   ```
   [root@virt-host ~]# virsh edit sbc-test
   ```

3. If you want to map physical cores 1-4 to the virtual machine (vcpu 0-3), change the CPU mapping so that it appears as follows:

   ```
   …
     <vcpu placement='static'>4</vcpu>
     <cputune>
       <vcpupin vcpu='0' cpuset='1,17'/>
       <vcpupin vcpu='1' cpuset='2,18'/>
       <vcpupin vcpu='2' cpuset='3,19'/>
       <vcpupin vcpu='3' cpuset='4,20'/>
     </cputune>
   ```

4. Save the new configuration and exit.
5. Start the virtual machine, by clicking the **Start** ▷ button; the virtual machine now operates with four physical cores.

3.5 Installing Mediant VE SBC on Microsoft Hyper-V

This section describes the installation process of Mediant VE SBC on Microsoft Hyper-V Server 2019:

➢ **To install Mediant VE SBC on Hyper-V:**

1. Update Windows Server 2019 to the latest updates and fixes (see Section 3.5.1)
2. Disable Virtual Machine Queues (VMQ) of Broadcom adapters (see Section 3.5.2)
3. Install the Virtual Machine (see Section 3.5.3).
4. Adjust the deployed virtual machine (see Sections 3.5.4)
5. Start the Mediant VE SBC (see Section 3.5.5)
6. Reconfigure the default IP address to match your network settings (see Section 3.7).

This section shows how to install the Mediant VE SBC on Microsoft Hyper-V.

3.5.1 Updating Windows Server 2019

You need to update Windows Server 2019, on which the Hyper-V hypervisor is installed, to the latest updates and fixes.

➢ **To update Windows Server 2019:**

1. Activate your Windows license (if not activated).
2. In the **Start** menu search box, type "windows update" and then click the searched result to open it.
3. Click **Check for updates** to check for Windows updates; suggested updates (optional and non-optional) are displayed.
4. Select the update (optional updates as well) to install it.
5. Repeat Steps 3 through 4 until you receive a message that Windows is up-to-date.

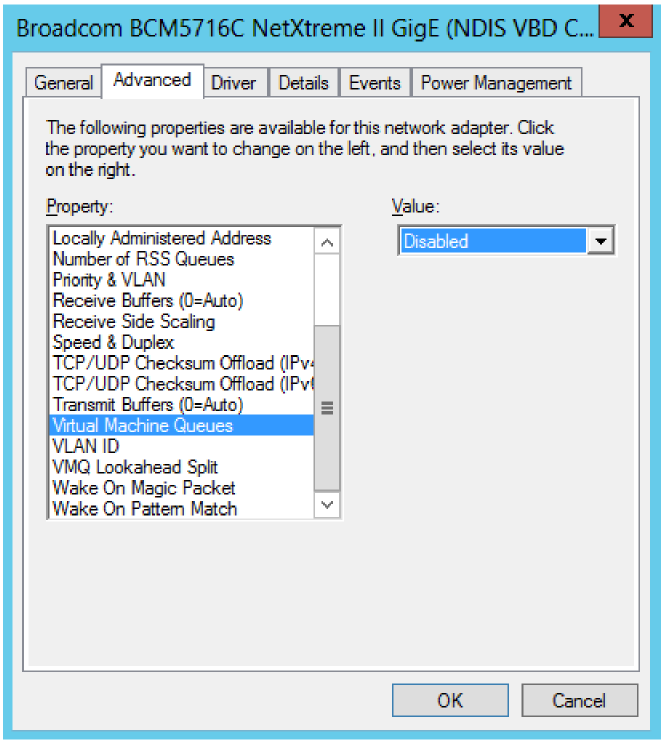
3.5.2 Disabling Virtual Machine Queues (VMQ) on Broadcom Adapters

If your Hyper-V host server uses Broadcom network adapters (NICs), you must disable the virtual machine queues (VMQ) on these network adapters.

➢ **To disable VMQ on Broadcom network adapters:**

1. Enter Control Panel\Network and Internet\Network Connections.
2. Right-click each Broadcom network connection, and then choose **Properties**.
3. Click **Configure**.
4. Select the **Advanced** tab.

5. In the Property list, select **Virtual Machine Queues** and set its value to "Disabled":



Broadcom BCM5716C NetXtreme II GigE (NDIS VBD C...

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: Virtual Machine Queues

Value: Disabled

6. Click **OK**.

7. Repeat steps 2 through 6 for all the Broadcom network connections.

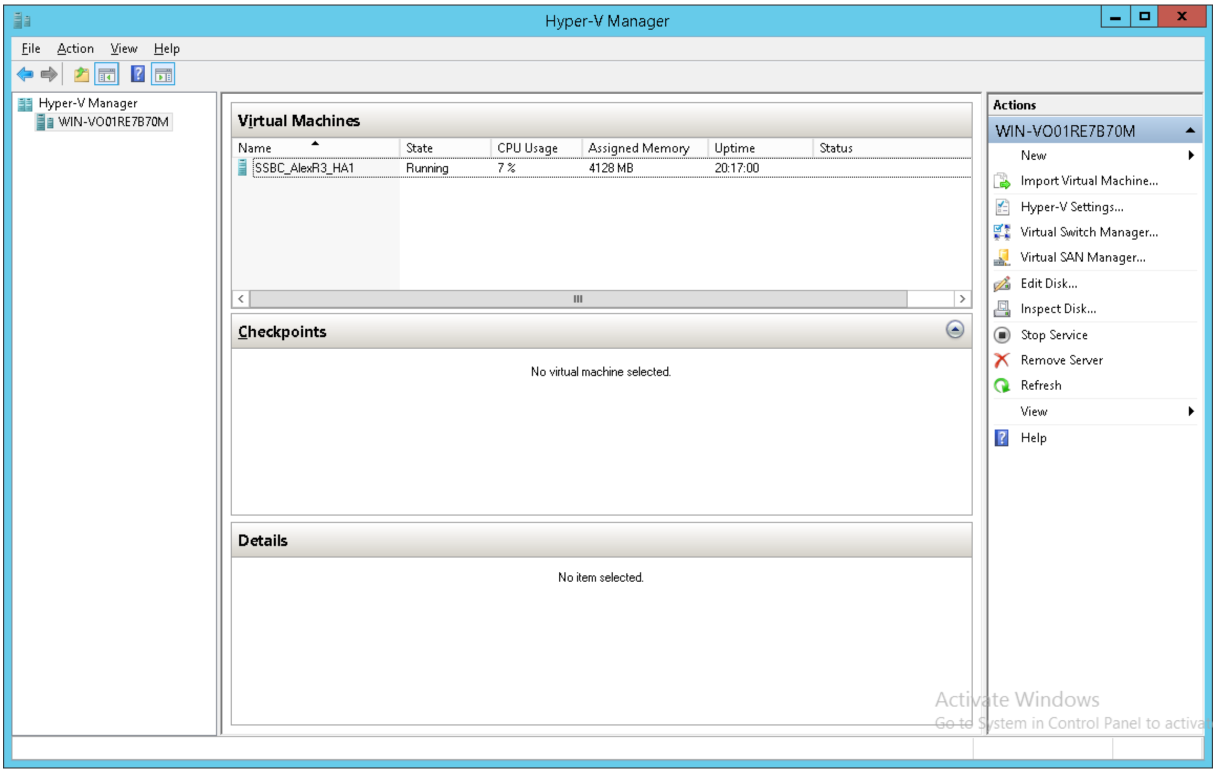
3.5.3 Installing the Virtual Machine

The Mediant VE SBC is distributed in the form of a virtual machine image.

➢ **To install the Mediant VE SBC on Microsoft Hyper-V:**

1. Extract the zip file containing the SBC virtual machine installation received from AudioCodes, to a local directory.
2. Open Hyper-V Manager (**Start** > **Administrative Tools** > **Hyper-V Manager**).

Figure 3-9: Installing the Mediant VE SBC on Hyper-V – Hyper-V Manager



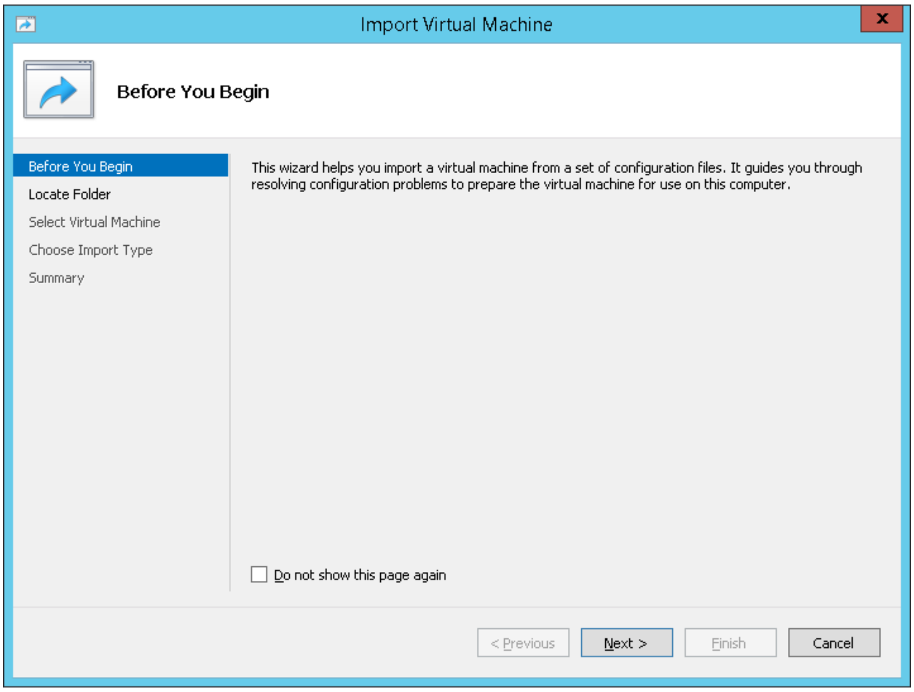
Name	State	CPU Usage	Assigned Memory	Uptime	Status
SSBC_AlexR3_HA1	Running	7 %	4128 MB	20:17:00	

No virtual machine selected.

No item selected.

3. Start the Import Virtual Machine wizard (**Action** > **Import Virtual Machine**), the the Before You Begin screen opens.

Figure 3-10: Installing Mediant VE SBC on Hyper-V – Import Virtual Machine Wizard

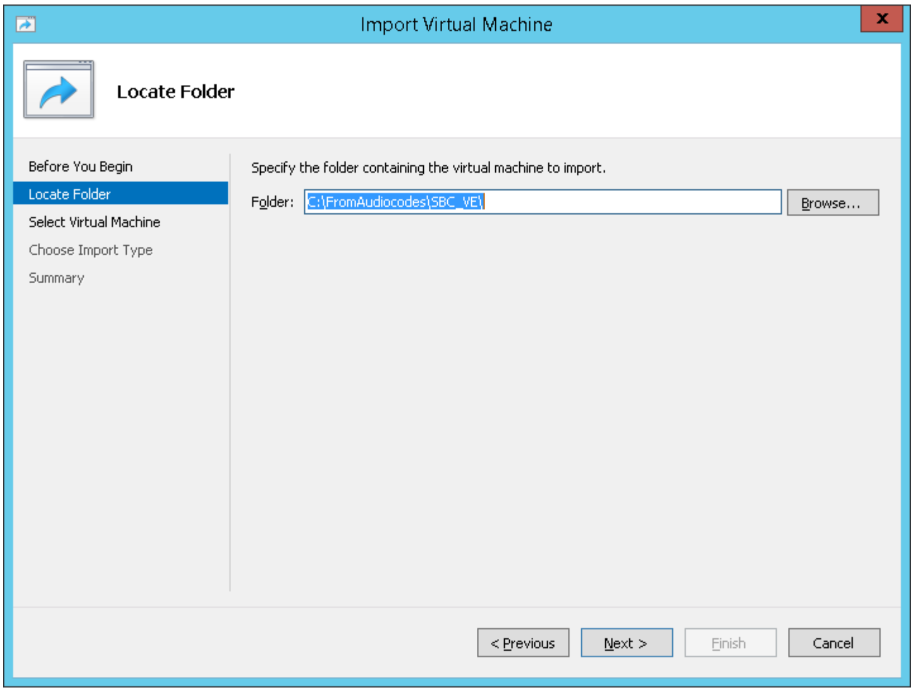


This wizard helps you import a virtual machine from a set of configuration files. It guides you through resolving configuration problems to prepare the virtual machine for use on this computer.

□ Do not show this page again

4. Click **Next**; this screen opens:

Figure 3-11: Installing Mediant VE SBC on Hyper-V – Enter Location of virtual machine Installation

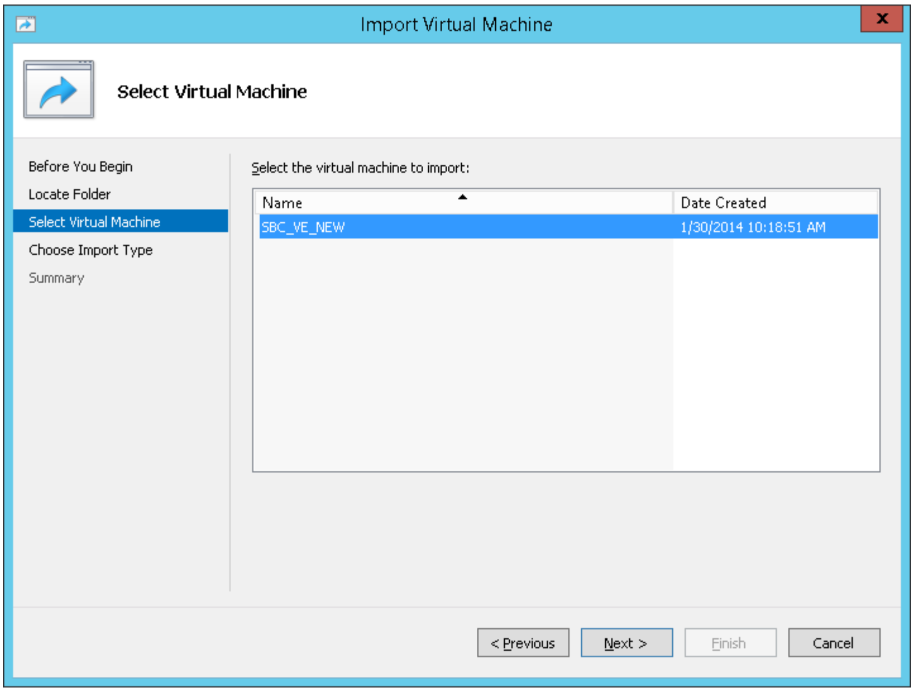


Specify the folder containing the virtual machine to import.

Folder: C:\FromAudiocodes\SBC_VE\

5. Enter the location of the virtual machine installation received from AudioCodes, and click **Next**.

Figure 3-12: Installing Mediant VE SBC on Hyper-V – Select Virtual Machine



Import Virtual Machine

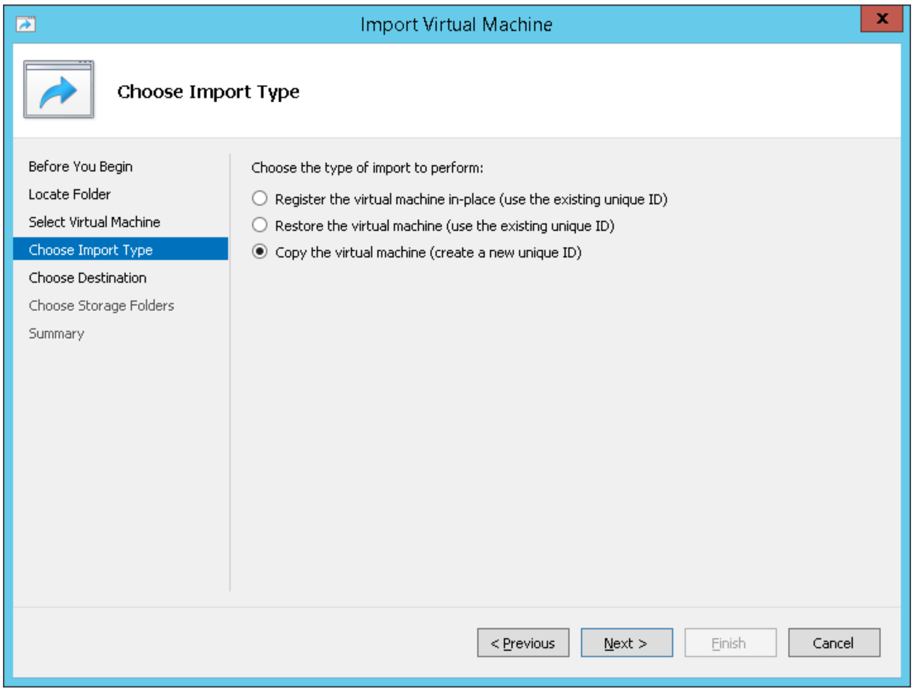
Select Virtual Machine

Select the virtual machine to import:

Name	Date Created
SBC_VE_NEW	1/30/2014 10:18:51 AM

6. Select the Virtual Machine and click **Next**.

Figure 3-13: Installing Mediant VE SBC on Hyper-V – Choose Import Type



Import Virtual Machine

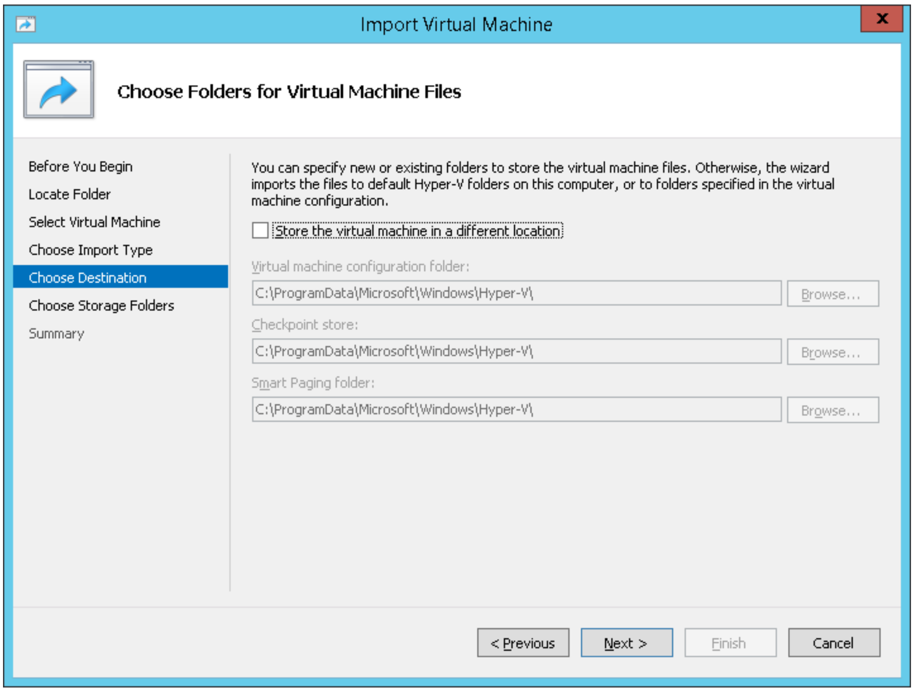
Choose Import Type

Choose the type of import to perform:

○ Register the virtual machine in-place (use the existing unique ID)
○ Restore the virtual machine (use the existing unique ID)
● Copy the virtual machine (create a new unique ID)

7. Select the **Copy the virtual machine** import type, and then click **Next.**

Figure 3-14: Installing Mediant VE SBC on Hyper-V – Choose Destination



You can specify new or existing folders to store the virtual machine files. Otherwise, the wizard imports the files to default Hyper-V folders on this computer, or to folders specified in the virtual machine configuration.

□ Store the virtual machine in a different location

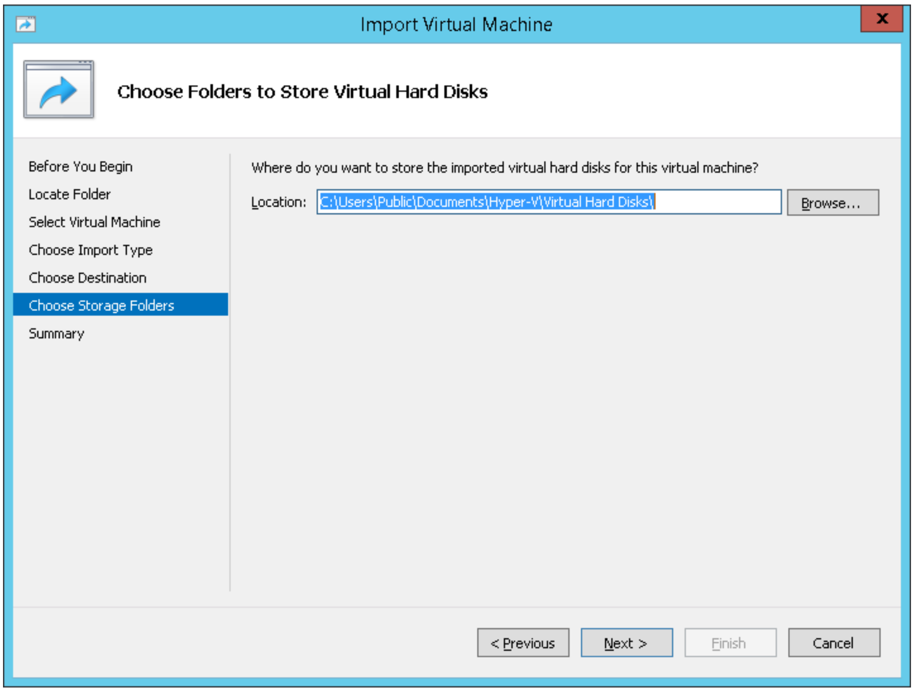
Virtual machine configuration folder: C:\ProgramData\Microsoft\Windows\Hyper-V\

Checkpoint store: C:\ProgramData\Microsoft\Windows\Hyper-V\

Smart Paging folder: C:\ProgramData\Microsoft\Windows\Hyper-V\

8. Choose the folders in which to store the Virtual Machine on your storage.

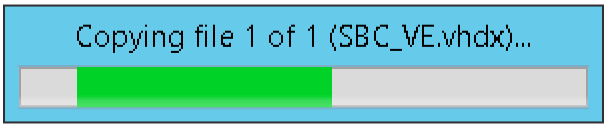
Figure 3-15: Installing Mediant VE SBC on Hyper-V – Choose Storage Folders



Where do you want to store the imported virtual hard disks for this virtual machine?

Location: C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\

9. Select the location of the virtual hard disk and click **Next**.
10. Click **Finish** to start the creation of the virtual machine; the installation progress indicator is shown.



Copying file 1 of 1 (SBC_VE.vhdx)...

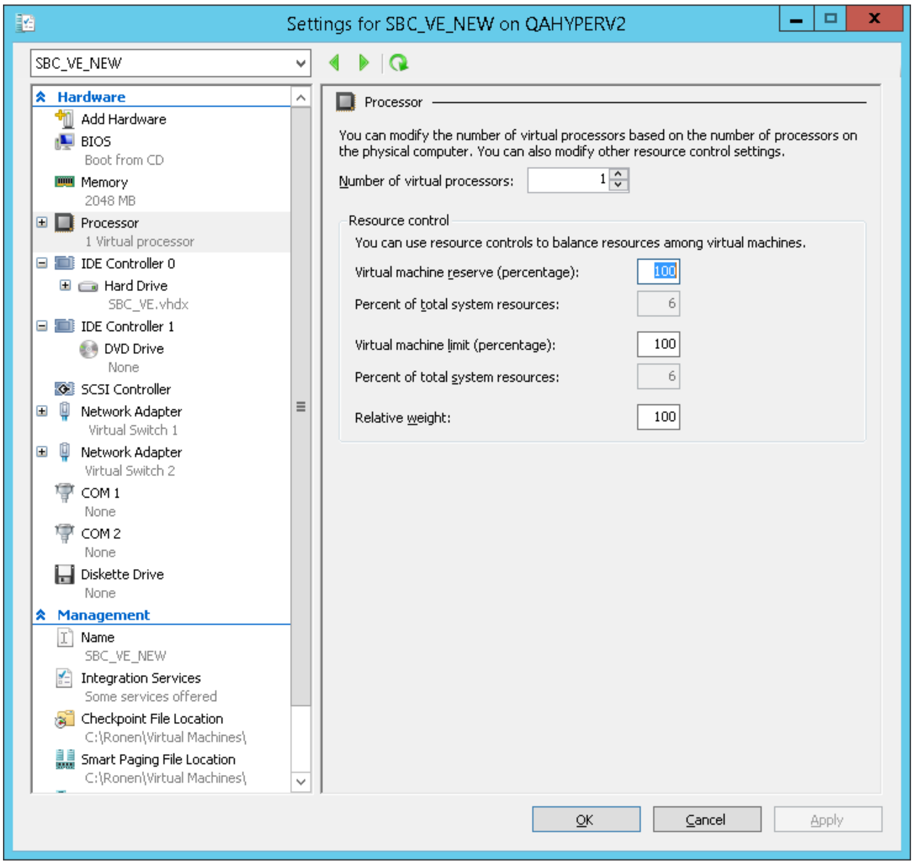
11. After the virtual machine is created, adjust its properties as described in Section 3.5.4.

3.5.4 Adjusting Virtual Machine to Chosen Mediant VE Flavor

➢ **To adjust the installed virtual machine for the selected SBC flavor:**

1. Locate the new virtual machine in the tree in the Hyper-V Manager, right-click it, and select **Settings**; the SBC Virtual Machine Properties screen opens.

Figure 3-16: Adjusting Virtual Machine for Mediant VE SBC – Settings



Settings for SBC_VE_NEW on QAHYPERV2

SBC_VE_NEW

Hardware
- Add Hardware
- BIOS
Boot from CD
- Memory
2048 MB
- Processor
1 Virtual processor
- IDE Controller 0
  - Hard Drive
SBC_VE.vhdx
- IDE Controller 1
  - DVD Drive
None
- SCSI Controller
- Network Adapter
Virtual Switch 1
- Network Adapter
Virtual Switch 2
- COM 1
None
- COM 2
None
- Diskette Drive
None

Management
- Name
SBC_VE_NEW
- Integration Services
Some services offered
- Checkpoint File Location
C:\Ronen\Virtual Machines\
- Smart Paging File Location
C:\Ronen\Virtual Machines\

Processor

You can modify the number of virtual processors based on the number of processors on the physical computer. You can also modify other resource control settings.

Number of virtual processors: 1

Resource control

You can use resource controls to balance resources among virtual machines.

Virtual machine reserve (percentage): 100

Percent of total system resources: 6

Virtual machine limit (percentage): 100

Percent of total system resources: 6

Relative weight: 100

2. Under the Hardware folder, select **Processor**.
3. Configure the number of virtual processors to match the number of CPUs for the chosen Mediant VE SBC flavor.
4. Configure 'Virtual machine reserve (percentage)' to **100%**, and then click **OK**.
5. Under the Hardware folder, select **Memory**.
6. Configure the memory size according to the chosen Mediant VE SBC flavor, and then click **OK**.

3.5.5 Starting the Mediant VE SBC

1. In Hyper-V Manager, right-click the name of the virtual machine, and then click **Connect**.
2. In the Virtual Machine Connection window, click Start.
3. Proceed to Section 3.7.

3.6 Deploying Mediant VE SBC in Amazon AWS

You can deploy Mediant VE SBC as a virtual server (*instance*) in the Amazon Web Services (AWS) cloud environment. For more information, refer to the document *Mediant VE Installation Manual for Amazon AWS.*

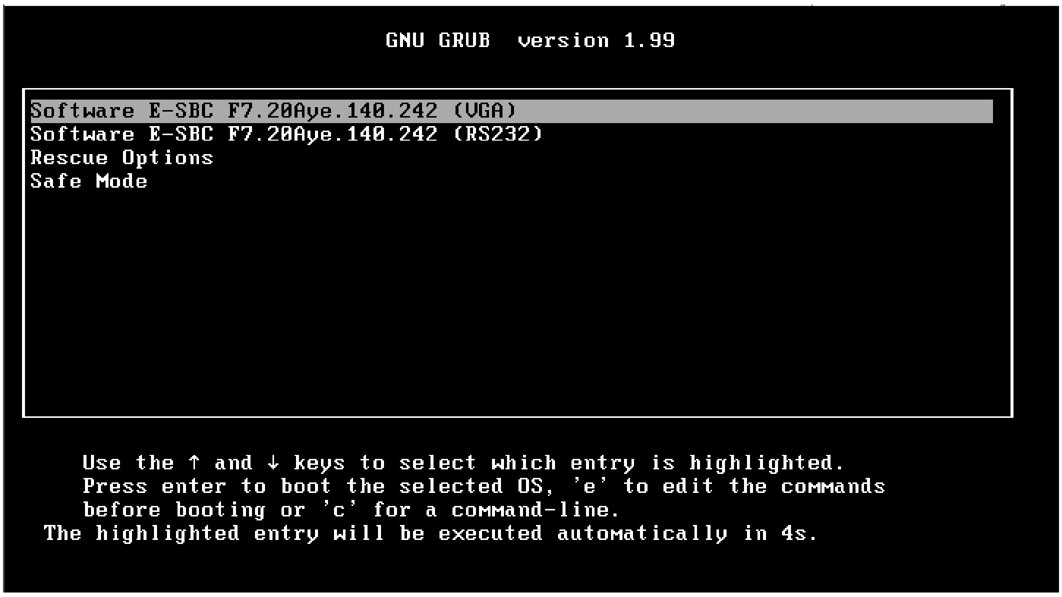
3.7 Configuring Console Access Method

The console lets you use the device's Command-Line Interface (CLI) to configure and manage the device. You can access the console through either VGA or serial RS-232 interface. By default, when the device boots up (for example after a reset), it accesses the console through VGA, which means that it can be captured by the console tools provided by the chosen Hypervisor.

To change the access method, use one of the following options:

- **GRUB Boot Loader Menu:**
  1. Reboot Mediant VE SBC.
  2. In the GRUB menu that's displayed during the boot-up, press the down key to select **VGA** or **RS232**.
  3. Press **Enter**.

Figure A-17: Console Access Method via GRUB Menu



```
                         GNU GRUB  version 1.99

 Software E-SBC F7.20Aye.140.242 (VGA)
 Software E-SBC F7.20Aye.140.242 (RS232)
 Rescue Options
 Safe Mode

      Use the ↑ and ↓ keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the commands
      before booting or 'c' for a command-line.
   The highlighted entry will be executed automatically in 4s.
```

Note: When you select the console access method through GRUB, if the Mediant VE SBC resets at a later stage, the access method reverts to VGA.

- **CLI:**

```
# configure troubleshoot
(config-troubleshoot)# startup-n-recovery
(startup-n-recovery)# system-console-mode {vga|rs232}
```

- **Ini File:** SystemConsoleMode (0 = VGA; 1 = RS232)

If you want to access the Mediant VE SBC console through the serial Port (RS-232) of the host, do the following:

1. Add a virtual serial port device on the virtual machine and map it to the required serial port of the host.
2. Change the access method of the Mediant VE SBC console to **RS232**, as described above.

3.8 Reconfiguring Default IP Address to Match Network Settings

After installation, the Mediant VE SBC is assigned a default IP address that will most likely be inaccessible from the customer's network. This address is assigned to the first virtual network interface card, connected to the 'trusted' virtual network switch during Mediant VE SBC installation.

Table 3-1: Default IP Address

Parameter	Value
IP Address	192.168.0.2
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0

Note: This section is not applicable to Amazon EC2 and OpenStack deployments, where the Mediant VE SBC automatically acquires IP address(es) assigned to it by the cloud environment.

Reconfigure the IP address in order to connect to the Mediant VE SBC's Web-based Management Tool (referred to as 'Web interface' in this document). The procedure below shows how to change the default IP address using the CLI. The procedure uses the regular CLI commands. Alternatively, you can use the CLI Wizard utility to set up your Mediant VE SBC with the initial OAMP settings. The utility provides a fast-and-easy method for initial configuration of the Mediant VE SBC through CLI. For more information, refer to the *CLI Wizard User's Guide*.

➢ **To reconfigure the IP address using CLI:**

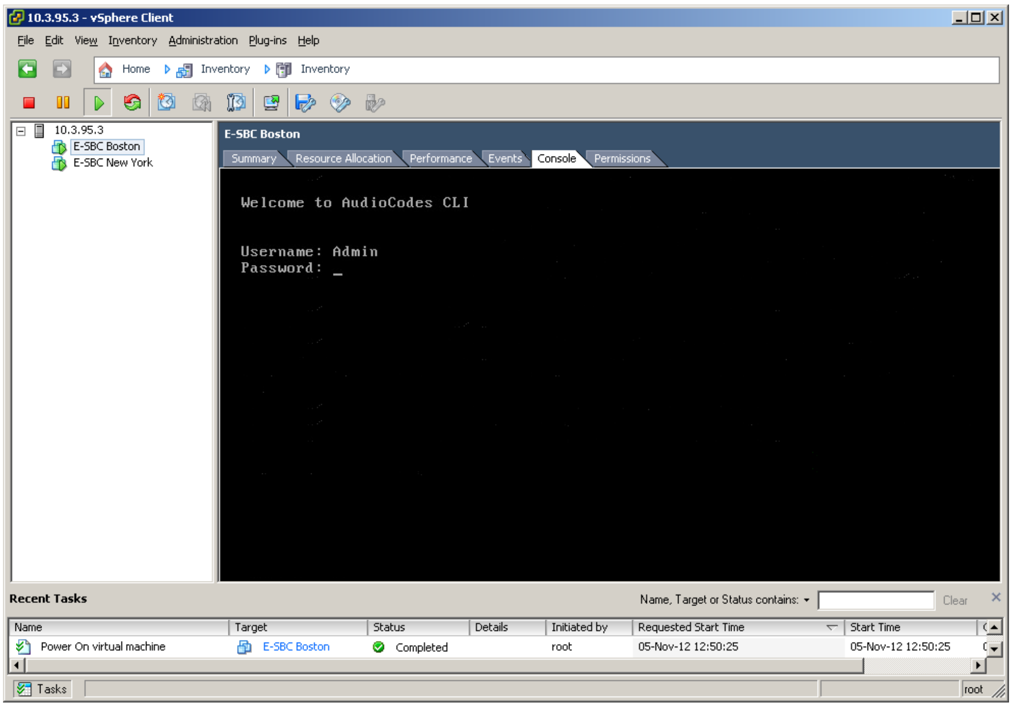
1. Connect to the virtual machine’s console (e.g., in vSphere, switch to **Console** tab); you will be presented with the CLI management interface.
2. At the prompt, type the username (default is **Admin** - case sensitive), and then press ENTER:

```
Username: Admin
```

3. At the prompt, type the password (default is **Admin** - case sensitive), and then press ENTER:

```
Password: Admin
```

Figure 3-18: CLI Management Interface



```
Welcome to AudioCodes CLI

Username: Admin
Password: _
```

Name	Target	Status	Details	Initiated by	Requested Start Time	Start Time
Power On virtual machine	E-SBC Boston	Completed		root	05-Nov-12 12:50:25	05-Nov-12 12:50:25

4. At the prompt, type **enable** and press ENTER:

```
> enable
```

5. At the prompt, type the password again and press ENTER:

```
Password: Admin
```

6. At the prompt, type the following commands to access the network interface configuration:

```
# configure network
(config-network)# interface network-if 0
(network-if-0)#
```

Note: Use the **Tab** key to auto-complete partially entered commands.

7. At the prompt, type the following commands to configure the IP address, prefix length and default gateway:

```
(network-if-0)# ip-address 10.4.212.155
(network-if-0)# prefix-length 16
(network-if-0)# gateway 10.4.0.1
```

Note: The IP and gateway addresses above are *by way of example only*.
Use IP and gateway addresses appropriate to your network configuration.

8. At the prompt, type the following command to apply the network interface configuration and exit the table:

```
(network-if-0)# activate
(network-if-0)# exit
```

9. If Mediant SE SBC is connected to the IP network that uses VLAN ID, type the following commands to configure it (otherwise skip to step 10):

```
(config-network)# network-dev 0
(network-dev-0)# vlan-id 10
(network-dev-0)# activate
(network-dev-0)# exit
```

10. At the prompt, type **`exit`** to complete the configuration:

```
(config-network)# exit
```

Connect to the Mediant VE SBC through the Web interface to provision it. For details on provisioning, see the *Mediant Server & Virtual Editions SBC User's Manual*.

3.9 Adding Transcoding Capabilities

Certain Mediant VE SBC flavors support transcoding capabilities. To provide this support, the following is required:

1. Upgrade the License Key to include transcoding capabilities.
2. Increase the number of vCPUs to support the required transcoding capabilities (as defined in the *Release Notes*). The number of vCPUs should be increased according to chosen Mediant VE SBC flavor.
3. Increase memory allocated for the virtual machine according to chosen Mediant VE SBC flavor.
4. Configure the 'SBC Performance Profile' parameter to **Optimize for Transcoding** (for more information, refer to the *User's Manual)*.

3.10 Identifying Incompatible Hardware Components

Each time Mediant VE SBC is started, it validates its virtual machine configuration and issues a warning if incompatible hardware and/or virtual machine configuration is detected. The warning is displayed on the virtual machine console for 10 seconds during the boot up sequence, after which the normal startup sequence continues.

You can also view details of the virtual machine configuration and/or hardware platform using the **`show system hardware`** CLI command. Incompatible components are indicated by asterisk (*).

Note: Incompatible components should be replaced or, alternatively, not be mapped to the SBC virtual machine.

The example below shows an incompatible NIC:

```
# show system hardware
cpu: Intel<R> Xeon<R> CPU E31220 @ 3.10GHz, total 4 cores
memory:  4096 MB
chassis: Microsoft Hyper-V Server
network:
Intel Corporation 82574L Gigabit Network Connection
Intel Corporation 82574L Gigabit Network Connection
*Realtek Semiconductor Co., Ltd. RTL-8169 Gigabit Ethernet (rev 10)
*Realtek Semiconductor Co., Ltd. RTL-8169 Gigabit Ethernet (rev 10)
```

3.11 Changing MAC Addresses from 'Dynamic' to 'Static'

By default, the MAC addresses of the SBC Virtual Machine are set dynamically by the hypervisor. Consequently, they might be changed under certain circumstances – for example, after moving the virtual machine between Hyper-V hosts.

To prevent this, it's advisable to change the MAC Addressees from Dynamic to Static.

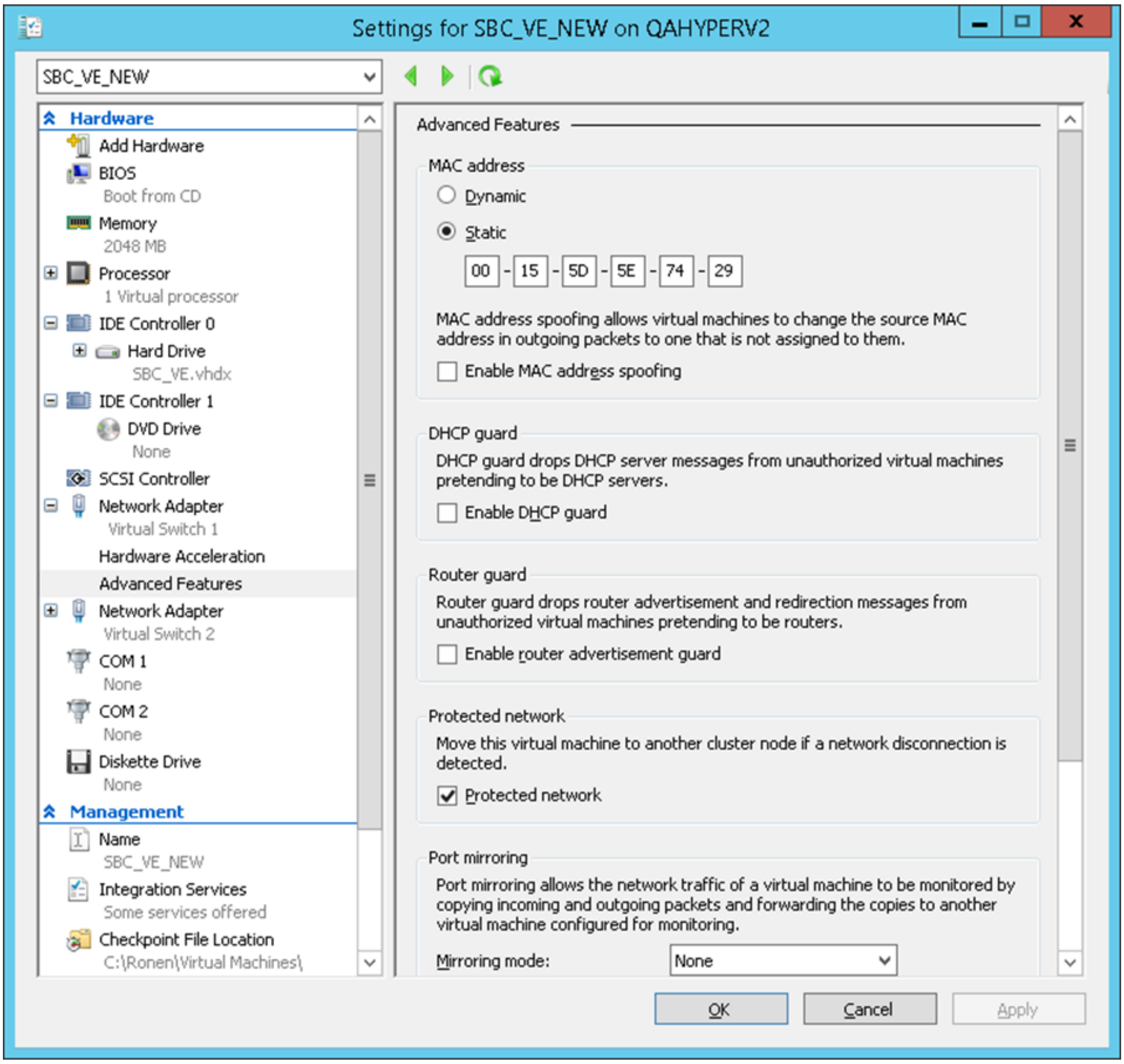
3.11.1 Changing MAC Addresses to 'Static' in Microsoft Hyper-V

This section shows how to change the MAC address to **Static** in Microsoft Hyper-V.

➢ **To change the MAC address to 'Static' in Microsoft Hyper-V:**

1. *Turn-off* the SBC virtual machine.
2. Enter the **Settings** of the selected SBC virtual machine
3. For each **Network Adapter**, do the following:
   a. Choose **Advanced Features**
   b. Change the 'MAC address' option to **Static**.

Figure 3-19: Advanced Features - Network Adapter – Static MAC Address



Settings for SBC_VE_NEW on QAHYPERV2

SBC_VE_NEW

Hardware
- Add Hardware
- BIOS
  Boot from CD
- Memory
  2048 MB
- Processor
  1 Virtual processor
- IDE Controller 0
  - Hard Drive
    SBC_VE.vhdx
- IDE Controller 1
  - DVD Drive
    None
- SCSI Controller
- Network Adapter
  Virtual Switch 1
  - Hardware Acceleration
  - Advanced Features
- Network Adapter
  Virtual Switch 2
- COM 1
  None
- COM 2
  None
- Diskette Drive
  None

Management
- Name
  SBC_VE_NEW
- Integration Services
  Some services offered
- Checkpoint File Location
  C:\Ronen\Virtual Machines\

Advanced Features

MAC address

○ Dynamic

● Static

00 - 15 - 5D - 5E - 74 - 29

MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

□ Enable MAC address spoofing

DHCP guard

DHCP guard drops DHCP server messages from unauthorized virtual machines pretending to be DHCP servers.

□ Enable DHCP guard

Router guard

Router guard drops router advertisement and redirection messages from unauthorized virtual machines pretending to be routers.

□ Enable router advertisement guard

Protected network

Move this virtual machine to another cluster node if a network disconnection is detected.

☑ Protected network

Port mirroring

Port mirroring allows the network traffic of a virtual machine to be monitored by copying incoming and outgoing packets and forwarding the copies to another virtual machine configured for monitoring.

Mirroring mode: None

3.12 Ensuring Optimal Performance

The maximum capacity figures supported by Mediant VE SBC, as specified in the *Release Notes,* are highly dependable on the configuration of the hypervisor and the virtual machine. The maximum capacity figures are based on the following assumptions:

- The CPU type and speed reference that was tested with:
  - Xeon SP (Skylake) Gold 6126, 2.6 GHz
- BIOS settings, as described in Section 3.1.
- No other virtual machines are overloading the shared server's resources (such as the shared NICs).
- Maximum average packet loss of up to 0.02%. This average packets loss doesn't affect voice quality processed by Mediant VE SBC.
- Optimized configuration is done:
  - **VMWare:** Optimization settings of the hypervisor, as described in Section 3.2.2.
  - **KVM:** Optimization settings of the virtual machine, as described in Section 3.4.8. Note that it is possible to improve performance significantly by adding physical cores to handle the emulator. For example, assuming the ***sbc-test*** virtual machine is pinned to use two physical cores 8 and 9, to improve performance it is possible to place the emulator tasks on different physical core(s) such as core #7 in the example below:

```
# virsh edit sbc-test
   …
   <vcpu placement='static' cpuset='8-9'>2</vcpu>
  <cputune>
    <emulatorpin cpuset='7'/>
  </cputune>
   …
```

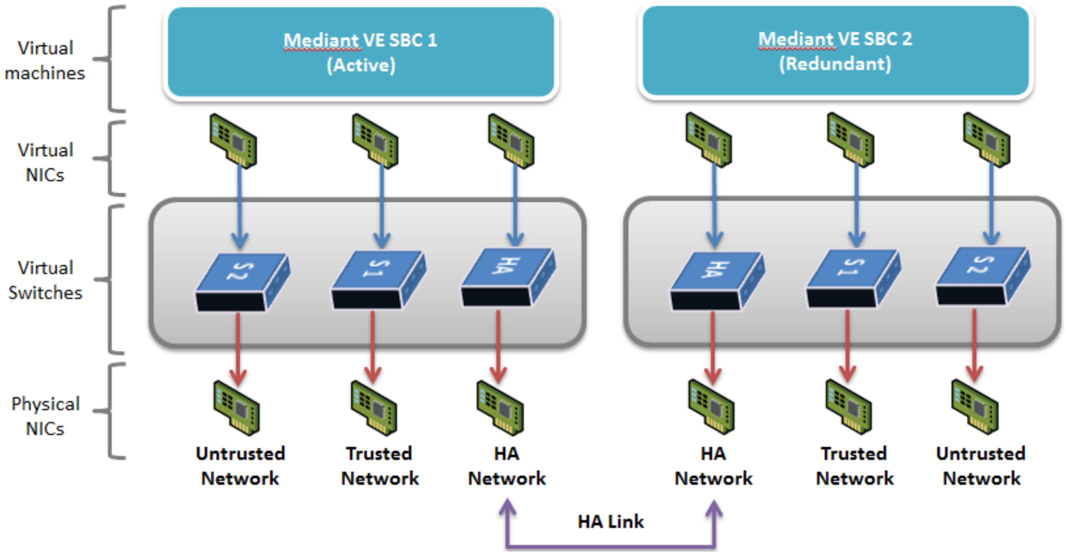
3.13 Installing a High-Availability System

Users can configure two Virtual Machines, running on different servers to work in a High Availability (HA) configuration.

➢ **To configure an HA system:**

1. Reconfigure a temporary IP address for each Mediant VE SBC, according to the instructions under Section 3.7.
2. To support HA functionality, the Mediant VE SBCs must be installed with the an HA-enabled license. Follow the instructions described in Chapter 4 for licensing each Mediant VE SBC in the HA system.
3. Follow the instructions described in the section 'High Availability System' in the *Mediant Server & Virtual Editions SBC User's Manual*, and configure each Mediant VE SBC accordingly using the Web interface.

Figure 3-20: Virtual Networking Configuration for HA System



Note: The physical NICs used by the Mediant SBC VE virtual machine must not share traffic with other applications such as other virtual machines or the hypervisor itself. This also applies to the physical NICs used for the HA link because overloading these NICs may cause false switchovers.

4 Licensing the Product

Once you have successfully completed the product's software installation, you can operate the product with one of the following license options.

Note: By default, the product software installation provides a free license for up to three concurrent sessions (signaling and media) and three user registrations (far-end users). This allows you to evaluate the product prior to purchasing it with your required capacity and features. To allow call transcoding with this free license, you need to configure the 'SBC Performance Profile' parameter to **Optimize for Transcoding** (for more information, refer to the *User's Manual*).

4.1 Obtaining and Activating a Purchased License Key

For the product to provide you with all your capacity and feature requirements, you need to purchase a new License Key that allows these capabilities. The following procedure describes how to obtain and activate your purchased License Key.

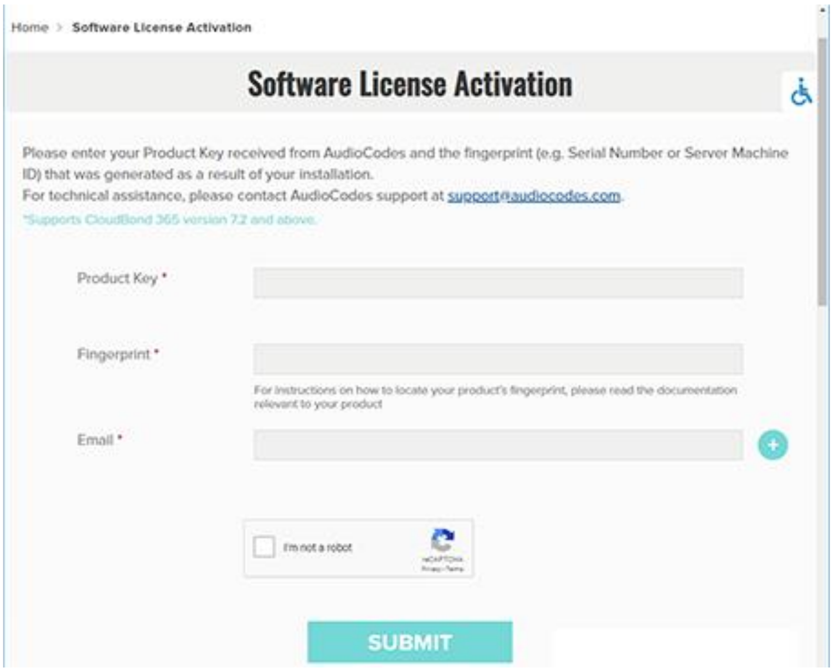
Note:

- License activation is intended **only** for first-time software activation upon product purchase (or if your License Key is "lost", due to whatever reason). For subsequent software feature upgrades, the License Key file is e-mailed to you after your Purchase Order has been processed.
- For HA, each unit has its own Serial Number, Product Key and License Key. Therefore, the instructions in this section must be done for each unit.

➢ To obtain and activate the License Key:

1. Open AudioCodes Web-based Software License Activation tool at https://www.audiocodes.com/swactivation:

Figure 4-1: Software License Activation Tool



Home > Software License Activation

Software License Activation

Please enter your Product Key received from AudioCodes and the fingerprint (e.g. Serial Number or Server Machine ID) that was generated as a result of your installation.
For technical assistance, please contact AudioCodes support at support@audiocodes.com.
*Supports CloudBond 365 version 7.2 and above.

Product Key *

Fingerprint *

For instructions on how to locate your product's fingerprint, please read the documentation relevant to your product

Email *

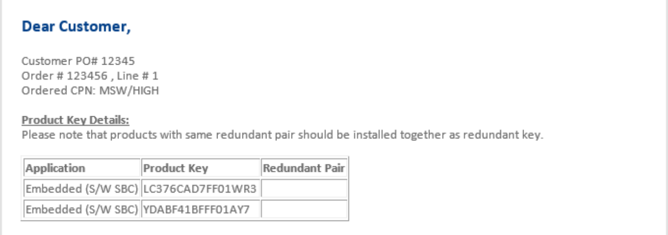
□ I'm not a robot

SUBMIT

2. Enter the following information:

   - **Product Key:** The Product Key identifies your specific Mediant VE SBC purchase for the purpose of subsequent communication with AudioCodes (for example, for support and software upgrades). The Product Key is provided in the Order Confirmation email sent to you by AudioCodes upon your purchase, as shown in the example below:

Figure 4-2: Product Key in Order Confirmation Email



Dear Customer,

Customer PO# 12345
Order # 123456 , Line # 1
Ordered CPN: MSW/HIGH

Product Key Details:
Please note that products with same redundant pair should be installed together as redundant key.

Application	Product Key	Redundant Pair
Embedded (S/W SBC)	LC376CAD7FF01WR3	
Embedded (S/W SBC)	YDABF41BFFF01AY7	

Note: For 1+1 High-Availability orders, you are provided with two Product Keys, one for each unit. In such cases, you need to perform license activation twice in order to obtain License Keys for both units.

   - **Fingerprint:** The fingerprint is the Mediant VE SBC's Serial Number. The Serial Number uniquely identifies the software installation. The Serial Number is displayed in the 'Serial Number' field on the Device Information page (**Monitor** menu > **Monitor** menu > **Summary** tab > **Device Information**).
   - **Email:** Provide one or more e-mail addresses to where you want the License Key to be sent.

3. Click **Send** to submit your license activation request.

4. Once AudioCodes processes and completes your license activation, you will receive an email notification with the License Key file attached. Open the file with any text-based program (such as Notepad) and make sure that the serial number ("**S/N**") in the License Key is correct and reflects the Serial Number of your Mediant VE SBC.

Warning: Do not modify the contents of the License Key file.

4.2 Installing the License Key

For installing the License Key on Mediant CE, refer to the *Mediant Software SBC User's Manual*.

> **Note:** The License Key file for HA contains two License Keys - one for the active device and one for the redundant device. Each License Key has a different serial number ("S/N"), which reflects the serial number of each device in the HA system.

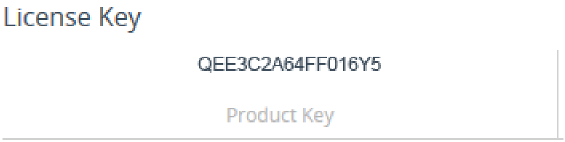
4.3 Product Key

The Product Key identifies a specific purchase of your device installation for the purpose of subsequent communication with AudioCodes (e.g., for support and software upgrades). The Product Key is provided in the order-confirmation email sent to you upon your product purchase and is used for activating your license through AudioCodes Software License Activation tool.

The Product Key is included in the License Key. Once the License Key is installed, you can view the Product Key in the following Web pages:

- License Key page (**Setup** menu > **Administration** tab > **License** folder > **License Key**). The Product Key is displayed in the read-only 'Product Key' field, as shown in the example below:

Figure 4-3: Viewing Product Key



License Key

QEE3C2A64FF016Y5

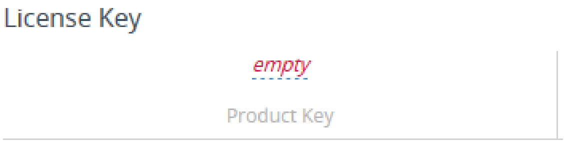
Product Key

- Device Information page.

If your License Key was purchased in an earlier version (for example, 7.0), the 'Product Key' field may appear empty. In such a scenario, request the Product Key from your AudioCodes sales representative. Once received, do the following:

1. Open the License Key page.
2. Locate the Product Key group:

Figure 4-4: Empty Product Key Field



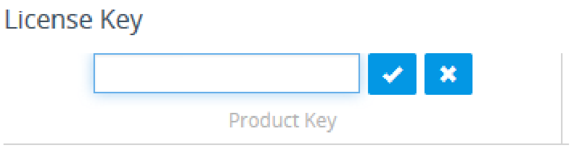
License Key

empty

Product Key

3. Click "empty"; the following appears:

Figure 4-5: Entering Product Key



License Key

Product Key

4. In the field, enter the Product Key, and then click **Submit** ✓ (or **Cancel** ✖ to discard your entry).

A Configuring the Network

A.1 Virtual NIC Types

The Mediant VE SBC virtual appliance provided by AudioCodes contains two virtual NICs.

- **VMware ESXi:** The OVF template contains two virtual NICs of type VMXNET3. This configuration provides optimal network and CPU performance. If you add additional virtual NICs, make sure that they are of the same VMXNET3 type.
- **Hyper-V:** The Virtual Machine image contains two virtual NICs of type "network adapter". If you add additional virtual NICs, make sure that they are of the same type (and are not "legacy network adapters").
- **KVM:** Use virtual NICs of type **virtio** and connect them to Open vSwitch bridges for optimal performance.
- **OpenStack:** Use flat provider networks for optimal performance.

Mediant VE SBC also supports passthrough NICs. This option gives the best network and CPU performance but requires allocation of a NIC to a specific virtual machine without the capability of sharing it with other virtual machines. For details, refer to the *hypervisor documentation*.

Warning: For VMware, the Mediant VE SBC supports only virtual NICs of type VMXNET3. Other vNIC types, e.g., E1000 or VMXNET2 are not supported.

A.2 Changing the Number of Virtual NIC Adapters

You can add/remove virtual adapters to the Mediant VE SBC. When adding/removing a NIC, shutdown is required. For details, refer to the *hypervisor documentation*.

It’s recommended to take a System Snapshot before you add/remove a NIC (see Section Rescue Options on page63).

A.3 Virtual Network Configuration

The virtual network can be configured in various configurations depending on your implementation, number of virtual machines, physical adapters, network security requirements, VLANs topology, etc.

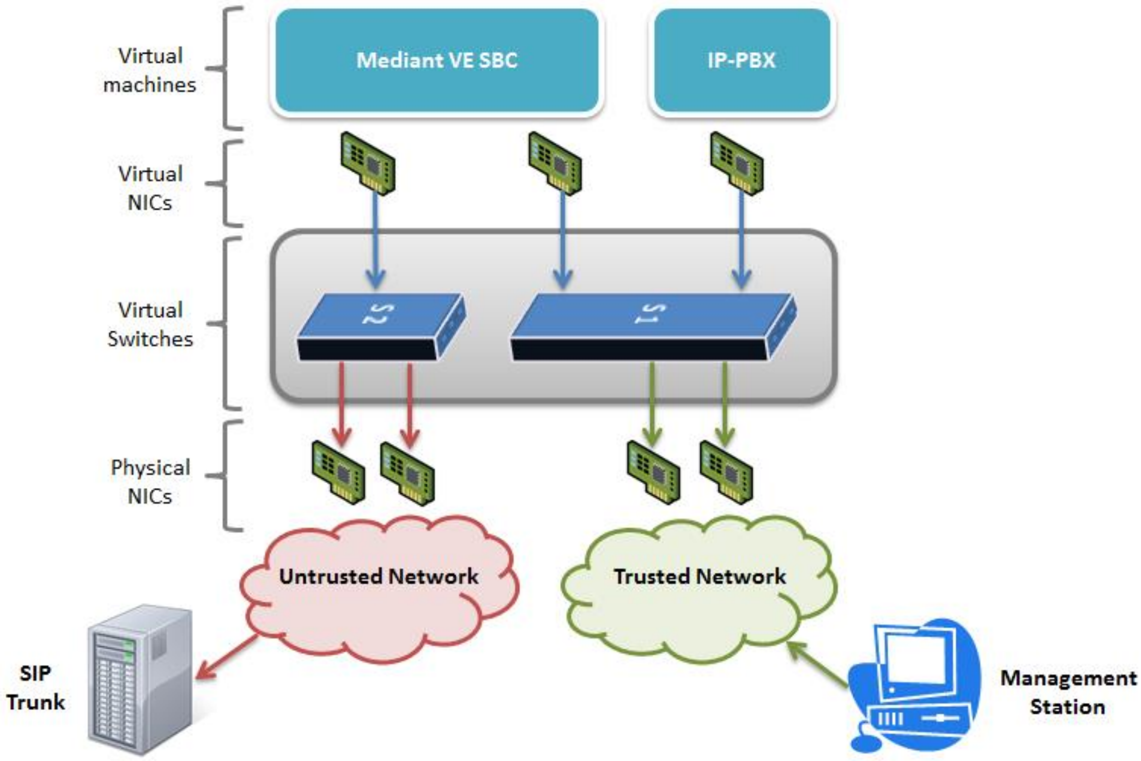
Use the following guidelines when implementing virtual network configuration:

- Create separate networks for trusted and untrusted traffic.
- Connect two physical network ports to each virtual network – to enable Ethernet port redundancy.

Note: Mediant VE SBC supports Ethernet port redundancy on its own (via Ethernet Groups that may be connected to two vNICs). In most deployments, however, this functionality is not needed – instead, only one vNIC is used and Ethernet port redundancy is implemented by virtual switch.

- Use trusted network for management traffic (Web, CLI, SNMP).
- For HA configurations, create a separate network for HA traffic.

Figure A-1: Network Configuration Example



B Rescue Options

The Mediant VE SBC features a System Snapshots mechanism that provides the capability of returning the system to a previous state. The mechanism may be used as a rescue option if a system malfunction occurs.

For more information, refer to the Section 'Rescue Options using System Snapshots' in the *Mediant Software SBC User's Manual*.

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
80 Kingsbridge Rd
Piscataway, NJ 08854, USA
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide
Website: https://www.audiocodes.com

Document #: LTRT-10899