

Mediant Virtual Edition (VE) SBC

Deployment in Microsoft Azure

Version 7.6



Table of Contents

1	Introduction	7
2	Deployment Topologies.....	9
2.1	Standalone Topology	9
2.2	High-Availability (HA) Topology	10
2.2.1	Example HA Topologies	12
2.2.2	Networking Environment for Example Topologies	12
2.2.3	Example #1: Dedicated Network Interface for HA Traffic, Single Public Interface for All the Rest	13
2.2.3.1	Configuration Parameters	13
2.2.3.2	Deployment Details	13
2.2.3.3	Detailed Description	13
2.2.3.4	Summary and Limitations.....	14
2.2.4	Example #2: Dedicated Network Interfaces for HA, Management, Trusted and Untrusted Traffic	15
2.2.4.1	Configuration Parameters	15
2.2.4.2	Deployment Details	15
2.2.4.3	Detailed Description	16
2.2.4.4	Summary and Limitations.....	17
2.2.5	Example #3: No Dedicated Network Interface for HA, Two Interfaces for Trusted and Untrusted Traffic	18
2.2.5.1	Configuration Parameters	18
2.2.5.2	Deployment Details	18
2.2.5.3	Detailed Description	19
2.2.5.4	Summary and Limitations.....	19
3	Deployment Methods	21
4	Deploying Standalone Mediant VE via Azure Marketplace / Portal.....	23
4.1	Deleting a Deployed Mediant VE	28
5	Deploying Standalone Mediant VE via PowerShell CLI	29
5.1	Installing Azure PowerShell CLI.....	29
5.2	Deploying a Mediant VE.....	29
5.3	Deleting a Deployed Mediant VE	32
6	Deploying Mediant VE via Stack Manager	33
6.1	Installing Stack Manager.....	33
6.2	Deploying a Mediant VE.....	33
6.2.1	Deploying a Mediant VE as Container Image	38
6.3	Resources Created by Stack Manager	40
6.4	Public IP Addresses	41
6.5	Private IP Addresses.....	42
6.6	Security Groups	44
6.6.1	Default Security Groups.....	44
6.6.2	Adjusting Default Security Groups.....	45
6.6.3	Using Custom Security Groups	46
7	Changing Network Configuration After Deployment	47
7.1	Changing Network Configuration for Standalone Mediant VE Deployments Performed via Azure Portal or CLI	47

7.2 Adding Network Interface to Standalone Mediant VE Deployed via Azure Portal or CLI 48

7.3 Deleting the Network Interface from Standalone Mediant VE Deployed via Azure Portal or CLI 51

8 Upgrading the Software Version.....53

8.1 Upgrading the Software Version for Container Deployments 55

9 Licensing the Product.....57

9.1 Obtaining and Activating a Purchased License Key 57

9.2 Installing the License Key 59

9.3 Product Key..... 61

List of Figures

Figure 2-1: Standalone Topology9

Figure 2-2: HA Topology – Principles of Operation.....10

Figure 4-1: Azure Marketplace23

Figure 4-2: Basics Step24

Figure 4-3: Virtual Machine Settings Step.....25

Figure 4-4: Network Settings Step26

Figure 4-5: Review + Create Step.....27

Figure 4-6: Determining IP Address of Deployed VM28

Figure 6-1: Stack Manager Tool.....33

Figure 6-2: Stack Manager: Create Stack Dialog – Step 134

Figure 6-3: Stack Manager: Create Stack Dialog – Step 234

Figure 6-4: Stack Manager: Create Stack Dialog – Step 335

Figure 6-5: Stack Manager: Create Stack Dialog – Step 436

Figure 6-6: Stack Manager: Successful Stack Creation.....37

Figure 6-7: Stack Manager Tool.....38

Figure 6-8: Stack Manager: Files Screen.....38

Figure 6-9: Stack Manager: Add File Dialog39

Figure 6-10: Stack Manager: File Added.....39

Table 6-11: Assignment of Default Security Groups44

Table 6-12: Inbound Rules for Default Security Groups44

Figure 7-1: New Physical Ports Configuration Object49

Figure 7-2: New Ethernet Device (VLAN) Configuration.....49

Figure 7-3: New IP Interface Configuration50

Figure 7-4: New Network Configuration50

Figure 7-5: Remaining Network Configuration Objects.....52

Figure 7-6: Deleing Remaining IP Interface52

Figure 7-7: Saving the updated configuration52

Figure 8-1: Opening Web Interface's Software Upgrade Wizard53

Figure 8-2: Stack Manager: Add File Dialog54

Figure 8-3: Stack Manager: CMP File Upload.....54

Figure 8-4: Stack Manager: Upgrade Dialog.....54

Figure 8-5: Stack Manager: Upgrade Dialog.....55

Figure 9-1: Software License Activation Tool.....57

Figure 9-2: Product Key in Order Confirmation E-mail.....58

Figure 9-3: Apply New License Key Message.....59

Figure 9-4: Reset in Progress for License Key60

Figure 9-5: Reset and Save-to-Flash Success Message60

Figure 9-6: Viewing Product Key61

Figure 9-7: Empty Product Key Field61

Figure 6-9-8: Entering Product Key61

Notice

Information contained in this document is believed to be accurate and reliable at the time of publishing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of published 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: April-08-2025

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.

Document Revision Record

LTRT	Description
11018	Initial document release for Version 7.6

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

This document describes the deployment of AudioCodes' Mediant Virtual Edition (VE) Session Border Controller (SBC), hereafter referred to as *Mediant VE*, in a Microsoft Azure environment.

For detailed instructions on how to install Mediant VE in other virtual environments, for example, VMware, refer to their respective documents.

**Note:**

- The scope of this document does not fully cover security aspects for deploying the product in the Microsoft Azure cloud. Security measures should be carried out in accordance with Azure security policies and recommendations.
- For configuring the Mediant VE, refer to the *Mediant Software SBC User's Manual*.

This page is intentionally left blank.

2 Deployment Topologies

Mediant VE SBC may be deployed on the Azure platform in one of the following topologies:

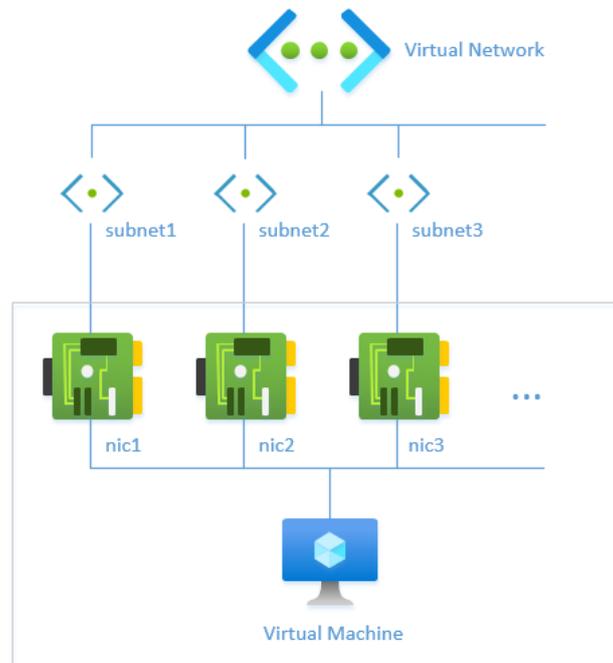
- **Standalone Topology:** Mediant VE SBC is deployed on a single virtual machine.
- **High-Availability (HA) Topology:** Mediant VE SBC is deployed on two virtual machines that operate in active/standby mode. The Azure Load Balancer is used to steer management and signaling traffic towards the active virtual machine.

2.1 Standalone Topology

Standalone topology deploys the SBC software on a single virtual machine. Refer to the *Release Notes* for a list of supported virtual machine sizes.

Up to eight network interfaces are supported, depending on the virtual machine size. The first network interface (eth0) is used for management traffic (e.g., HTTP and SSH). All network interfaces (including the first one) may be used for signaling (SIP) and media (RTP/RTCP) traffic.

Figure 2-1: Standalone Topology

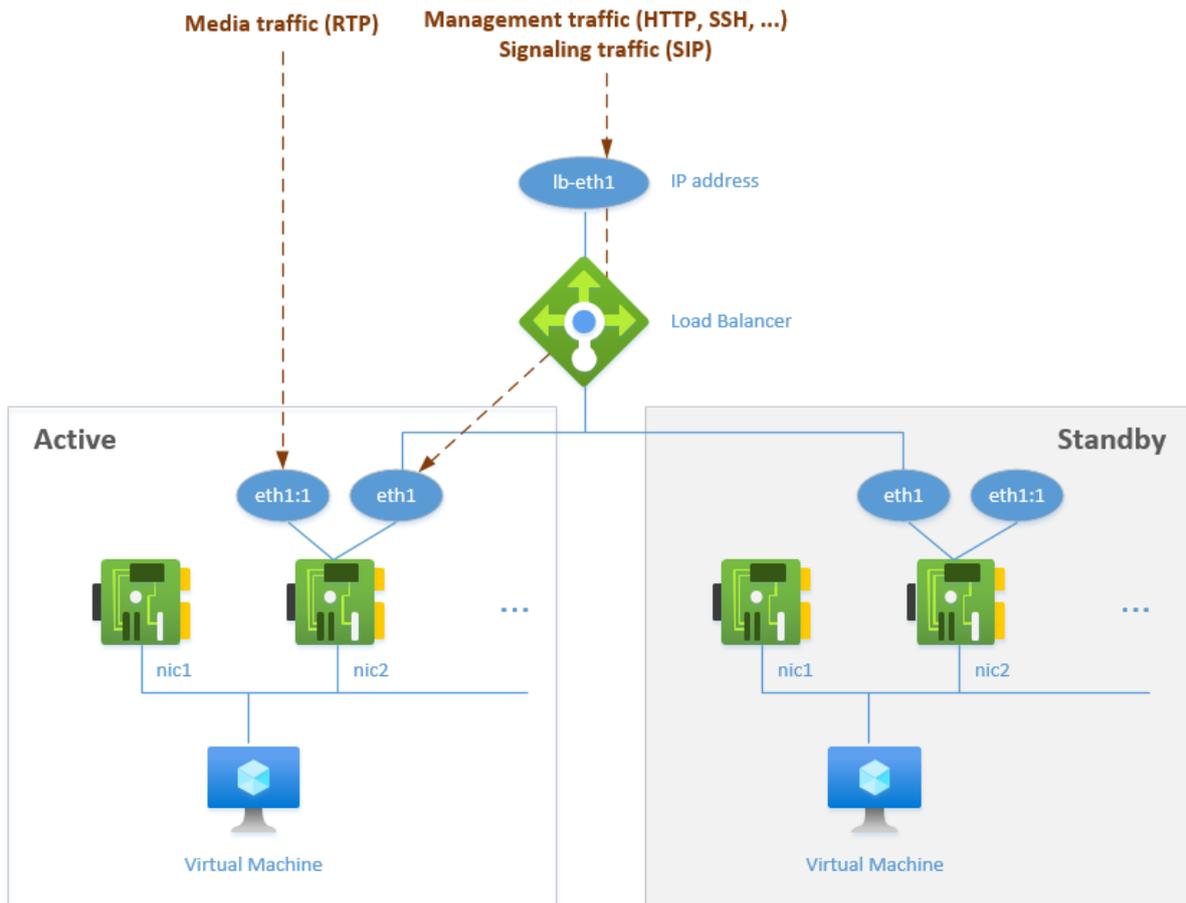


2.2 High-Availability (HA) Topology

HA topology deploys the SBC software on two virtual machines that operate in active/standby mode. Azure Load Balancer is deployed in front of these virtual machines and is used to steer signaling and management traffic towards the active instance. Media traffic bypasses the Load Balancer and flows directly to the active instance through its local IP addresses.

The following figure demonstrates the principles of HA topology operation. In the example, the second network interface (nic2) on each virtual machine is assigned two IP addresses. The first (primary) IP address (eth1) is located behind Azure Load Balancer and is used for signaling and management traffic. The second (secondary) IP address (eth1:1) is not located behind Azure Load Balancer and is used for media traffic.

Figure 2-2: HA Topology – Principles of Operation



External equipment (e.g., IP-PBX or SIP Trunk) should be configured to send signaling (SIP) traffic to the IP address of the Load Balancer. The latter “passes” it to the active SBC instance. During call establishment, the SBC “publishes” its secondary IP address in the SDP body, thus latching the media stream to it. If a switchover occurs, the newly active SBC instance relatches the media streams (using SIP re-INVITE requests) to the new media IP addresses.



Note: External equipment (e.g., IP-PBX or SIP Trunk) must support symmetric response routing (RFC 3581) for proper media relatching during a switchover.

HA topology supports both public and private IP addresses. It creates a Public or Internal Load Balancer in front of the virtual machines accordingly and assigns public IP addresses to secondary IP addresses, if needed. However, there is one important difference in these use cases:

- When the Public Load Balancer is used, it also functions as a NAT Gateway for outbound flows. Therefore, when the SBC software sends an outbound request (e.g., SIP INVITE), external equipment (e.g., IP-PBX or SIP Trunk) receives packets with the IP address of the Load Balancer in the source header.
- When the Internal Load Balancer is used, no NAT translation is performed for outbound flows. Therefore, when the SBC software sends an outbound request (e.g., SIP INVITE), external equipment sees the traffic coming from the IP address of the active SBC instance. This address changes upon a switchover. External equipment should use the Via and Contact SIP headers, containing the IP address of the Load Balancer, for incoming requests, classification, and as a destination address for new SIP dialogs.

Two virtual machines are deployed by default in the proximity group and availability set with two fault and update domains. Alternatively, you may deploy them in two different availability zones.

2.2.1 Example HA Topologies

HA deployment topology varies, depending on the configuration parameters provided during initial stack deployment. The following sections describe the most common HA deployment topologies; these are only examples, which are by no means exhaustive. See Section 6 for a detailed description of the deployment procedure and for supported configuration parameters.

It is recommended to review all examples to ensure that you understand how implementation is done, understanding the pros and cons of each deployment option. Nevertheless, for most deployments, Example #3 in Section 2.2.5 is the most flexible and cost-effective option.

2.2.2 Networking Environment for Example Topologies

The subsequent sections assume that Mediant VE is deployed in the following networking environment:

Resource Name	Resource Type	Address Range	Description
vnet1	Virtual Network	10.0.0.0/16	Virtual network where Mediant VE is deployed.
ha	Subnet	10.0.0.0/24	(Optional) Dedicated subnet for HA traffic.
main	Subnet	10.0.1.0/24	Main subnet that carries management traffic. Optionally, this can also carry signaling (SIP) and media (RTP) traffic.
voip1	Subnet	10.0.2.0/24	(Optional) 1st additional subnet for signaling (SIP) and media (RTP) traffic.
voip2	Subnet	10.0.3.0/24	(Optional) 1st additional subnet for signaling (SIP) and media (RTP) traffic.

2.2.3 Example #1: Dedicated Network Interface for HA Traffic, Single Public Interface for All the Rest

This is the most basic example, which uses a dedicated network interface for HA traffic and a single network interface with public IP addresses for all the rest.

2.2.3.1 Configuration Parameters

```
Virtual network:      vnet1
HA subnet:           ha
Main subnet:         main
1st Additional subnet: none
2nd Additional subnet: none
Public IPs:          main subnet
```

2.2.3.2 Deployment Details

```
sbc-1 - 1st virtual machine
  eth0
    10.0.0.11 - type: Maintenance
  eth1
    10.0.1.11 - type: O+M+C - resides behind public load balancer
    10.0.1.12 - type: M+C   - has public IP 20.3.5.11 attached

sbc-2 - 2nd virtual machine
  eth0
    10.0.0.21 - type: Maintenance
  eth1
    10.0.1.21 - type: O+M+C - resides behind public load balancer
    10.0.1.22 - type: M+C   - has public IP 20.3.5.21 attached

lb - public Load Balancer
  frontend IPs:
    eth1 20.3.5.31
    backends: sbc-1/eth1/10.0.1.11, sbc-2/eth1/10.0.1.21
  routing rules: for management and signaling ports
```

2.2.3.3 Detailed Description

The first network interface (eth0) is used for internal communication between virtual machines.

The second network interface (eth1) is used for management (HTTP, SSH, etc.), signaling (SIP) and media (RTP) traffic. This interface has two IP addresses:

- The first address resides behind the Public Azure Load Balancer and is used for management and signaling traffic.
- The second address is local and is used for media traffic.

Management and signaling traffic are sent towards the front-end IP address of the Public Load Balancer (20.3.5.31). If the sbc-1 instance is currently active, the traffic is then sent towards the first IP address on eth1 interface of the active SBC instance (10.0.1.11).

Outbound NAT translation rules on Public Load Balancer make outbound flows (e.g., outbound INVITE messages) appear as “coming” from the same IP address (20.3.5.31).

Media traffic is sent directly towards the active SBC instance through the second IP address on eth1 interface. These addresses (on both SBC instances) have local public IP addresses assigned to them to ensure that they are reachable over the public internet. If the sbc-1 instance is currently active, media streams are sent towards the public IP address 20.3.5.11 attached directly to its local IP address 10.0.1.12.

If the sbc-1 instance fails, the following events occur:

- The sbc-2 instance discovers that the sbc-1 instance is not responding to keep-alives and assumes the “active” role.
- The sbc-2 instance sends re-INVITE requests to “relatch” all media streams to the public IP address 20.3.5.21, attached to its local IP address 10.0.1.22.
- Azure Load Balancer discovers that the sbc-1 instance is “dead” and the sbc-2 instance is “alive”, and starts “steering” management and signaling traffic towards the sbc-2 instance.

2.2.3.4 Summary and Limitations

This example created a single public interface for all types of traffic. This implies that **all** communication with Mediant VE, including management traffic, must be done through the public IP addresses. As such, it is mostly suited for PoC and lab deployments.

Refer to the following examples for more “production-grade” examples that use internal IP addresses for management traffic.

2.2.4 Example #2: Dedicated Network Interfaces for HA, Management, Trusted and Untrusted Traffic

This example uses dedicated network interfaces for each traffic type:

- HA traffic (communication between SBC instances)
- Management traffic (HTTP, SSH, etc.) – through internal IP addresses
- Trusted VoIP traffic (signaling and media) – through internal IP addresses
- Untrusted VoIP traffic (signaling and media) – through public IP addresses

2.2.4.1 Configuration Parameters

```
Virtual network:      vnet1

HA subnet:           ha
Main subnet:         main
1st Additional subnet: voip1
2nd Additional subnet: voip2

Public IPs:          2nd additional subnet
```

2.2.4.2 Deployment Details

```
sbc-1 - 1st virtual machine
  eth0
    10.0.0.11 - type: Maintenance
  eth1
    10.0.1.11 - type: O+M+C - resides behind internal load balancer
    10.0.1.12 - type: M+C - not used
  eth2
    10.0.2.11 - type: M+C - resides behind internal load balancer
    10.0.2.12 - type: M+C
  eth3
    10.0.3.11 - type: M+C - resides behind public load balancer
    10.0.3.12 - type: M+C - has public IP 20.3.5.11 attached

sbc-2 - 2nd virtual machine
  eth0
    10.0.0.21 - type: Maintenance
  eth1
    10.0.1.21 - type: O+M+C - resides behind internal load balancer
    10.0.1.22 - type: M+C - not used
  eth2
    10.0.2.21 - type: M+C - resides behind internal load balancer
    10.0.2.22 - type: M+C
  eth3
    10.0.3.21 - type: M+C - resides behind public load balancer
    10.0.3.22 - type: M+C - has public IP 20.3.5.21 attached

intlbal - internal Load Balancer
  frontend IPs:
    eth1 10.0.1.31
```

```

        backends: sbc-1/eth1/10.0.1.11, sbc-2/eth1/10.0.1.21
        routing rules: for management ports
eth2 10.0.2.31
        backends: sbc-1/eth2/10.0.2.11, sbc-2/eth2/10.0.2.21
        routing rules: for signaling ports

lb - public Load Balancer
  frontend IPs:
    eth1 20.3.5.31
        backends: sbc-1/eth3/10.0.3.11, sbc-2/eth3/10.0.3.21
        routing rules: for signaling ports

```

2.2.4.3 Detailed Description

The first network interface (eth0) is used for internal communication between virtual machines.

The second network interface (eth1) is used for management (HTTP, SSH, etc.) traffic. Its first IP address resides behind the Internal Azure Load Balancer. Management traffic is sent towards the front-end address of the Internal Load Balancer (10.0.1.31). If the sbc-1 instance is currently active, the traffic is then sent towards the first IP address on eth1 interface of the active SBC instance (10.0.1.11). Note that the Internal Azure Load Balancer doesn't perform outbound NAT translation and therefore, outbound flows (e.g., Syslog packets) appear as "coming" from the local IP address of active instance (10.0.1.11) if the sbc-1 instance is currently active.

The third and fourth network interfaces (eth2 and eth3) are used for signaling (SIP) and media (RTP) traffic. Their first IP addresses reside behind the Internal and Public Azure Load Balancers, respectively. Signaling traffic is sent towards the front-end address of the Load Balancer (10.0.2.31) for trusted traffic and 20.35.5.31 for untrusted traffic. If the sbc-1 instance is currently active, the traffic is then sent towards the first IP address on eth2 and eth3 interfaces of the active SBC instance (10.0.2.11 and 10.0.3.11, respectively).

- For trusted traffic that uses internal IP addresses, outbound flows (e.g., outbound INVITE requests) appear as "coming" from the local IP address of the active instance (10.0.2.11) if the sbc-1 instance is currently active. This is because Internal Load Balancer doesn't perform outbound NAT translation.
- For untrusted traffic that uses public IP addresses, Public Azure Load Balancer performs outbound NAT translation and outbound flows appear as "coming" from Load Balancer's front-end IP address (20.35.5.31).

Media traffic is sent directly towards the active SBC instance through the second IP address on the eth2 and eth3 interfaces. For untrusted traffic, these addresses have local public IP addresses assigned to them to ensure that they are reachable over the public internet. If the sbc-1 instance is currently active:

- Media streams for trusted traffic are sent towards the internal IP address 10.0.2.12.
- Media streams for untrusted traffic are sent towards the public IP address 20.3.5.11 attached to its local IP address 10.0.3.12.

If the sbc-1 instance fails, the following events occur:

- The sbc-2 instance discovers that the sbc-1 instance is not responding to keep-alives and assumes the "active" role.
- The sbc-2 instance sends re-INVITE requests to "relatch" all trusted media streams to its local IP address 10.0.2.22 and all untrusted media streams to the public IP address 20.3.5.21, attached to its local IP address 10.0.3.22.
- Azure Load Balancers discover that the sbc-1 instance is "dead" and the sbc-2 instance is "alive" and starts "steering" management and signaling traffic towards the sbc-2 instance.

2.2.4.4 Summary and Limitations

This deployment topology provides complete traffic separation and supports VoIP traffic through both internal and public IP addresses.

It may be easily adapted to handle trusted VoIP traffic through the public IP addresses, while keeping internal IP addresses for management traffic by specifying the following during stack creation:

```
Public IPs:          main and 1st additional subnet
```

This places the first IP addresses on both eth2 and eth3 network interfaces behind the Public Azure Load Balancer, while keeping the first IP address on the eth1 network interface behind the Internal Azure Load Balancer.

The main drawback of this deployment topology is the need to use four network interfaces. This implies that the SBC instances must use relatively large and expensive VM sizes (e.g., D8ds_v5).

Refer to the next example for a configuration that requires only two network interfaces, but still achieves a similar level of connectivity and traffic separation.

2.2.5 Example #3: No Dedicated Network Interface for HA, Two Interfaces for Trusted and Untrusted Traffic

This example uses only two interfaces:

- eth0 is used for management, trusted VoIP, and HA traffic.
- eth1 is used for untrusted VoIP traffic.

2.2.5.1 Configuration Parameters

```
Virtual network:      vnet1
HA subnet:           none
Main subnet:         main
1st Additional subnet: voip1
2nd Additional subnet: none
Public IPs:          1st additional subnet
```

2.2.5.2 Deployment Details

```
sbc-1 - 1st virtual machine
  eth0
    10.0.1.11 - type: O+M+C - resides behind internal load balancer
    10.0.1.12 - type: M+C
    10.0.1.13 - type: Maintenance
  eth1
    10.0.2.11 - type: M+C - resides behind public load balancer
    10.0.2.12 - type: M+C - has public IP 20.3.5.11 attached

sbc-2 - 2nd virtual machine
  eth0
    10.0.1.21 - type: O+M+C - resides behind internal load balancer
    10.0.1.22 - type: M+C
    10.0.1.23 - type: Maintenance
  eth2
    10.0.2.21 - type: M+C - resides behind public load balancer
    10.0.2.22 - type: M+C - has public IP 20.3.5.21 attached

intlb - internal Load Balancer
  frontend IPs:
    eth0 10.0.1.31
    backends: sbc-1/eth0/10.0.1.11, sbc-2/eth0/10.0.1.21
    routing rules: for management and signaling ports

lb - public Load Balancer
  frontend IPs:
    eth1 20.3.5.31
    backends: sbc-1/eth1/10.0.2.11, sbc-2/eth1/10.0.2.21
    routing rules: for signaling ports
```

2.2.5.3 Detailed Description

The first network interface (eth0) is used for:

- Management traffic (HTTP, SSH, etc.)
- Trusted VoIP traffic (SIP and RTP – through internal IP addresses)
- Maintenance (HA) traffic between SBC instances

Its first IP address resides behind the Internal Azure Load Balancer. Management and trusted signaling traffic are sent towards the front-end address of the Internal Load Balancer (10.0.1.31). If the sbc-1 instance is currently active, the traffic is then sent towards the first IP address of the active SBC instance (10.0.1.11). Note that Internal Azure Load Balancer doesn't perform outbound NAT translation and therefore, outbound flows (e.g., outbound INVITE requests) appear as "coming" from the local IP address of the active instance (10.0.1.11) if the sbc-1 instance is currently active.

The third IP address on the eth0 network interface is used for internal communication between SBC instances.

The second network interface eth1 is used for untrusted signaling traffic. Its first IP address resides behind the Public Azure Load Balancer. Untrusted signaling traffic is sent towards the front-end address of the Load Balancer (20.35.5.31). If the sbc-1 instance is currently active, the traffic is then sent towards the first IP address on eth1 interface of the active SBC instance (10.0.2.11). Public Azure Load Balancer performs outbound NAT translation and outbound flows appear as "coming" from Load Balancer's front-end IP address (20.35.5.31).

Media traffic is sent directly towards the active SBC instance through the second IP address on eth0 and eth1 interfaces. For untrusted traffic (on eth1 interface), these addresses have local public IP addresses assigned to them to ensure that they are reachable over the public internet. If the sbc-1 instance is currently active:

- Media streams for trusted traffic are sent towards the internal IP address 10.0.1.12.
- Media streams for untrusted traffic are sent towards the public IP address 20.3.5.11 attached to its local IP address 10.0.2.12.

If the sbc-1 instance fails, the following events occur:

- The sbc-2 instance discovers that the sbc-1 instance is not responding to keep-alives and assumes the "active" role.
- The sbc-2 instance sends re-INVITE requests to "relatch" all trusted media streams to its local IP address 10.0.1.22 and all untrusted media streams to the public IP address 20.3.5.21, attached to its local IP address 10.0.2.22.
- Azure Load Balancers discover that the sbc-1 instance is "dead" and the sbc-2 instance is "alive", and starts "steering" management and signaling traffic towards the sbc-2 instance.

2.2.5.4 Summary and Limitations

This deployment topology uses only two network interfaces and therefore, allows the use of very small and cost-effective VM sizes for SBC instances (e.g., DS1_v2 or D2ds_v5). It uses internal IP addresses for management traffic and supports both public and internal IP addresses for VoIP traffic.

The topology may be easily adapted to handle trusted VoIP traffic through the public IP addresses, while keeping internal IP addresses for management traffic by specifying the following during stack creation:

```
Public IPs:                main and 1st additional
subnet
Use private IP address for management:  checked
```

This implements the following changes:

- Place the first IP addresses on the eth0 network interface behind the Public Azure Load Balancer.
- Create an additional pair of IP addresses on the eth0 network interface and place the first of them behind Internal Azure Load Balancer. This new address is used for management traffic.

The only drawback of this deployment topology is the lack of separation of different traffic types into different subnets. If such separation is important, then use the deployment topology similar to Example #2 in Section 2.2.4.

3 Deployment Methods

You can deploy Mediant VE in the Microsoft Azure cloud environment, using one of the following methods:

- **Standalone topology:**
 - Microsoft Azure Marketplace / Portal (see Section 4 on page 23)
 - PowerShell CLI (see Section 5 on page 29)
 - Stack Manager tool (see Section 6 on page 33)
- **HA topology:**
 - Stack Manager tool (see Section 6 on page 33)

Starting with Version 7.6, Mediant VE software is available in the following deployment modes:

- **Virtual Appliance:** A fully integrated virtual machine image that includes the Mediant VE software along with a fully secured and highly optimized Linux kernel. This mode does not provide access to the underlying Linux OS and does not support the installation of additional software or third-party tools on the same virtual machine.
- **Container Image:** Deployed on a standard Linux OS host and run by a Docker engine. This mode allows for the installation of additional software and third-party tools on the same virtual machine. However, it may offer slightly lower performance compared to the Virtual Appliance due to the absence of kernel-level optimizations

The rest of this document mostly refers to the deployment of Mediant VE software as a **virtual appliance**.

If you need the deployment of Mediant VE software as a **container image**, refer to *Mediant VE Container Installation Manual*.

This page is intentionally left blank.

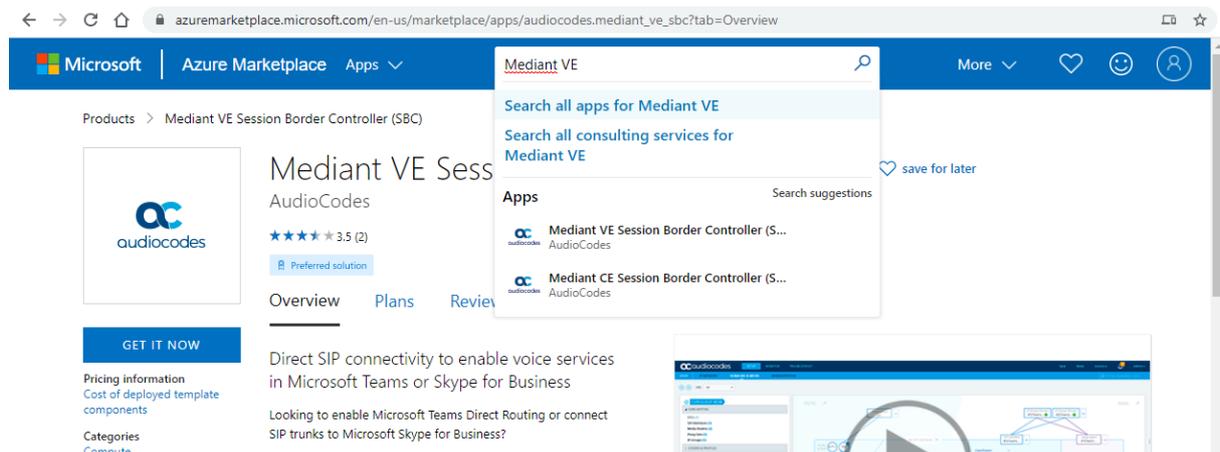
4 Deploying Standalone Mediant VE via Azure Marketplace / Portal

This section describes the deployment of a standalone Mediant VE through the Azure Marketplace / portal. This deployment method provides graphical user interface and is therefore, most suited if you are not familiar with the Azure cloud environment.

➤ **To deploy a standalone Mediant VE through Azure Marketplace / Portal:**

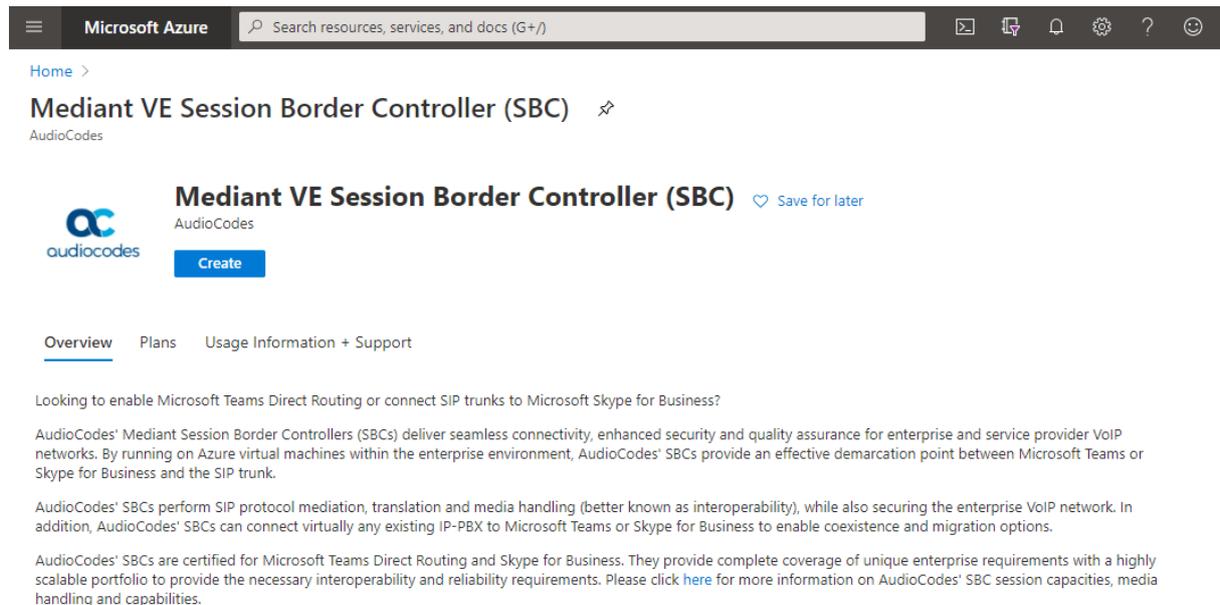
1. Open the Azure Marketplace at <https://azuremarketplace.microsoft.com/>.
2. Search for the product "Mediant VE Session Border Controller (SBC)" published by AudioCodes.

Figure 4-1: Azure Marketplace



3. Click **GET IT NOW**; the Azure portal and Mediant VE SBC Product Overview screen appears:

Figure 4-2: Mediant VE SBC Product Overview



4. Click **Create** to start a new Mediant VE deployment; the Create AudioCodes Mediant VE SBC for Microsoft Azure dialog appears. The dialog contains multiple steps. Complete each step according to the description below.

- In the **Basics** step, do the following:

Figure 4-2: Basics Step

The screenshot shows the 'Basics Step' of the Azure portal for creating a Mediant VE Session Border Controller (SBC). The breadcrumb path is 'Home > Mediant VE Session Border Controller (SBC) >'. The main heading is 'Create Mediant VE Session Border Controller (SBC)'. Below the heading are four tabs: 'Basics' (selected), 'Virtual Machine Settings', 'Network Settings', and 'Review + create'. Under 'Project details', there is a description: 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' The form fields are as follows:

- Subscription ***: A dropdown menu with 'SBC Lab' selected.
- Resource group ***: A dropdown menu with '(New) sbc-test1' selected. Below it is a blue link labeled 'Create new'.
- Instance details**:
 - Region ***: A dropdown menu with 'West US 2' selected.
 - Virtual Machine name ***: A text input field with 'sbc-test1' and a green checkmark.
 - Username ***: A text input field with 'sbcadmin' and a green checkmark.
 - Authentication type ***: Two radio buttons: 'Password' (selected) and 'SSH Public Key'.
 - Password ***: A password input field with masked characters and a green checkmark.
 - Confirm password ***: A password input field with masked characters and a green checkmark.

At the bottom of the form, there are three buttons: a blue 'Review + create' button, a grey '< Previous' button, and a grey 'Next : Virtual Machine Settings >' button.

- In the 'Subscription' field, select a proper subscription for your deployment.
- In the 'Resource group' field, click **Create new** and then enter a unique name for the new resource group. Alternatively, you may select an existing empty resource group from the list.
- In the 'Region' field, select a proper region for your deployment.
- In the 'Virtual Machine name' field, enter a unique name for the new VM.
- In the 'Username' field, enter a username.
- For 'Authentication type', select **Password**.
- In the 'Password' field, enter a password, and then enter it again in the 'Confirm password' field.

These credentials are used to connect to the management interface of the deployed Mediant VE (instead of the default **Admin/Admin** credentials, as used in other environments).



Note: Azure imposes some limitations on username and password. For example, it prohibits the use of "Admin" for username and requires the use of strong passwords that meet the following policy:

- A minimum of 12 characters
- Use of three out of four of the following: lowercase characters, uppercase characters, numbers, and symbols

- h. Click **OK**.
6. In the **Virtual Machine Settings** step, do the following:
 - a. Choose the virtual machine size. For a list of supported virtual machine sizes and corresponding capacity figures, refer to the *SBC-Gateway-MSBR Series Release Notes*.
 - b. Choose the disk type for the virtual machine. As SBC software typically doesn't perform extensive disk activity, **Standard HDD** disk type is adequate for most deployments.
 - c. Choose the OS version for the deployed SBC software:
 - ◆ **6** – this version corresponds to the 7.20A stream, which is based on CentOS 6.
 - ◆ **8** – this version corresponds to the new 7.40A stream, which is based on CentOS Stream 8 / Rocky Linux 8 and provides significantly better performance and capacity (refer to the *SBC-Gateway Series Release Notes* for details).
 - d. Choose whether to enable virtual machine's boot diagnostics.
 - e. Optionally, provide Mediant VE automatic configuration script (cloud-init file). For more information, refer to the *Automatic Provisioning of Mediant VE SBC via Cloud-Init Configuration Note*.
 - f. Click **OK**.

Figure 4-3: Virtual Machine Settings Step

The screenshot shows the 'Virtual Machine Settings' step in the Azure portal. The breadcrumb navigation is 'Home > Mediant VE Session Border Controller (SBC) >'. The main heading is 'Create Mediant VE Session Border Controller (SBC)'. Below the heading are four tabs: 'Basics', 'Virtual Machine Settings' (which is selected), 'Network Settings', and 'Review + create'. The 'Virtual Machine Settings' tab contains the following configuration options:

- Virtual machine size ***: 1x Standard DS1 v2 (1 vcpu, 3.5 GB memory). A 'Change size' link is available.
- Disk type**: Standard HDD (selected from a dropdown menu).
- OS version**: 6 (selected with a radio button), 8 (unselected).
- Boot diagnostics**: Enable (selected with a radio button), Disable (unselected).
- SBC cloud-init file**: Select a file (with a file upload icon).

At the bottom of the form, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Network Settings >'.

7. In the **Network Settings** step, do the following:
 - a. Choose the number of network interfaces for the new virtual machine. Deployment via Azure Marketplace supports up to two network interfaces. If you need more interfaces, perform deployment via the PowerShell CLI, as described in Chapter 5.
 - b. Configure the virtual network where the new VM will be deployed. You may either create a new virtual network or select an existing one. Azure virtual machine is always connected to a single virtual network, regardless of the number of its network interfaces.
 - c. Configure the subnet for each network interface. You may either create a new subnet (for new virtual network) or select an existing one.
 - ◆ If you choose two network interfaces, you must connect each interface to a different subnet. This is a limitation of Azure Marketplace UI and may be overcome by performing the deployment via the PowerShell CLI, as described in Chapter 5.
 - ◆ If you choose two network interfaces, you can access the SBC management interfaces (Web and SSH) through the 1st network interface only.
 - d. Configure the virtual machine's Public IP Address. You may either create a new Public IP Address or select an existing one.
 - ◆ If you create a new Public IP Address, select **Static Assignment**. This ensures that the IP address remains unchanged if you stop the virtual machine.
 - ◆ If you choose two network interfaces, the public IP address will be attached to the 1st network interface.
 - e. Click **OK**

Figure 4-4: Network Settings Step

The screenshot shows the 'Network Settings' step in the Azure portal. At the top, there is a search bar and navigation links. The main heading is 'Create Mediant VE Session Border Controller (SBC)'. Below this, there are tabs for 'Basics', 'Virtual Machine Settings', 'Network Settings' (which is active), and 'Review + create'. The 'Number of network interfaces' is set to 1. Under 'Configure virtual networks', the 'Virtual network' is set to 'VnetWestUS2' and the 'Subnet' is set to 'oam (10.23.0.0/24)'. The 'Public IP Address' is set to '(new) sbc-test1-ip' and the 'Public DNS Prefix' is set to 'sbc-test1-9e3f9d360c'. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Review + create >'. The URL '.westus2.cloudapp.azure.com' is visible at the bottom right.

8. In the **Review + create** step, review the Mediant VE SBC terms of use and virtual machine configuration, and then click **Create**.

Figure 4-5: Review + Create Step

Microsoft Azure Search resources, services, and docs (G+)

Home > Mediant VE Session Border Controller (SBC) >

Create Mediant VE Session Border Controller (SBC)

Validation Passed

Basics Virtual Machine Settings Network Settings Review + create

PRODUCT DETAILS

Mediant VE Session Border Controller (SBC)
by AudioCodes
[Terms of use](#) | [Privacy policy](#)

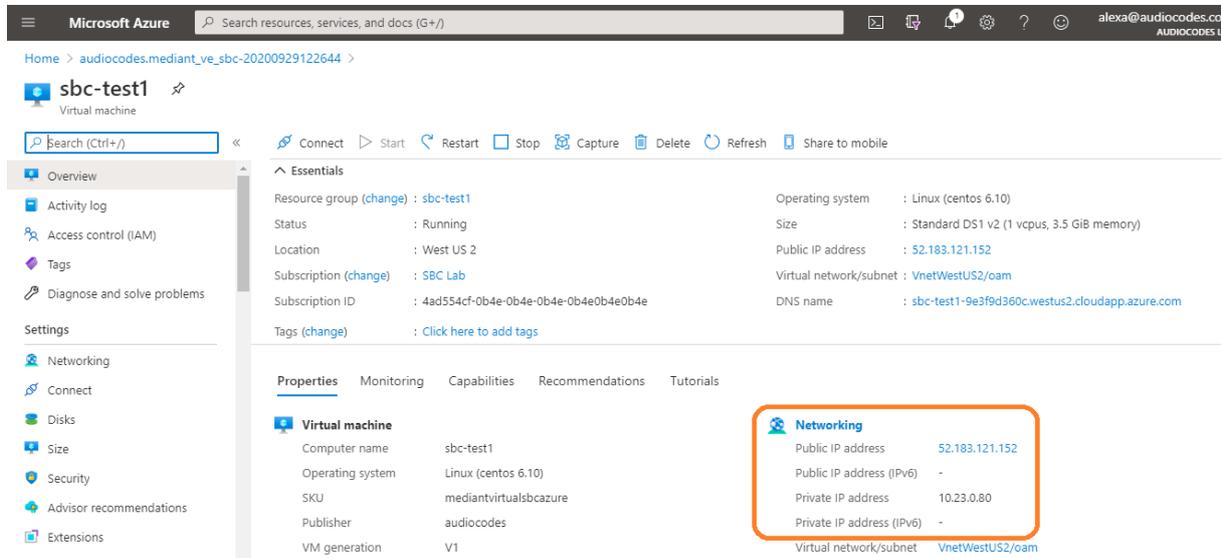
TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Create < Previous Next [Download a template for automation](#)

9. Wait until the virtual machine deployment is complete, and then determine the IP address that is assigned to your virtual machine that can be used to access management interface:
 - ◆ If you assigned a public IP address to the VM, you may use it to access the management interface.
 - ◆ Alternatively, you may use a private IP address of the 1st network interface.

Figure 4-6: Determining IP Address of Deployed VM



10. Log in to the management interface (through Web or SSH) using the credentials that you configured during the virtual machine set up.

4.1 Deleting a Deployed Mediant VE

To delete Mediant VE deployed through the Azure Portal, simply delete the corresponding Resource Group.

5 Deploying Standalone Mediant VE via PowerShell CLI

This section describes the deployment of a standalone Mediant VE via the Azure PowerShell CLI. This deployment method provides maximum flexibility and is therefore, most suited for advanced Azure users who want to exercise full control over their deployment.

5.1 Installing Azure PowerShell CLI

Before you can use the Azure PowerShell CLI, you need to install it.

➤ **To install Azure PowerShell CLI:**

1. Run PowerShell with Administrator privileges.
2. Install Azure PowerShell CLI, using the following commands:

```
Install-Module PowerShellGet -Force
Install-Module -Name Az -AllowClobber
```

5.2 Deploying a Mediant VE

This section describes how to deploy a standalone Mediant VE.

➤ **To deploy a Mediant VE:**

1. Run PowerShell.
2. Sign in to your Azure account, and then select the appropriate subscription:

```
Connect-AzAccount
Select-AzSubscription -SubscriptionName "SBC Lab"
# replace "SBC Lab" with your subscription name
```

3. Get the parameters of the pre-configured virtual network and subnet:

```
$VNetResourceGroupName = "SbcWestUS2"
# replace "SbcWestUS2" with virtual network's resource
# group name
$VNetName = "VnetWestUS2"
# replace "VnetWestUS2" with virtual network name
$SubnetName = "oam"
# replace "oam" with subnet name

$VNet = Get-AzVirtualNetwork -Name $VNetName `
    -ResourceGroupName $VNetResourceGroupName

$Subnet = Get-AzVirtualNetworkSubnetConfig `
    -Name $SubnetName -VirtualNetwork $VNet
```

4. Accept Marketplace terms for Mediant VE SBC offer:

```

$Publisher = "audiocodes"
$Product = "mediantsessionbordercontroller"
$Sku = "mediantvesbcazure"
# for 7.20A image based on OS Version 6 use
# $Sku = "mediantvirtualsbcazure" instead

$Terms = Get-AzMarketplaceTerms -Publisher $Publisher `
        -Product $Product -Name $Sku

Set-AzMarketplaceTerms -Publisher $Publisher `
        -Product $Product -Name $Sku -Terms $Terms -Accept
    
```

5. Create the new Resource Group:

```

$VMName = "sbc-test1"
# replace "sbc-test1" with your VM name
$Location = "WestUS2"
# replace "WestUS2" with your location name

$ResourceGroupName = $VMName + "-rg"

New-AzResourceGroup -Name $ResourceGroupName `
        -Location $Location
    
```

6. Create the new virtual machine configuration:

```

$VMSize = "Standard_DS1_v2"
# replace "Standard_DS1_v2" with VM size

$VM = New-AzVMConfig -VMName $VMName -VMSize $VMSize
    
```

7. Create the new public IP address:

```

$PublicIPName = $VMName + "-ip"

$PublicIP = New-AzPublicIpAddress -Name $PublicIPName `
        -ResourceGroupName $ResourceGroupName `
        -Location $Location -AllocationMethod Static
    
```

8. Create the first network interface:

```

$Interface1Name = $VMName + "-eth0"

$Interface1 = New-AzNetworkInterface -Name $Interface1Name `
        -ResourceGroupName $ResourceGroupName `
        -Location $Location -SubnetId $Subnet.id `
        -PublicIpAddressId $PublicIP.id

Add-AzVMNetworkInterface -VM $VM -Id $Interface1.Id -Primary
    
```

9. Create additional network interfaces if needed (optional):

```
$Interface2Name = $VMName + "-eth1"

$Interface2 = New-AzNetworkInterface -Name $Interface2Name `
    -ResourceGroupName $ResourceGroupName `
    -Location $Location -SubnetId $Subnet.id

Add-AzVMNetworkInterface -VM $VM -Id $Interface2.Id
```

10. Configure the source image:

```
Set-AzVMSourceImage -VM $VM -PublisherName $Publisher `
    -Offer $Product -Skus $Sku -Version latest

Set-AzVMPlan -VM $VM -Publisher audiocodes `
    -Product $Product -Name $Sku
```

11. Configure the managed disk:

```
$DiskName = $VMName + "-disk"

Set-AzVMOSDisk -VM $VM -Name $DiskName `
    -StorageAccountType "Standard_LRS" `
    -CreateOption fromImage -Linux
```

12. Configure the Admin user credentials:

```
$AdminUsername = "sbcadmin"
$AdminPassword = "Admin#123456"

$Credential = New-Object PSCredential $AdminUsername, `
    ($AdminPassword | ConvertTo-SecureString -AsPlainText -Force)

Set-AzVMOperatingSystem -VM $VM -Linux `
    -ComputerName $VMName -Credential $Credential
```

13. Create the new virtual machine:

```
New-AzVM -ResourceGroupName $ResourceGroupName `
    -Location $Location -VM $VM
```

14. Find the public IP address of the new Mediant VE instance:

```
$PublicIP = Get-AzPublicIpAddress -Name $PublicIPName `
    -ResourceGroupName $ResourceGroupName
Write-Output $PublicIP.IpAddress
```

15. Use this IP address to connect to the Mediant VE's management interface through the Web or SSH.

5.3 Deleting a Deployed Mediant VE

To delete Mediant VE deployed via the PowerShell CLI, simply delete the corresponding Resource Group:

```
Remove-AzResourceGroup -Name $ResourceGroupName
```

6 Deploying Mediant VE via Stack Manager

This section describes the deployment of Mediant VE via the Stack Manager tool. This deployment method supports both standalone and HA topologies and provides complete Mediant VE lifecycle management, including update of network topology after the initial deployment, software upgrade, resizing of virtual machines etc.

6.1 Installing Stack Manager

Before you can use the Stack Manager tool, you need to install it. Detailed installation instructions are provided in the [Stack Manager User Manual](#).

6.2 Deploying a Mediant VE

This section describes how to deploy a Mediant VE via Stack Manager.

For simplicity, the Stack Manager's Web interface is described. However, the same task may be completed through the CLI and REST management interfaces. Refer to *Stack Manager User Manual* for details.

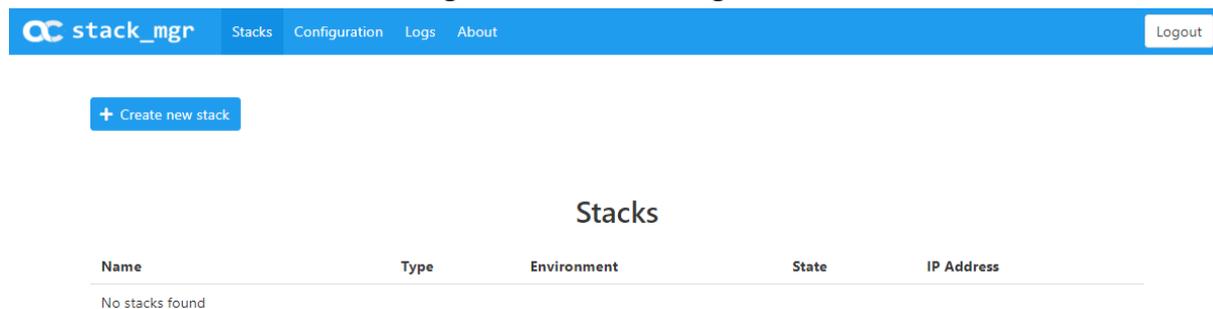


Note: The following instructions describe installation of Mediant VE as a **virtual appliance**. If you want to install Mediant VE as a **container image**, see Section 6.2.1, Deploying a Mediant VE as Container Image.

➤ To deploy a Mediant VE:

1. Log in to Stack Manager.

Figure 6-1: Stack Manager Tool



2. Click **Create new stack**; the Create new stack dialog box appears.

Figure 6-2: Stack Manager: Create Stack Dialog – Step 1

3. In the **Name** field, enter the stack name. Stack Manager creates a resource group with the specified name and uses it as a prefix for all created resources (virtual machines, network interfaces etc.).
4. From the **Stack type** drop-down list, select **Mediant VE**.
5. From the **Region** drop-down list, select the region where Mediant VE is to be deployed.

Figure 6-3: Stack Manager: Create Stack Dialog – Step 2

6. For **HA mode**, select **enable** for HA topology, or **disable** for standalone topology.
7. For **VM type**, Stack Manager automatically determines the appropriate VM type based on the number of configured network interfaces. If you want to use a different VM type, click **Customize** and then select the custom VM type.
8. For **Deployment topology**, select whether you want to deploy Mediant VE instances into **availability set** or across **availability zones**. If you choose the latter, you are prompted to provide the names of two availability zones. Deployment across availability zones provides higher SLA compared to deployment into an availability set (99.99% vs 99.95%). However, a temporary lack of capacity may be experienced for the specific

VM type in the selected availability zones.

9. For **Virtual network**, select the virtual network where Mediant VE is to be deployed.

Figure 6-4: Stack Manager: Create Stack Dialog – Step 3

10. For **HA subnet**, select the **HA** subnet for internal communication between the two SBC instances. The subnet is applicable to HA topology only and is connected through the first network interface (eth0). It is possible to skip this subnet for HA topology, by selecting **-- none--**. In this case, the “main” subnet is used for internal communication between the two SBC instances.
11. For **Main subnet**, select the **main** subnet for management traffic (e.g., HTTP and SSH) and optionally, signaling (SIP) and media (RTP/RTCP) traffic. The subnet is connected to the virtual machine(s) through the first network interface (eth0) for standalone topology or for HA topology without an HA subnet, or through the second network interface (eth1) for HA topology with an HA subnet.
12. For **1st and 2nd Additional subnets**, select “additional” subnets for signaling (SIP) and media (RTP/RTCP) traffic. The subnets are connected to the virtual machine(s) through additional network interfaces: eth2/eth3 for standalone topology or for HA topology without an HA subnet, eth3/eth4 for HA topology with an HA subnet. If you don't need these additional network interfaces, leave it at **-- none --**.
13. For **Public IPs**, select the subnets (and corresponding network interfaces) that must be assigned with public IP addresses.
14. If you assign a public IP address to the Main subnet, Stack Manager by default uses this public IP address for communicating with the deployed stack. You may override this behavior, by checking the **Use private IP address for management** check box. In this case, Stack Manager uses a private IP address to communicate with the deployed stack.

Figure 6-5: Stack Manager: Create Stack Dialog – Step 4

- For **Admin User**, enter the default credentials for Mediant VE management interfaces (Web, SSH, and serial console).



Note: Azure imposes some limitations on the username and password. For example, it prohibits the use of “Admin” for username and requires the use of strong passwords that meet the following policy:

- A minimum of 12 characters.
- Use of three out of four of the following: lowercase characters, uppercase characters, numbers, and symbols.

- Under **Advanced**, configure additional stack parameters. Refer to *Stack Manager User Manual* for a detailed description.

- 17. Click **Create** to start stack creation.
- 18. Wait until the stack creation process completes.

Figure 6-6: Stack Manager: Successful Stack Creation

The screenshot shows the Stack Manager web interface. At the top is a blue navigation bar with the 'stack_mgr' logo and menu items: Stacks, Configuration, Logs, About, Admin, and Logout. Below the navigation bar is a light blue header area containing a '+ Create new stack' button and a search input field. The main content area features a large green box with a vertical green bar on the left, containing the following text: 'Creating stack 'test-ve-1'', 'Initializing Azure client... done', 'Creating resource group 'test-ve-1'... done', 'Creating stack..... done', 'Waiting until SBC is up..... done', 'Waiting until SBC is ready..... done', 'Creating system snapshot... done', 'Use http://20.190.25.218 to connect to the management interface.', 'Stack 'test-ve-1' is successfully created', and 'Done'. Below this box is a section titled 'Stacks' with a table listing the created stack.

Name	Type	Environment	State	Alarms	IP Address
test-ve-1	Mediant VE	Azure	running		20.190.25.218

6.2.1 Deploying a Mediant VE as Container Image

Starting from Version 7.6, you have an option to deploy Mediant VE software in Azure public cloud as a **container image**. In this mode, Mediant VE software is run on a standard Linux host using a Docker engine.

Mediant VE and CE share the same container image, which is delivered as a **docker_debian_sip_F7.60A.xxx.yyy.tar.gz** archive that can be downloaded from AudioCodes [Services Portal](#) - Mediant Virtual Edition (VE), Mediant Cloud Edition (CE) and Mediant Server Edition (SE) Session Border Controllers (SBC).

You need Stack Manager Version 3.8.5 or later to deploy Mediant VE as a container image.

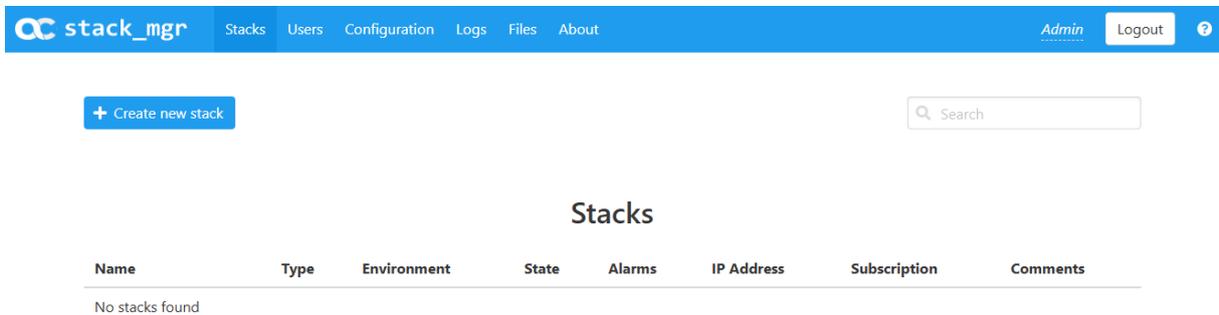
You also need to enable the **docker deployment mode**:

1. Log in to Stack Manager.
2. On the menu bar, click **Configuration**; the Configuration screen appears.
3. Under the **Advanced** group, set 'Docker Deployment Mode' to **Enabled**.
4. Click **Update** to apply the new configuration.

➤ **To deploy a Mediant VE as container image:**

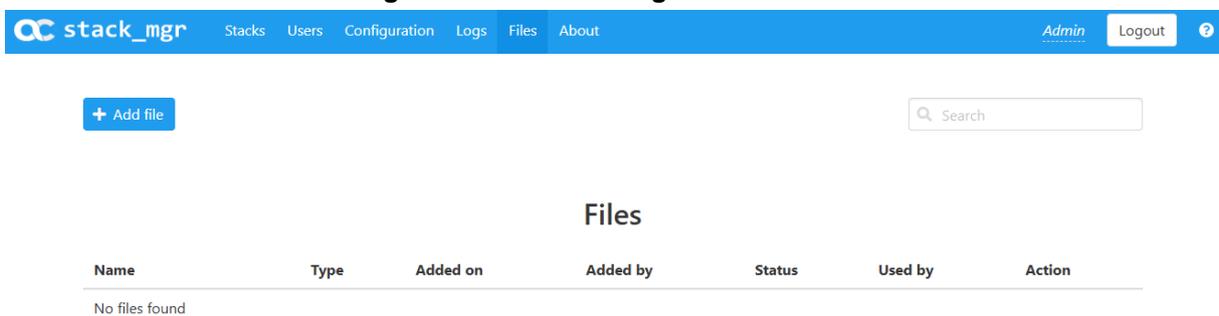
1. Log in to Stack Manager.

Figure 6-7: Stack Manager Tool



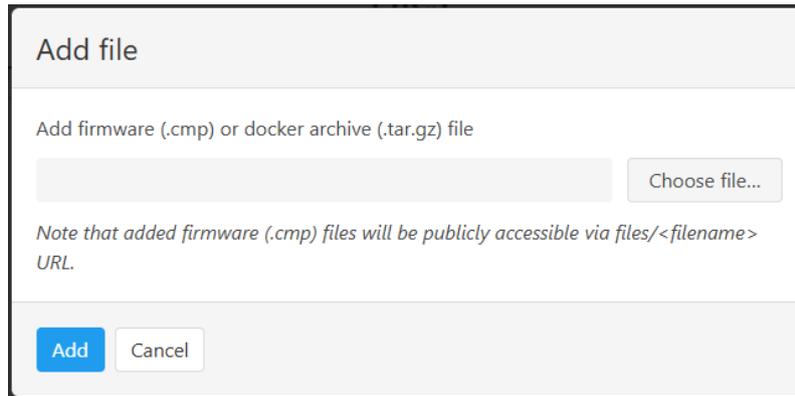
2. On the menu bar, click **Files**; the Files screen appears.

Figure 6-8: Stack Manager: Files Screen



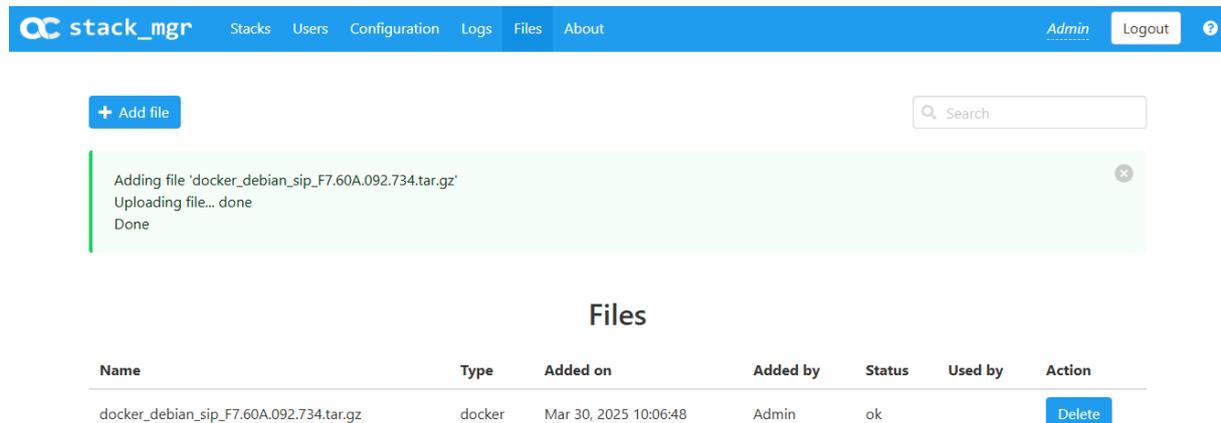
3. Click **Add file**; the Add file dialog box appears.

Figure 6-9: Stack Manager: Add File Dialog



4. Choose the Mediant VE container image file, and then click **Add**; the file is uploaded to Stack Manager and appears in the **Files** list.

Figure 6-10: Stack Manager: File Added



5. On the menu bar, click **Stacks**; the Stacks screen appears.
6. Proceed with Mediant VE installation, as described in Section 6.2, Deploying a Mediant VE with the following changes:
 - In the stack's **Advanced** group, from the 'Deployment mode' drop-down list, select **Docker**.
 - Select the **container image** file that you uploaded in Step 4.

6.3 Resources Created by Stack Manager

The following Azure resources are created by Stack Manager during Mediant VE deployment:

Resource	Description
Resource Group	Resource group into which all stack resources are deployed. This may be overridden by the <code>resource_group</code> advanced config parameter.
Virtual Machines	Virtual machines that run the SBC software.
Disks	Disks attached to the virtual machines.
Network Security Groups	Network security groups for different traffic types. This may be overridden by the <code>ha_nsg_id</code> , <code>main_nsg_id</code> , <code>voip_nsg_id</code> and <code>nsg_id_ethX</code> advanced config parameters.
Storage Account	Storage account for storing VM diagnostics data. This may be overridden by the <code>diag_account</code> advanced config parameter.
Load Balancers	This is applicable only to HA topology. Load balancers are used for steering management and signaling traffic towards the active SBC instance. For Mediant VE Container deployments, NAT rules are also created to access each host machine via SSH interface. These rules forward traffic to port 1022 on backend instances (host machines). The functionality is required during 'upgrade' operation and other maintenance tasks.
Public IPs	Public IP addresses assigned to virtual machines and load balancers. This may be overridden by the <code>public_ip_*</code> or <code>public_ip_prefix</code> advanced config parameters.
Availability Set	This is applicable only to HA topology. Availability set into which virtual machines are deployed. This may be overridden by the <code>use_availability_set</code> advanced config parameter. If the <code>availability_zones</code> advanced config parameter is specified, virtual machines are deployed into two availability zones and an availability set is not created.
Proximity Placement Group	This is applicable only to HA topology. Proximity placement group into which virtual machines are deployed. This may be overridden by the <code>use_proximity_placement_group</code> advanced config parameter. If the <code>availability_zones</code> advanced config parameter is specified, virtual machines are deployed into two availability zones and a proximity placement group is not created.

6.4 Public IP Addresses

During Mediant VE stack creation, Stack Manager lets you specify which subnets (and corresponding network interfaces) will be assigned with public IP addresses via the **Public IPs** parameter in the **Networking** section.

For each subnet that is configured to use a Public IP address, the following is created:

■ **For standalone deployments:**

- Public IP address on the corresponding network interface
- Corresponding entry in the NAT Translation SBC configuration table, which implements SNAT translation for outbound traffic at the application level (SIP and SDP)

■ **For HA deployments:**

- Front-end rule with Public IP address on Azure Public Load Balancer
- Forwarding rules on Azure Public Load Balancer, which implement forwarding of incoming signaling traffic towards the active instance
- Outbound rules on Azure Public Load Balancer, which implement SNAT translation for outbound signaling traffic at the IP level
- Public IP addresses on the corresponding network interface of both instances, which are used for media traffic
- Corresponding entry in the NAT Translation SBC configuration table, which implements SNAT translation for outbound traffic at the application level (SIP and SDP)

It is possible to attach multiple public IP addresses to the same network interface. This may be done by specifying the **public_ips** advanced configuration parameter in the **Advanced Config** section during stack creation, or updating **Public IPs** parameter for existing stack via the **Modify** action.



Note: When the **public_ips** advanced configuration parameter is specified in the **Advanced Config** section during stack creation, it overrides any value configured via the **Public IPs** parameter in the **Networking** section.

■ **public_ips**

Contains comma-separated list of subnet names (main, additional1, additional2, etc.), which will be assigned with public IP addresses and optionally, the number of public IP addresses on the corresponding network interface.

For example:

```
public_ips = main:2,additional1
```

attaches two public IP addresses to the network interface connected to the “Main” subnet (eth0 for standalone deployment or HA deployment without HA subnet, eth1 for HA deployment with HA subnet) and one public IP address to the network interface connected to the “Additional 1” subnet (eth1 for standalone deployment or HA deployment without HA subnet, eth2 for HA deployment with HA subnet).

Stack Manager automatically creates secondary private IP addresses on the network interfaces that may be required for public IP attachment.

6.5 Private IP Addresses

For each subnet that is configured **not** to use a Public IP address, the following is created:

- **For standalone deployments:**
 - Private IP addresses on corresponding network interfaces are used
- **For HA deployments:**
 - Front-end rule on Azure Internal Load Balancer
 - Forwarding rule on Azure Internal Load Balancer, which implements forwarding of incoming signaling traffic towards the active Signaling Component instance
 - Private IP addresses on corresponding network interfaces, which are used for media traffic
 - Corresponding entry in the NAT Translation SBC configuration table, which implements SNAT translation for outbound traffic at application levels (SIP and SDP)

If you want to enable communication via both public and private IP addresses on the same subnet or add multiple private IP addresses to some network interface, specify **additional_ips** advanced configuration parameters in **Advanced Config** section during stack creation, or update **Additional IPs** parameter for existing stack via the **Modify** action.

- **additional_ips**

Contains a comma-separated list of subnet names (main, additional1, additional2, etc.), which will be assigned with additional private IP addresses and optionally, the number of additional private IP addresses on the corresponding network interface.

For example:

```
additional_ips = main,additional1:2
```

creates one additional private IP address on the network interface connected to the Main subnet (eth0 for standalone deployment or HA deployment without HA subnet, eth1 for HA deployment with HA subnet) and two additional private IP addresses on the network interface connected to the Additional 1 subnet (eth1 for standalone deployment or HA deployment without HA subnet, eth2 for HA deployment with HA subnet).

The number of additional private IP addresses is added *on top* of any private IP addresses created by Stack Manager by default and/or due to the public IP addresses assigned to the specific network interface. For HA configuration “pair” of IP addresses – for signaling and media traffic correspondingly – is created for each address specified via **additional_ips** parameter.

For example, the following configuration:

```
HA mode: enable
Main Subnet: <ha-subnet-id>
Main Subnet: <main-subnet-id>
Public IPs: Main subnet
Advanced Config:
    additional_ips = main
```

creates the following networking configuration:

- **eth0** – one primary IP address; used for communication between Mediant VE instances
- **eth1** – one primary and three secondary IP addresses:
 - primary IP address – placed behind Public Load Balancer (due to **Public IPs** configuration parameter) and used for signaling traffic

- 1st secondary IP address – assigned with “local” Public IP address on each instance (due to **Public IPs** configuration parameter) and used for media traffic
- 2nd secondary IP address – created due to **additional_ips** advanced config parameter containing “main” element; placed behind Internal Load Balancer and used for signaling traffic
- 3rd secondary IP address – created due to **additional_ips** advanced config parameter containing “main” element; “local” private IP address on each instance used for media traffic

6.6 Security Groups

6.6.1 Default Security Groups

For Mediant VE deployed via Stack Manager, the following security groups are automatically created:

- **Main** – security group for the “Main” subnet
- **VoIP** – security group for the “Additional 1”, “Additional 2”, etc. subnets
- **HA** – security group for the “HA” subnet (applicable to HA deployments with HA subnet only)

These default security groups are assigned to the following network interfaces:

Table 6-11: Assignment of Default Security Groups

Security Group	Interface Names for HA Deployment with HA Subnet	Interface Names for Standalone Deployment or HA Deployment without HA Subnet
HA	eth0	n/a
Main	eth1	eth0
VoIP	eth2, eth3, ...	eth1, eth2, ...

The following inbound rules are created in the default security groups:

Table 6-12: Inbound Rules for Default Security Groups

Security Group	Traffic	Protocol	Port	Source	Notes
Main	SSH	TCP	22	0.0.0.0/0	
	HTTP	TCP	80	0.0.0.0/0	
	HTTPS	TCP	443	0.0.0.0/0	
	SIP over UDP	UDP	5060	0.0.0.0/0	
	SIP over TCP	TCP	5060	0.0.0.0/0	
	SIP over TLS	TLS	5061	0.0.0.0/0	
	Media	UDP	6000-65535	0.0.0.0/0	
VoIP	SIP over UDP	UDP	5060	0.0.0.0/0	
	SIP over TCP	TCP	5060	0.0.0.0/0	
	SIP over TLS	TCP	5061	0.0.0.0/0	
	Media	UDP	6000-65535	0.0.0.0/0	

Security Group	Traffic	Protocol	Port	Source	Notes
HA	Internal	UDP	669	VirtualNetwork	If HA subnet is not used, these rules are added to Main subnet
	Internal	UDP	680	VirtualNetwork	
	Internal	TCP	80	VirtualNetwork	
	Internal	TCP	2442	VirtualNetwork	

Outbound rules are configured by default to allow all traffic.

6.6.2 Adjusting Default Security Groups

The default **Main** and **VoIP** security groups are configured by default to accept traffic from all sources, which constitutes a significant security risk. It is highly recommended to modify them after Mediant VE creation to allow inbound traffic only from specific IP addresses and/or subnets, especially for management traffic.

Note that inbound rules in the **HA** security group allow only traffic that originates from instances that reside in the same Virtual Network. Therefore, there is typically no need to modify them.

For Mediant VE deployed via Stack Manager, such modification can be done via the following stack configuration parameters:

■ Management Ports

Defines a list of inbound management ports and corresponding transport protocols as provided by the **Main** security group.

The value is a comma-separated list of the following elements:

```
<port>/<protocol>/ [<cidr>]
```

Where:

- <port> is an individual port number (e.g., 22) or a port range (e.g., 22-23)
- <protocol> is tcp or udp
- <cidr> is optional and can be an IP address (e.g., 10.1.2.3) or a CIDR (e.g., 10.1.0.0/16)

For example:

```
22/tcp/10.11.2.0/24,80/tcp/10.11.2.34,443/tcp
```

■ Signaling Ports

Defines a list of inbound signaling ports and corresponding transport protocols as provided by the **Main** and **VoIP** security groups.

■ Media Ports

Defines a list of inbound media ports and corresponding transport protocols as provided by the **Main** and **VoIP** security groups.

You can also update inbound and/or outbound rules of the default security groups via the Azure Portal or CLI interfaces. In this case, consider using the “keep-“ prefix for inbound rules that you create. This ensures that these rules are preserved during “update” and “rebuild” operations via Stack Manager.

6.6.3 Using Custom Security Groups

Instead of modifying rules of the default security groups created by Stack Manager, you can use custom security groups, for example, created by your IT department.

For Mediant VE deployed via Stack Manager, such configuration can be done via the following stack advanced configuration parameters:

- **ha_nsg_id**

Defines a custom security group to be used instead of the default **HA** security group.

Syntax: <ResourceGroupName>/<NsgName>

For example:

```
ha_nsg_id = rg1/ha-nsg
```

- **main_nsg_id**

Defines a custom security group to be used instead of the default **Main** security group.

- **voip_nsg_id**

Defines a custom security group to be used instead of the default **VoIP** security group.

Alternatively, you can assign custom network security groups to a specific interface via the following stack advanced configuration parameters:

- **nsg_id_ethX**

Defines a custom security group for a specific network interface.

For example:

```
nsg_id_eth0 = rg1/main-nsg
```

```
nsg_id_eth1 = rg1/voip-nsg
```

7 Changing Network Configuration After Deployment

For Mediant VE deployments performed via Stack Manager, use the **Modify** action to change the network configuration of the deployed stack. Stack Manager takes care of all needed operations regardless of the deployed Mediant VE software version.

For standalone Mediant VE deployments performed via Azure Portal or CLI, refer to the following sections on detailed instructions on how to change the network configuration after deployment.

7.1 Changing Network Configuration for Standalone Mediant VE Deployments Performed via Azure Portal or CLI



Note: This section is not applicable to Mediant VE deployments performed via Stack Manager. Use the **Modify** action in Stack Manager to perform all needed changes.

During initial deployment, Mediant VE automatically discovers all attached network interfaces and public IP addresses and populates corresponding network configuration tables accordingly.

From Version 7.4.260, Mediant VE software automatically detects changes to the virtual machine's network configuration (performed through Azure management interfaces) and adjusts corresponding network configuration tables (e.g., IP Interfaces and Ethernet Devices tables) accordingly. For Version 7.4.250 and earlier, you need to manually update the corresponding Mediant VE network configuration tables to match the updated Azure configuration.

The following chapters describe most common network configuration changes to the deployed Mediant VE instance and provide detailed instructions on how to perform them. We use Azure PowerShell CLI to perform the changes, however the same actions may be performed via the Azure portal as well.



Note: Mediant VE's "write factory" CLI command restores configuration to factory settings and triggers automatic network discovery upon the following reboot. It may be used as an alternative to online network configuration, as described below, in cases where you do not care about losing current Mediant VE configuration.

7.2 Adding Network Interface to Standalone Mediant VE Deployed via Azure Portal or CLI



Note: This section is not applicable to Mediant VE deployments performed via Stack Manager. Use the **Modify** action in Stack Manager to perform all needed changes.

➤ **To add network interface to deployed Mediant VE:**

1. Stop the virtual machine:

```
$VMName = "sbc-test1"
$ResourceGroupName = $VMName + "-rg"

Stop-AzVM -Name $VMName -ResourceGroupName $ResourceGroupName
```

2. Get parameters of pre-configured virtual network and subnet:

```
$VNetResourceGroupName = "SbcWestUS2"
$VNetName = "VnetWestUS2"
$SubnetName = "oam"

$VNet = Get-AzVirtualNetwork -Name $VNetName `
    -ResourceGroupName $VNetResourceGroupName

$Subnet = Get-AzVirtualNetworkSubnetConfig `
    -Name $SubnetName -VirtualNetwork $VNet
```

3. Create the new network interface and attach it to the virtual machine:

```
$InterfaceName = $VMName + "-eth1"

$Interface = New-AzNetworkInterface -Name $InterfaceName `
    -ResourceGroupName $ResourceGroupName `
    -Location $Location -SubnetId $Subnet.id

$VM = Get-AzVM -Name $VMName `
    -ResourceGroupName $ResourceGroupName

Add-AzVMNetworkInterface -VM $VM -Id $Interface.Id

Update-AzVM -ResourceGroupName $ResourceGroupName -VM $VM
```

4. Start the virtual machine:

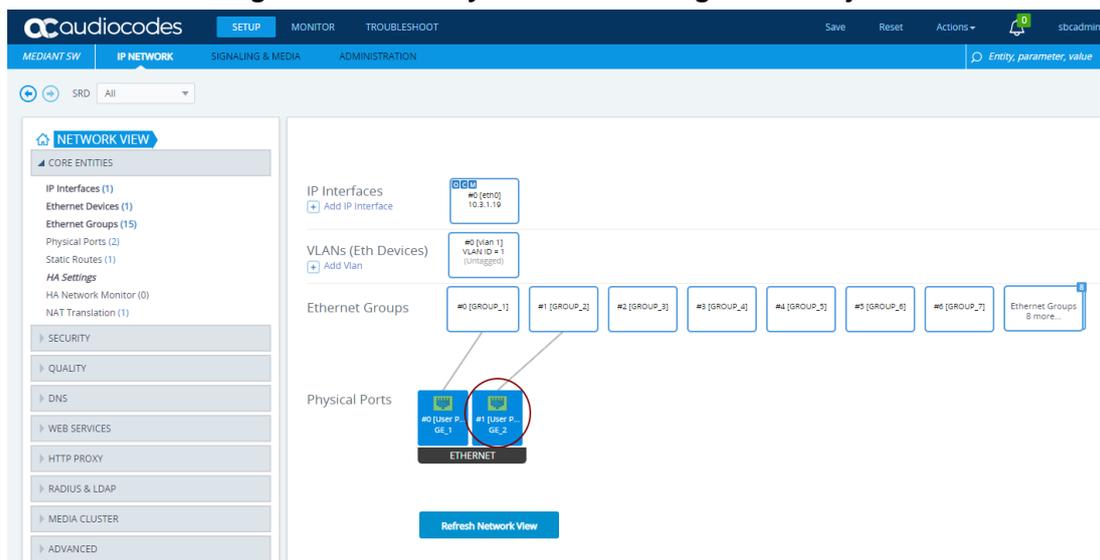
```
Start-AzVM -Name $VMName -ResourceGroupName $ResourceGroupName
```

5. Find the IP address of the created network interface

```
Write-Output $Interface.IpConfigurations.PrivateIpAddress
```

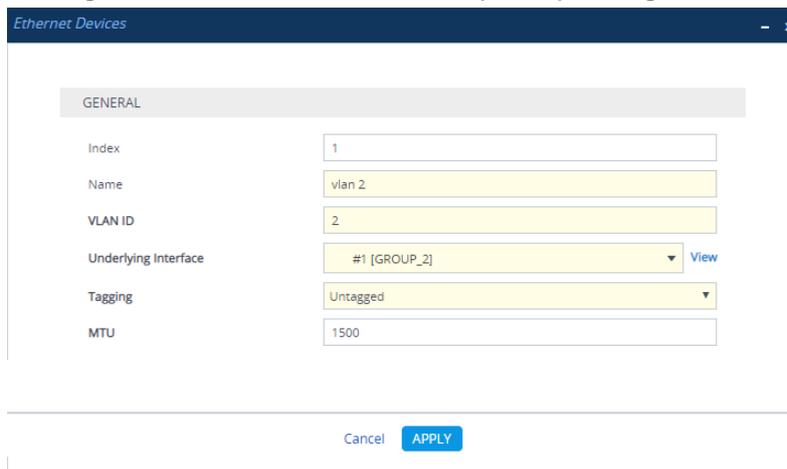
6. If you are using Version 7.4.260 or later, SBC software automatically detects configuration changes and updates the relevant configuration tables (e.g., IP Interfaces and Ethernet Devices tables) accordingly. Therefore, you can skip the following steps. If you are using Version 7.4.250 or earlier, continue with the following steps, which describe manual SBC software configuration.
7. Connect to the Mediant VE management interface through the Web.
8. Navigate to **SETUP > IP NETWORK**.
9. Note that Mediant VE detected a new network interface and created corresponding Physical Ports configuration object. The object is already attached to the corresponding Ethernet Group. However, Ethernet Device (VLAN) and IP Interface configuration is missing and must be manually created.

Figure 7-1: New Physical Ports Configuration Object



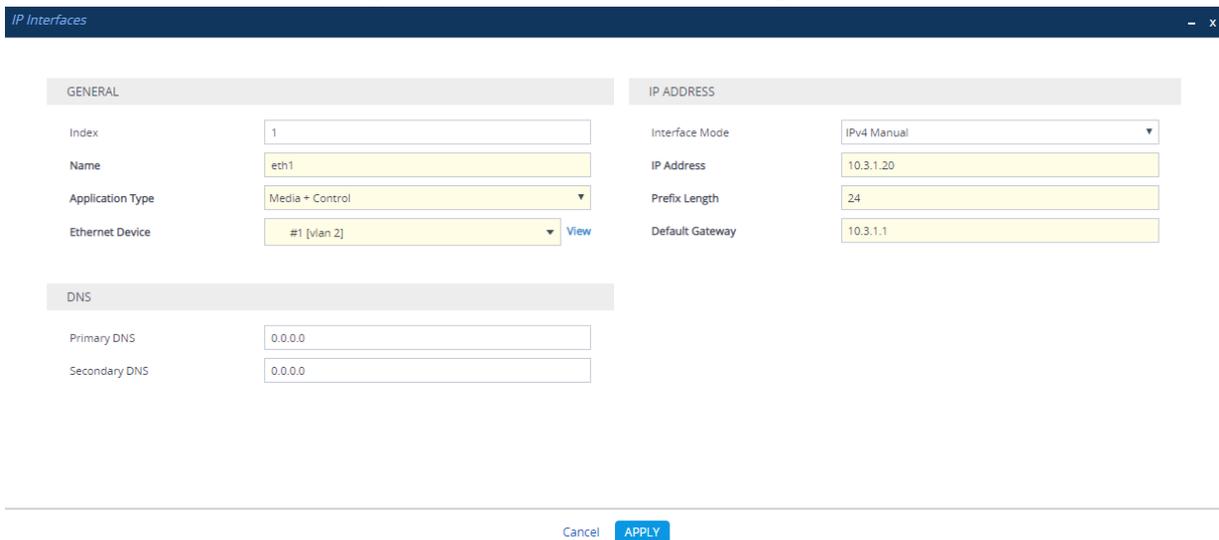
10. Click the **Add Vlan** link to create a new Ethernet Device (VLAN) configuration object, and configure it as follows:
 - Configure 'VLAN ID' as the next unused VLAN number.
 - Configure 'Tagging' as **Untagged**.
 - Configure 'Name' with some unique value (e.g., **vlan <VLAN ID>**).
 - Configure 'Underlying Interface' to reference the Ethernet Group associated with the new physical port.

Figure 7-2: New Ethernet Device (VLAN) Configuration



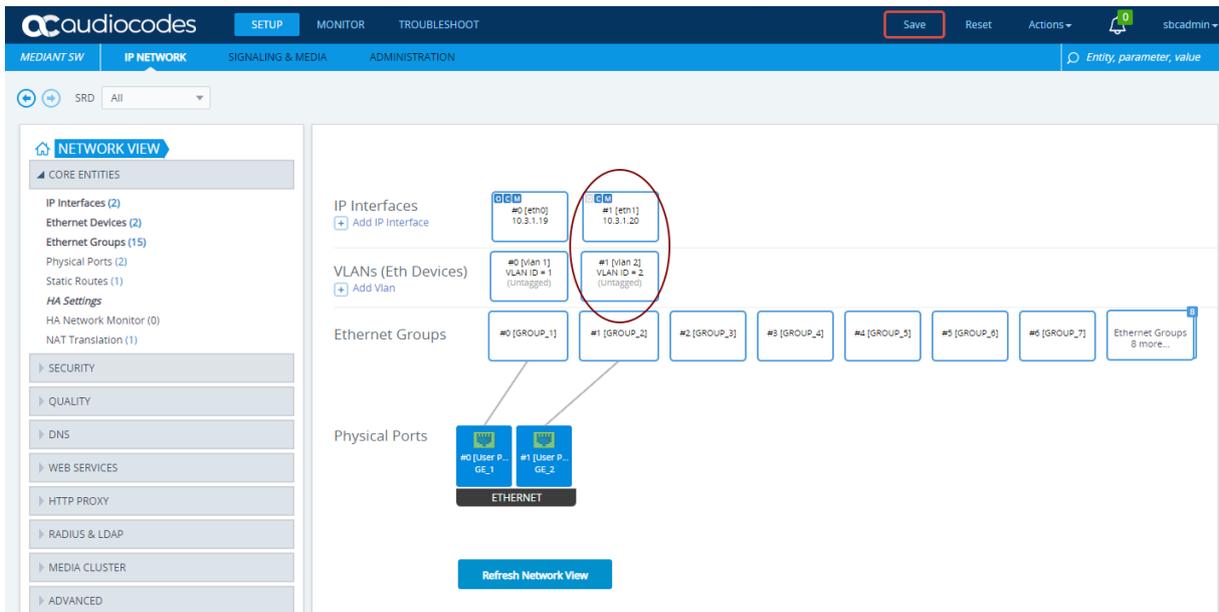
11. Click the **Add IP Interface** link to create a new IP Interface configuration object and configure it as follows:
 - Configure 'IP Address' with the IP address of the created network interface (as determined in step 5).
 - Configure 'Prefix Length' with the prefix length of the corresponding subnet.
 - Configure 'Default Gateway' with the corresponding default gateway.
 - Configure 'Name' with some unique value (e.g., **eth<id>**).
 - Configure 'Application Type' as **Media + Control**.
 - Configure 'Ethernet Device' to reference the Ethernet Device (VLAN) created in the previous step.

Figure 7-3: New IP Interface Configuration



12. Review the updated network configuration.

Figure 7-4: New Network Configuration



13. Click the **Save** button located on the toolbar to save the updated configuration.

7.3 Deleting the Network Interface from Standalone Mediant VE Deployed via Azure Portal or CLI



Note: This section is not applicable to Mediant VE deployments performed via Stack Manager. Use the **Modify** action in Stack Manager instead to perform all needed changes.

➤ **To delete network interface from the deployed Mediant VE:**

1. Stop the virtual machine:

```
$VMName = "sbc-test1"
```

```
$ResourceGroupName = $VMName + "-rg"
```

```
Stop-AzVM -Name $VMName -ResourceGroupName $ResourceGroupName
```

2. Detach the network interface from the virtual machine and delete it:

```
$InterfaceName = $VMName + "NetworkInterface2"
```

```
$Interface = Get-AzNetworkInterface -Name $InterfaceName `
             -ResourceGroupName $ResourceGroupName
```

```
$VM = Get-AzVM -Name $VMName `
      -ResourceGroupName $ResourceGroupName
```

```
Remove-AzVMNetworkInterface -VM $VM -Id $Interface.Id
```

```
Update-AzVM -ResourceGroupName $ResourceGroupName -VM $VM
```

```
Remove-AzNetworkInterface -Name $InterfaceName `
                          -ResourceGroupName $ResourceGroupName
```

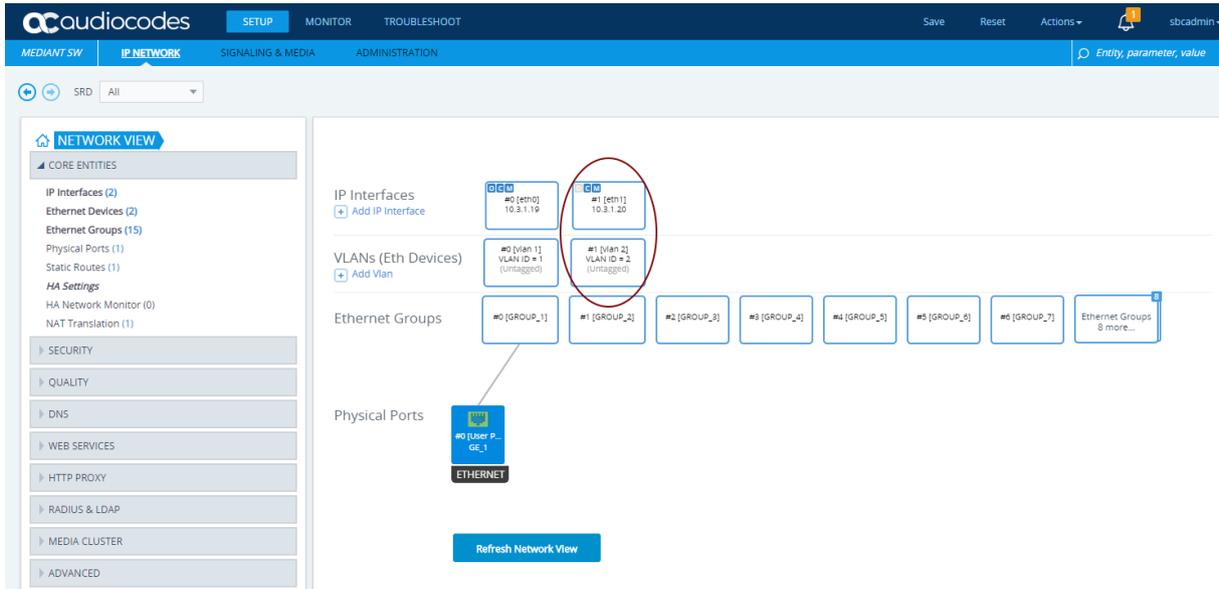
3. Start the virtual machine:

```
Start-AzVM -Name $VMName -ResourceGroupName $ResourceGroupName
```

4. If you are using Version 7.4.260 or later, SBC software automatically detects configuration changes and updates the relevant configuration tables (e.g., IP Interfaces and Ethernet Devices tables) accordingly. Therefore, you can skip the following steps. If you are using Version 7.4.250 or earlier, continue with the following steps, which describe manual SBC software configuration.
5. Connect to the Mediant VE management interface through the Web interface.

6. Navigate to **SETUP > IP NETWORK**.
7. Locate the remaining network configuration objects that correspond to the deleted network interface.

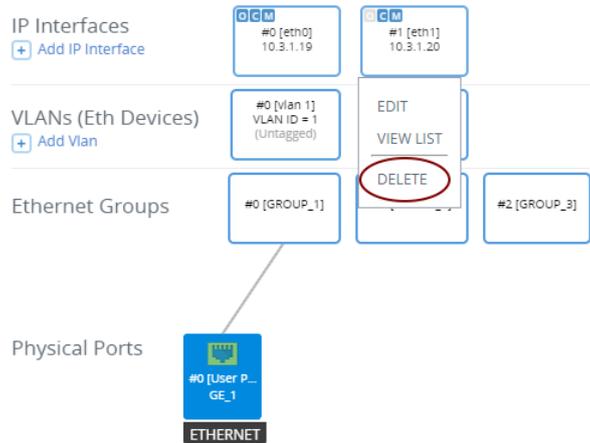
Figure 7-5: Remaining Network Configuration Objects



In the above example, the remaining network configuration objects include:

- IP Interface #1 [eth1]
 - VLAN #1 [vlan 2]
8. Delete the remaining configuration objects -- first the IP interface and then the VLAN -- by clicking them and then from the shortcut menu, choosing **Delete**.

Figure 7-6: Deleing Remaining IP Interface



9. Click the **Save** button located on the toolbar to save the updated configuration.

Figure 7-7: Saving the updated configuration



8 Upgrading the Software Version



Note: This section is applicable only to Mediant VE deployed as a **virtual appliance**. For deployments using **container images**, see Section 8.1, Upgrading the Software Version for Container Deployments.

You may upgrade the software version of the deployed Mediant VE software using the firmware (.cmp) file.

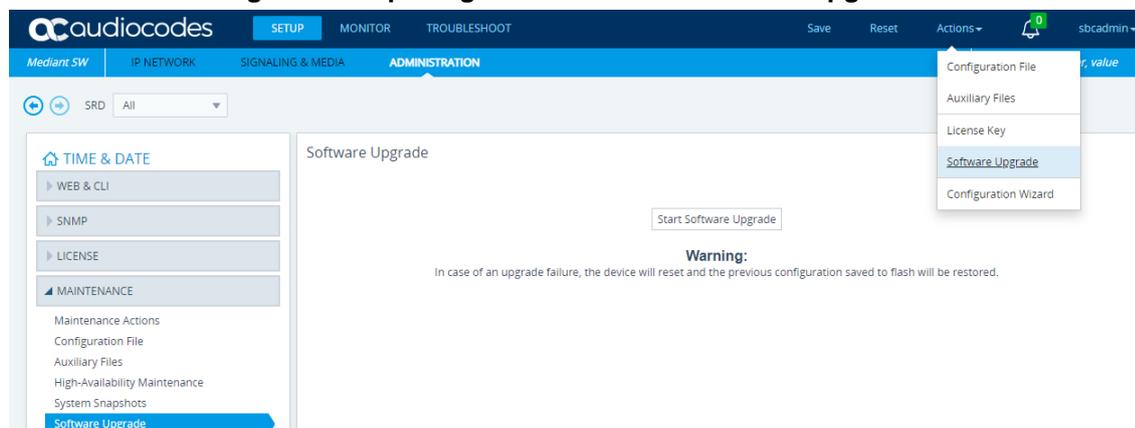
Upgrade to 7.6 stream (7.60A.xxx.yyy versions) can be performed only from 7.4 stream (7.40A.xxx.yyy versions).



Note: If you have an earlier version installed (e.g., from 7.2 stream), first upgrade to the 7.4 stream, and only then upgrade to the 7.6 stream. Refer to 7.4 version documentation for detailed upgrade instructions.

To upgrade Mediant VE via Web interface, click **Action > Software Upgrade** on the toolbar to open the Software Upgrade wizard.

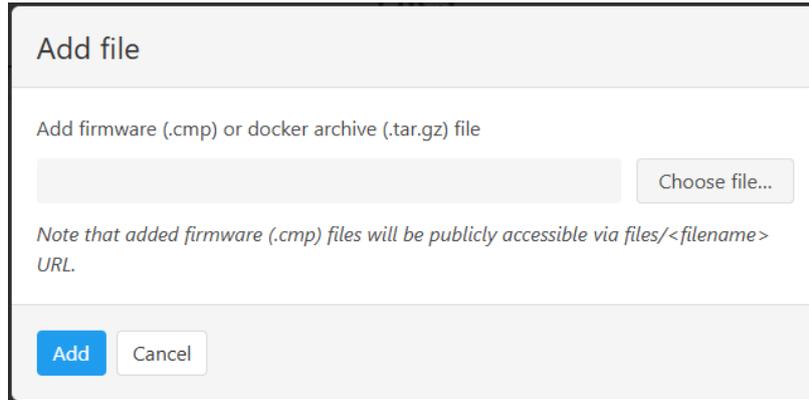
Figure 8-1: Opening Web Interface's Software Upgrade Wizard



Alternatively, if Mediant VE was installed via Stack Manager, you can upgrade it using the **Upgrade** action:

1. Log into the Stack Manager's Web interface.
2. On the menu bar, click **Files**; the Files screen appears.
3. Click **Add file**; the Add file dialog box appears.

Figure 8-2: Stack Manager: Add File Dialog



4. Choose the Mediant VE firmware (.cmp) file, and then click **Add**; the file is uploaded to Stack Manager and appears in the **Files** list.

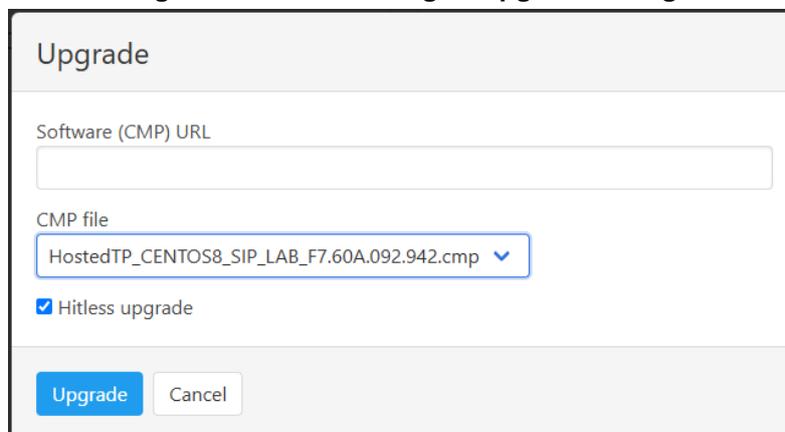
Figure 8-3: Stack Manager: CMP File Upload



Name	Type	Added on	Added by	Status	Used by	Action
HostedTP_CENTOS8_SIP_LAB_F7.60A.092.942.cmp	cmp	Mar 31, 2025 14:04:05	Admin	ok		Delete

5. On the menu bar, click **Stacks**; the Stacks screen appears.
6. Select your stack; the stack details screen appears.
7. In the action bar, click **Upgrade**; the upgrade dialog is displayed.

Figure 8-4: Stack Manager: Upgrade Dialog



8. Select the CMP file that you uploaded in Step 4. Alternatively, you can enter the URL of a CMP file that is hosted on a public HTTP/HTTPS server.

9. Select the **Hitless** upgrade check box if you want to perform upgrade without service interruption (HA deployments only). Clear the check box if you want a faster upgrade.
10. Click **Upgrade** to start the upgrade.

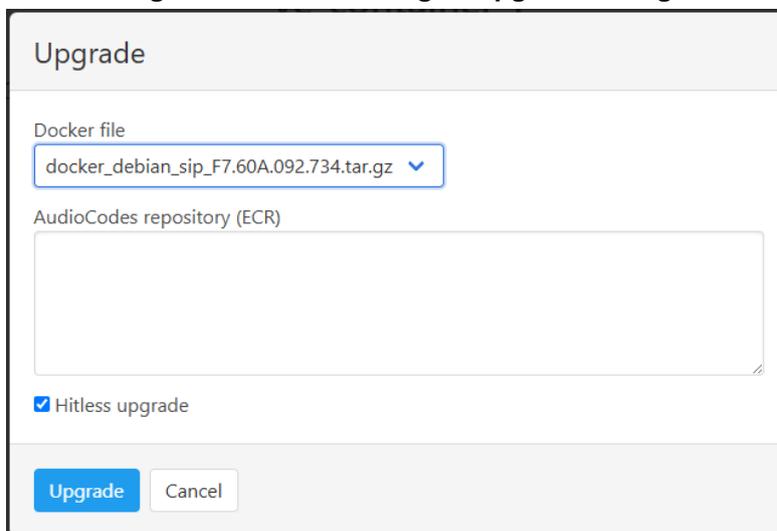
8.1 Upgrading the Software Version for Container Deployments

Mediant VE deployments that use **container images** don't support software upgrade via the firmware (.cmp) file. Instead Mediant VE Container installation package – **docker_debian_sip_F7.60A.xxx.yyy.tar.gz** – is used.

For simplicity, the Stack Manager's Web interface is described. However, the same task may be completed through the CLI and REST management interfaces. Refer to *Stack Manager User Manual* for details.

1. Log into the Stack Manager's Web interface.
2. Upload new Mediant VE Container image in Files screen.
3. In the Stacks screen, select your stack.
4. Click **Upgrade**; the upgrade dialog is displayed.

Figure 8-5: Stack Manager: Upgrade Dialog



The screenshot shows a web-based dialog box titled "Upgrade". It features a "Docker file" dropdown menu with the selected option "docker_debian_sip_F7.60A.092.734.tar.gz". Below this is a text input field for "AudioCodes repository (ECR)", which is currently empty. A checkbox labeled "Hitless upgrade" is checked. At the bottom of the dialog, there are two buttons: "Upgrade" (highlighted in blue) and "Cancel".

5. Select the new Mediant VE Container image (Docker file).
6. Leave the 'AudioCodes repository (ECR)' text box empty.
7. Select the **Hitless upgrade** check box if you want to perform the upgrade without service interruption (HA deployments only). Clear the check box if you want a faster upgrade.
8. Click **Upgrade** to start the upgrade.

This page is intentionally left blank.

9 Licensing the Product

Once you have successfully completed Mediant VE deployment, you need to obtain, activate and then install your purchased SBC license.



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*).

9.1 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 <http://www.audiocodes.com/swactivation>:

Figure 9-1: Software License Activation Tool

License Activation

Please enter your Product Key received from AudioCodes and the fingerprint (e.g. Serial Number or Server Signature) that was generated as a result of your installation.
For technical assistance, please contact AudioCodes support at support@audiocodes.com.

Product Key*

Fingerprint*

Email* +

Validation 3ECF8

Please enter the characters shown in the image. To refresh the image, [click here](#).

Send

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 e-mail sent to you by AudioCodes upon your purchase, as shown in the example below:

Figure 9-2: Product Key in Order Confirmation E-mail



- **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 e-mail 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.

9.2 Installing the License Key



Note: The License Key installation process includes a device reset and is therefore, traffic-affecting. To minimize the disruption of current calls, it is recommended to perform this procedure during periods of low traffic.

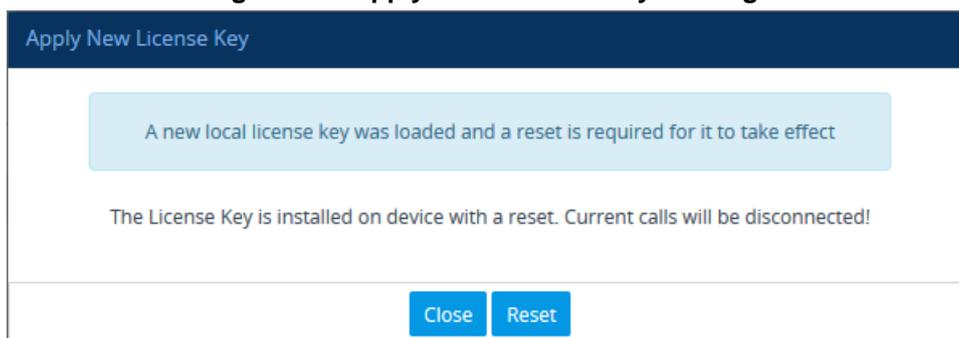
- **To install a License Key file for standalone devices through Web interface:**
1. Open the License Key page (**Setup** menu > **Administration** tab > **License** folder > **License Key**).
 2. Back up the currently installed License Key, as a precaution. If the new License Key does not comply with your requirements, you can re-load this backed-up License Key to restore the device's original capabilities. To back up the License Key, click  icon and save it as file on your PC.
 3. Click the **Load File** button, navigate to the License Key file on your computer, and then select the file to load to the device; the **Apply New License Key** button appears. The License Key page uses color-coded icons to indicate the changes between the previous License Key and the newly loaded License Key.



Note: If want to cancel installation, reset the device without a save to flash. For more information, see Resetting the Device.

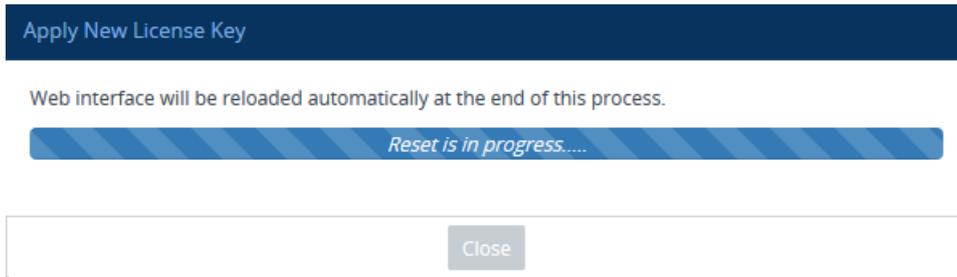
4. Click **Apply New License Key**; the following message box appears:

Figure 9-3: Apply New License Key Message



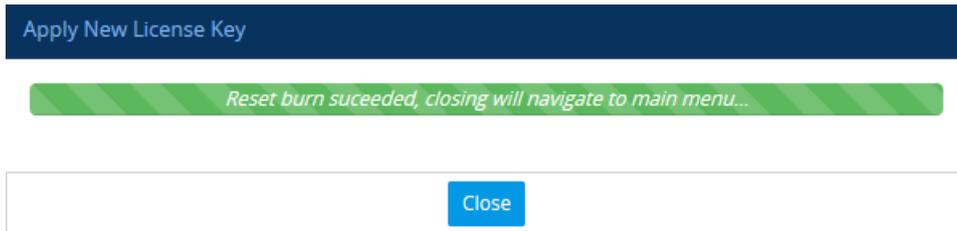
- Click **Reset**; the device begins to save the file to flash memory with a reset and the following progress message box appears:

Figure 9-4: Reset in Progress for License Key



When installation completes, the following message box appears:

Figure 9-5: Reset and Save-to-Flash Success Message



- Click **Close** to close the message box; you are logged out of the Web interface and prompted to log in again. The features and capabilities displayed on the License Key page now reflect the newly installed License Key.

9.3 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 9-6: Viewing Product Key

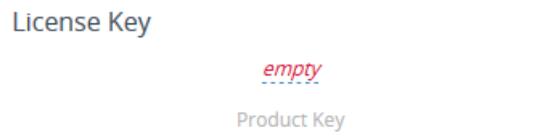


- Device Information page (**Monitor** menu > **Monitor** tab > **Summary** folder > **Device Information**).

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:

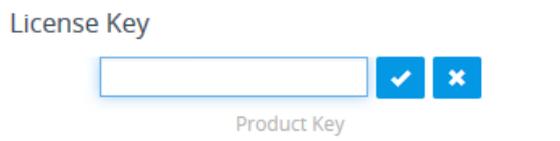
1. Open the License Key page.
2. Locate the Product Key group:

Figure 9-7: Empty Product Key Field



3. Click "empty"; the following appears:

Figure 6-9-8: Entering Product Key



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

International Headquarters

Naimi Park
6 Ofra Haza Street
Or Yehuda, 6032303, 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>

©2025 AudioCodes Ltd. All rights reserved. AudioCodes, AC, HD VoIP, HD VoIP Sounds Better, IPmedia, Mediant, MediaPack, What's Inside Matters, OSN, SmartTAP, User Management Pack, VMAS, VoIPerfect, VoIPerfectHD, Your Gateway To VoIP, 3GX, AudioCodes One Voice, AudioCodes Meeting Insights, and AudioCodes Room Experience are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners. Product specifications are subject to change without notice.

Document #: LTRT-11018

