Installation Manual

AudioCodes Family of Session Border Controllers (SBC)

Mediant Virtual Edition (VE) SBC

Deployment in Container Environments

Version 7.6

audiocodes

Table of Contents

Notice......iv

Security Vulnerabilities......iv

WEEE EU Directive......iv

Customer Support......iv

Stay in the Loop with AudioCodes......iv

Abbreviations and Terminology......iv

Related Documentation......iv

Document Revision Record......iv

Documentation Feedback......v

1 Introduction......2

1.1 Mediant VE Container......2

1.2 Product Package......3

1.3 Deployment Methods......3

2 Deploying Mediant VE Container using Installation Script......4

2.1 Prerequisites......4

2.2 Deploying Mediant VE Container......4

2.3 Mediant VE Container Data......5

2.4 Mediant VE Container Network Mode......6

2.4.1 Symmetric Routing and Per-Interface Route Tables......6

2.4.2 Network Features Parity......7

3 Operating Mediant VE Container......8

3.1 Connecting to Host Machine through SSH......8

3.2 Verifying Mediant VE Container is Up and Running......8

3.3 Starting and Stopping Mediant VE Container......8

3.4 Connecting to Mediant VE Container’s Web Interface......9

3.5 Connecting to Mediant VE Container’s CLI through SSH......9

3.6 Connecting to Mediant VE Container’s CLI through Console......10

3.7 Restarting Mediant VE Container......10

4 Deploying Mediant VE Container using Stack Manager......11

4.1 Installing Stack Manager......11

4.2 Deploying a Mediant VE Container......12

4.3 Resources Created by Stack Manager......18

4.4 Network Ports and Security Groups......19

4.4.1 Adjusting Network Security Groups......20

4.5 Connecting to Host Machine through SSH......20

4.6 Login Credentials......21

4.6.1 Logging in to Host Machine using SSH Key 22
4.6.2 Checking Connectivity 23
4.6.3 Mediant VE Container IP Addresses 24
5 Upgrading Mediant VE Container 26
5.1 Upgrade Procedure if Deployed using Installation Script 26
5.2 Upgrade Procedure if Deployed using Stack Manager 26

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: April-20-2025

Security Vulnerabilities

All security vulnerabilities should be reported to vulnerability@audiocodes.com.

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

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

Document Name
Stack Manager for Mediant VE-CE SBC User's Manual

Document Revision Record

LTRT	Description
11032	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

AudioCodes' Mediant Virtual Edition (VE) Session Border Controller (SBC) is a software product that enables connectivity and security between enterprises' and Service Providers' VoIP networks.

Mediant VE SBC can be deployed as the following:

- Virtual appliance
- Container

This document describes the deployment of Mediant VE SBC as a **Container**.

Throughout this document, the term *Mediant VE Container* is used to reflect the Mediant VE SBC container deployment.

For detailed instructions on Mediant VE installation as a virtual appliance, refer to the *Mediant Virtual Edition SBC for VMware-KVM-HyperV Installation Manual*.

> - 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 the *Recommended Security Guidelines*.
> - For configuring Mediant VE SBC, refer to the *Mediant Software SBC User's Manual*.

1.1 Mediant VE Container

Mediant VE Container runs inside a container on a supported Linux virtual host.

The virtual host must be deployed in one of the following virtual environments:

- **VMware:** VMware ESXi Version 6.5 or later
- **KVM:** Linux version 4.18 or later, with KVM/QEMU
- **Hyper-V:** Microsoft Server 2019 or later

> Version 7.6.100 officially supports the deployment of Mediant VE Container **only** in Microsoft Azure public cloud. If you're interested in deploying Mediant VE Container in a different environment, Contact your AudioCodes representative.

The virtual host must meet the following requirements:

- 2 vCPUs or more
  - 64-bit Intel® vCPUs
  - With Advanced Vector Extensions (AVX) and AES-NI support
- 8 GB of RAM or more
- 16 GB of free disk space or more
- Two vNICs are recommended (for trusted and untrusted traffic)

> If you're using multiple vNICs, make sure that they are connected to different subnets. For more information, see Section 2.4, Mediant VE Container Network Mode.

The virtual host must have one of the following 64-bit versions of **Linux OS** installed:

- Debian 11 or 12
- Ubuntu 22.04 or 24.04
- RHEL 9
- Rocky 9
- Alma Linux 9
- Amazon Linux 2023

Docker engine and **docker-compose** tool are used to run Mediant VE Container.

Customers are provided with access to the Linux host and may modify or maintain it according to their needs, using standard or custom tools.

Additional software packages, for example, Backup tool or Security scanner may be installed on the host machine alongside the Mediant VE Container. However, these tools should not generate high CPU, network or storage consumption, nor interfere with normal SBC container operations. For all means and purposes, Mediant VE Container should be the main workload running on the host.

1.2 Product Package

Mediant VE Container is delivered as a **docker_debian_sip_F7.60A.xxx.yyy.tar.gz** archive that can be downloaded from AudioCodes Services Portal site (registered Customers only), under **Mediant Virtual Edition (VE), Mediant Cloud Edition (CE) and Mediant Server Edition (SE) Session Border Controllers (SBC)**.

The archive contains the following files:

- Container image – **sbc_docker.tar.gz**
- Installation script – **install-docker.sh**
- Additional files required for software installation

1.3 Deployment Methods

Mediant VE Container can be deployed using one of the following methods:

- Installation script – see Section 2, Deploying Mediant VE Container using Installation Script
- Stack Manager – see Section 4, Deploying Mediant VE Container using Stack Manager

2 Deploying Mediant VE Container using Installation Script

This section describes the deployment of a **standalone** Mediant VE Container using the installation script method. For using Stack Manager to deploy Mediant VE Container, which supports **both standalone and High-Availability** (HA) topologies, see Section 4, Deploying Mediant VE Container using Stack Manager.

2.1 Prerequisites

The prerequisites for deploying Mediant VE Container using the installation script method include the following:

- The host machine must meet the hardware and software requirements, as specified in Section 1.1, Mediant VE Container.
- The host machine must have an administrative user account that is allowed to run the **sudo** command without additional authentication. For example, type **sudo ls** and verify that the output is displayed without the need to re-enter the password.
- The host machine must have access to the internet during Mediant VE Container installation. This is required as the installation downloads several packages (e.g., **docker** and **docker-compose**) from official software repositories. After the installation completes, internet access is no longer required and can be disabled.

Version 7.6.100 supports the deployment of Mediant VE Container in Microsoft Azure public cloud **only**.

2.2 Deploying Mediant VE Container

The following procedure describes how to deploy Mediant VE Container using the installation script.

To deploy Mediant VE Container using installation script:

1. Connect to the host machine through SSH or the serial console.
2. Log in as a user who has administrative privileges to run **sudo** command without additional authentication.
3. Use an SCP or SFTP client (e.g., WinSCP) to transfer the Mediant VE Container installation package (**docker_debian_sip_F7.60A.xxx.yyy.tar.gz**) to the host.
4. Extract the installation package (replace *.xxx.yyy* in the example below with the actual package name):

```
tar xvfz docker_debian_sip_F7.60A.xxx.yyy.tar.gz
```

5. Run the installation script:

   ```
   bash install-docker.sh
   ```

   The installation script does the following:

   - Downloads and installs the needed packages (e.g., **docker** and **docker-compose**)
   - Creates the **/opt/sbc** directory where most of the Mediant VE Container artifacts are stored (see below).
   - Creates the **sbc_startup** service, which configures the Linux host for optimal Mediant VE Container operation.
   - Creates the **load-image.sh** script, which is used for loading the Mediant VE Container image.

6. Run the **load-image.sh** script to load and run the Mediant VE Container image from the installation package. Specify the same package name as in Step 4:

   ```
   ./load-image.sh docker_debian_sip_F7.60A.xxx.yyy.tar.gz
   ```

   The loading of the image script does the following:

   - Extracts the Mediant VE Container image file from the installation package and loads it to the local **docker** image repository.
   - Creates the **docker-compose.yml** file in the **/opt/sbc** directory.
   - Starts Mediant VE Container using **docker-compose**.

2.3 Mediant VE Container Data

All Mediant VE Container data is stored in the **/opt/sbc/data** directory on a Linux host machine. If you want to back up Mediant VE Container configuration, you can make a backup of this directory.

2.4 Mediant VE Container Network Mode

Mediant VE Container runs in “host” network mode. This means that it shares networking namespace with the host machine and has access to all the host’s network interfaces.

Contrary to Mediant VE SBC virtual offering, Mediant VE Container doesn’t configure network interfaces, acquires addresses for them, nor alters them in any other way. It’s the host machine’s responsibility to acquire IP addresses on all network interfaces and maintain them, aligned with “real” virtual host’s configuration.

2.4.1 Symmetric Routing and Per-Interface Route Tables

During Mediant VE Container installation, dedicated route tables are created for each network interface on the host machine, with different default gateways in each route table. Traffic that egresses from the subnet that corresponds to the specific network interface is configured to use the corresponding route table.

This configuration ensures that responses for network packets received from a specific network interface are routed back through the same interface. This behavior is typically referred to as “symmetric routing”. It also facilitates the use of an interface-specific default gateway, thereby allowing Mediant VE Container to properly communicate with external equipment through any available IP address of the host machine.

Route tables created during Mediant VE Container installation are numbered 100 for the first vNIC, 101 for the second vNIC, 102 for the third vNIC , and so on.

You can use the following commands to view route tables:

- To view a specific route table (replace 100 in the example below with the route table that you want to view):

```
ip route show table 100
```

Example:

```
# ip route show table 100
default via 10.1.10.1 dev eth0
10.1.10.0/24 dev eth0 scope link

# ip route show table 101
default via 10.1.11.1 dev eth1
10.1.11.0/24 dev eth1 scope link
```

- To view all route tables:

```
ip route show table all
```

- To view the lookup table that determines which route table is used for a specific packet:

```
ip rule show
```

Example:

```
# ip rule show
0:      from all lookup local
32764:  from 10.1.11.0/24 lookup 101
32765:  from 10.1.10.0/24 lookup 100
32766:  from all lookup main
32767:  from all lookup default
```

Per-interface routing rules match traffic based on the source subnet CIDR. Therefore, they work correctly **only** if each network interface (vNIC) is connected to a different subnet. If you try to deploy Mediant VE Container on a virtual host that has two or more vNICs connected to the same subnet, per-route route tables won't work correctly. As a result, Mediant VE Container will sometimes respond to packets that are received from one vNIC via a different vNIC. In many cases, such "asymmetric routing" breaks application / equipment that communicates with the SBC and causes service degradation.

> ⓘ If your host machine has multiple vNICs, make sure that they're connected to different subnets. Failure to do so prevents Mediant VE Container from operating properly and causes service degradation.

2.4.2 Network Features Parity

Mediant VE Container automatically populates its **IP Interfaces** table with the network interface names and IP addresses available on the host machine.

The table is read-only and contains only a subset of the information available on the Mediant VE virtual appliance. The reason is because container deployment isn't responsible for network interface configuration, as described in the previous section, and only "binds" to the IP addresses available on the host machine.

As the IP Interfaces table is read-only, the 'Application Type' field has no meaning and is maintained for compatibility (legacy) reasons only. Dedicated row pointers are used to "bind" specific applications to the corresponding IP Interfaces table entries, for example:

```
[ OAMDefaultInterfaceIPv4 ]
FORMAT Index = InterfaceName;
OAMDefaultInterfaceIPv4 0 = eth0;
[ \\OAMDefaultInterfaceIPv4 ]
```

Mediant VE Container doesn't provide the following configuration tables and features:

- Ethernet Devices table
- Ethernet Groups
- Firewall / Access List
- Static Routes

The corresponding functionality can be implemented on the host machine using native Linux tools, such as **iptables**, **netplan** or **Network Manager**.

3 Operating Mediant VE Container

3.1 Connecting to Host Machine through SSH

Connecting to the host machine using an SSH client on the standard port 22 logs you into the Linux shell of the host machine.

For instructions on how to connect to Mediant VE Container's CLI interface, see the following sections:

- Section 3.5, Connecting to Mediant VE Container's CLI through SSH
- Section 3.6, Connecting to Mediant VE Container's CLI through Console

3.2 Verifying Mediant VE Container is Up and Running

To verify Mediant VE Container is up and running, use the following command:

```
docker ps
```

If Mediant VE Container is running, its container ID, status, uptime, and other information are displayed. For example:

```
CONTAINER ID   IMAGE                             COMMAND
CREATED           STATUS          PORTS      NAMES
04414b414982   docker_debian_sip:7.60a.092.936   "/start-
docker.sh"   16 minutes ago   Up 5 minutes
docker_debian_sip_7.60a.092.936
```

3.3 Starting and Stopping Mediant VE Container

To stop Mediant VE Container, use the following command:

```
cd /opt/sbc
docker-compose down
```

To start Mediant VE Container, use the following command:

```
cd /opt/sbc
docker-compose up -d
```

3.4 Connecting to Mediant VE Container’s Web Interface

When Mediant VE Container is running, you may connect to its Web interface by entering the primary IP address of the first host machine’s network interface.

To connect to Web interface:

1. Determine the primary IP address of the first host machine’s network interface, by using the following command:

   ```
   grep InterfaceTable /opt/sbc/data/ac/ini/user.ini
   ```

   The command displays the InterfaceTable (as determined by SBC software). For example:

   ```
   [ InterfaceTable ]
   InterfaceTable 0 = 6, 10, 10.1.10.161, 24, eth0;
   InterfaceTable 1 = 5, 10, 10.1.10.166, 24, eth0:1;
   InterfaceTable 2 = 5, 10, 10.1.10.167, 24, eth0:2;
   [ \InterfaceTable ]
   ```

2. Use the first address listed in the output above to access the Web interface through your web browser; the Login Screen appears.
3. Log in using the following default credentials:
   - Username: **Admin**
   - Password: **Admin**

3.5 Connecting to Mediant VE Container’s CLI through SSH

When Mediant VE Container is running, you can connect to its Command-Line Interface (CLI) using an SSH client. Use port 1122 and the primary IP address of the first Linux host’s network interface.

To connect to CLI:

1. Determine the IP address of the first Linux host’s network interface (see Section 3.4, Connecting to Mediant VE Container’s Web Interface).
2. Use the following command (replace 10.1.10.161 in example below with actual IP address):

   ```
   ssh 10.1.10.161 -p 1122
   ```

3. Log in using the same credentials that you used for the Web interface.

3.6 Connecting to Mediant VE Container’s CLI through Console

The following procedure describes how to connect to Mediant VE Container’s CLI through the console.

To connect to CLI through console:

1. Determine the container ID, by using the following command:

   ```
   docker ps
   ```

2. Attach to the container ID:

   ```
   docker attach <container-id>
   ```

3. Log in using the same credentials that you used for the Web interface.

To disconnect from console:

- Detach from the Mediant VE Container’s console, by pressing the following key sequence on your keyboard:

  CTRL-P / CTRL-Q

3.7 Restarting Mediant VE Container

If you use **Restart** button in Mediant VE Container’s Web interface or **reload now** command in Mediant VE Container’s CLI, only the container is restarted.

If you want to restart the host machine on which the container is running, log into the virtual host’s Linux shell (via SSH or serial console), and then use the **reboot** command.

4 Deploying Mediant VE Container using Stack Manager

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

> ⓘ Version 7.6.100 supports the deployment of Mediant VE Container **only** in Microsoft Azure public cloud.

4.1 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*.

> ⓘ You need Stack Manager Version 3.8.9 or later to deploy Mediant VE Container.

You also need to enable **docker deployment mode** as follows:

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

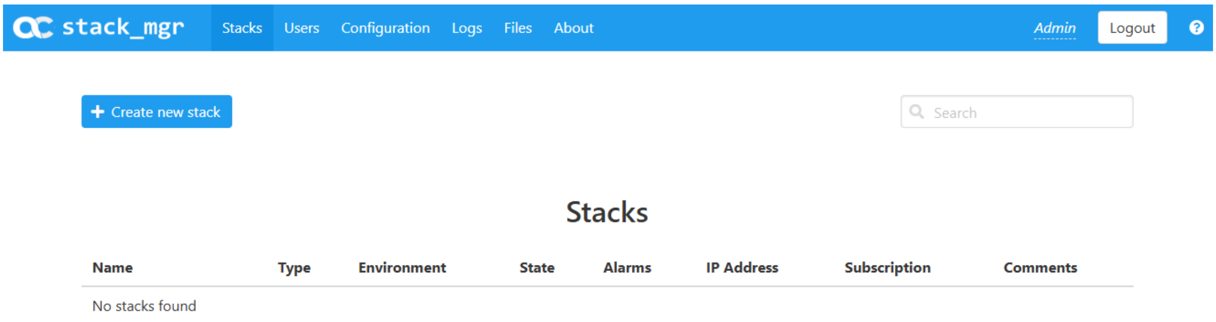
4.2 Deploying a Mediant VE Container

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

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

To deploy a Mediant VE Container:

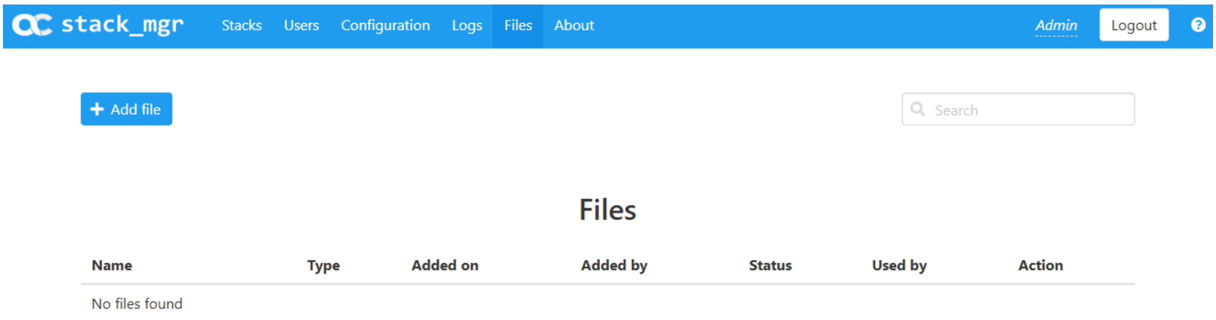
1. Log in to Stack Manager.



Stacks

Name	Type	Environment	State	Alarms	IP Address	Subscription	Comments
No stacks found							

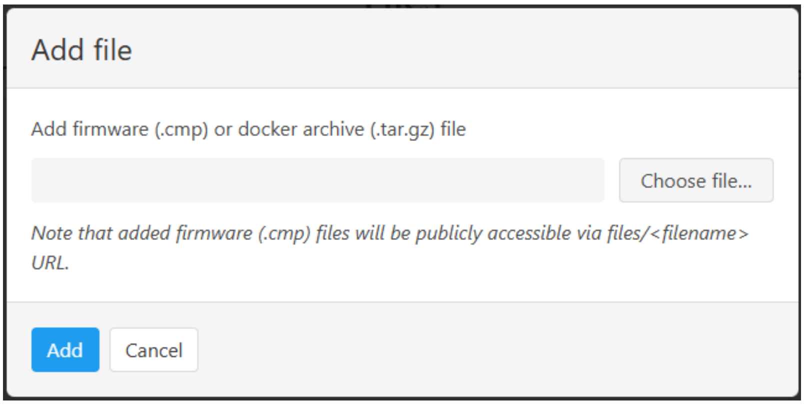
2. On the menu bar, click **Files**; the Files screen appears.



Files

Name	Type	Added on	Added by	Status	Used by	Action
No files found						

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



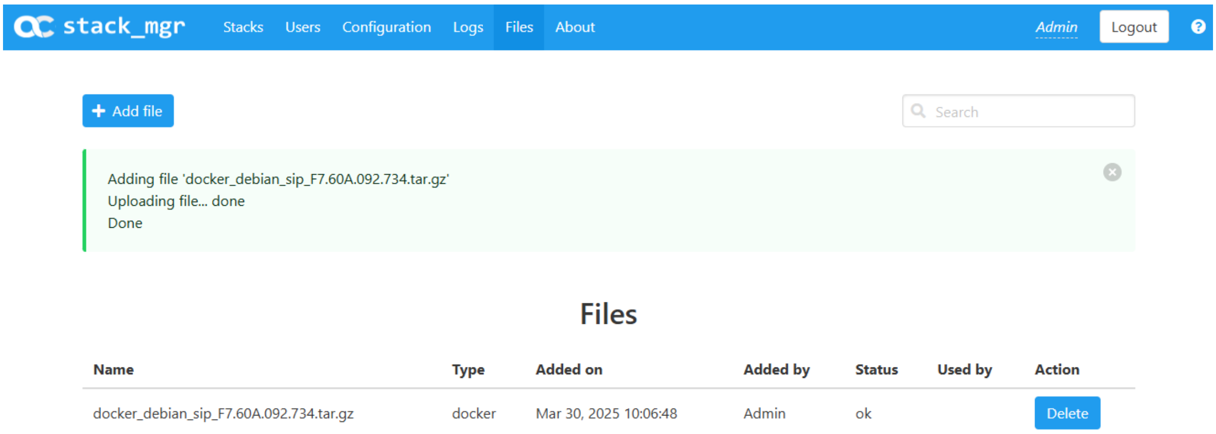
Add file

Add firmware (.cmp) or docker archive (.tar.gz) file

Note that added firmware (.cmp) files will be publicly accessible via files/<filename> URL.

4. Click **Choose file**, and then select the Mediant VE Container image file.

5. Click **Add**; the file is uploaded to Stack Manager and appears in the **Files** list:



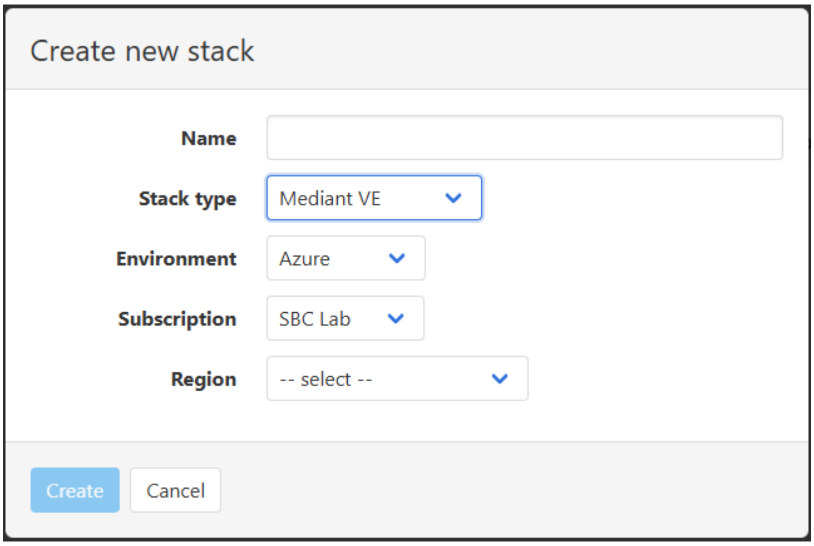
Adding file 'docker_debian_sip_F7.60A.092.734.tar.gz'
Uploading file... done
Done

Files

Name	Type	Added on	Added by	Status	Used by	Action
docker_debian_sip_F7.60A.092.734.tar.gz	docker	Mar 30, 2025 10:06:48	Admin	ok		Delete

6. On the menu bar, click **Stacks**; the Stacks screen appears.

7. Click **Create new stack**; the Create new stack dialog box appears:



Create new stack

Name:

Stack type: Mediant VE

Environment: Azure

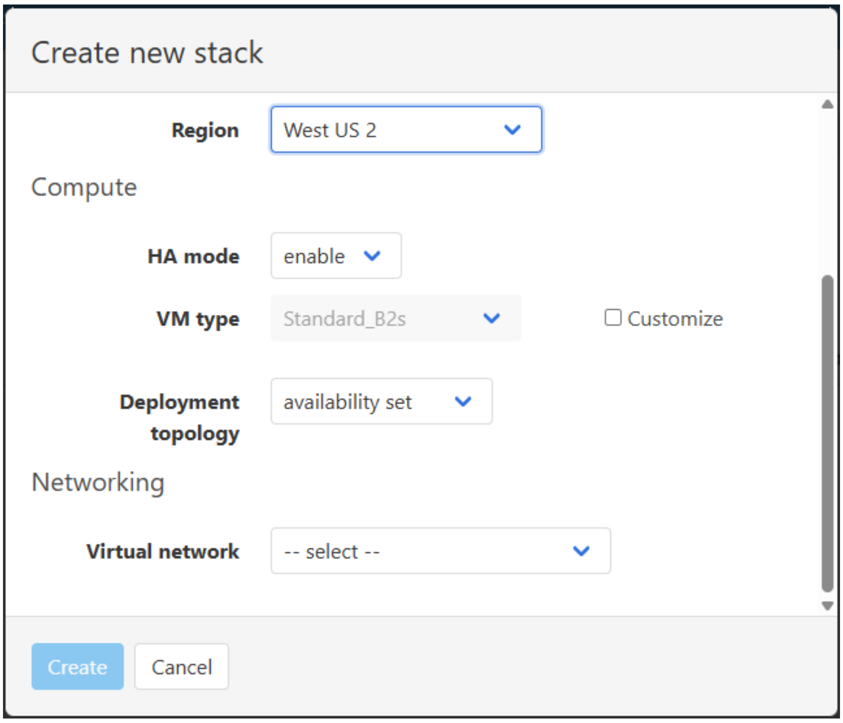
Subscription: SBC Lab

Region: -- select --

8. 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.).

9. From the 'Stack type' drop-down list, select **Mediant VE**.

10. From the ‘Region’ drop-down list, select the region where Mediant VE Container is to be deployed.



Create new stack

Region: West US 2

Compute

HA mode: enable

VM type: Standard_B2s

☐ Customize

Deployment topology: availability set

Networking

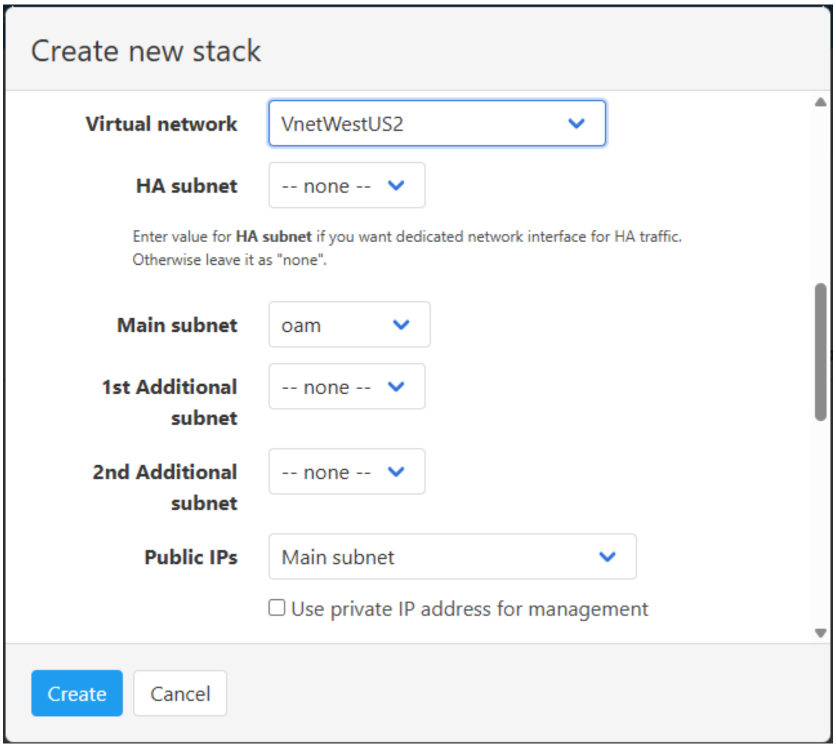
Virtual network: -- select --

11. From the ‘HA mode’ drop-down list, select **enable** for HA topology, or **disable** for standalone topology.

12. For ‘VM type’, Stack Manager automatically determines the appropriate virtual machine type based on the number of configured network interfaces. If you want to use a different VM type, select the ‘Customize’ check box, and then choose the custom VM type.

13. From the ‘Deployment topology’ drop-down list, select whether you want to deploy Mediant VE Container instances into **availability set** or across **availability zones**. If you choose **availability zones**, you’re 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.

14. From the ‘Virtual network’ drop-down list, select the virtual network where Mediant VE Container is to be deployed.



Create new stack

Virtual network: VnetWestUS2

HA subnet: -- none --

Enter value for **HA subnet** if you want dedicated network interface for HA traffic. Otherwise leave it as "none".

Main subnet: oam

1st Additional subnet: -- none --

2nd Additional subnet: -- none --

Public IPs: Main subnet

□ Use private IP address for management

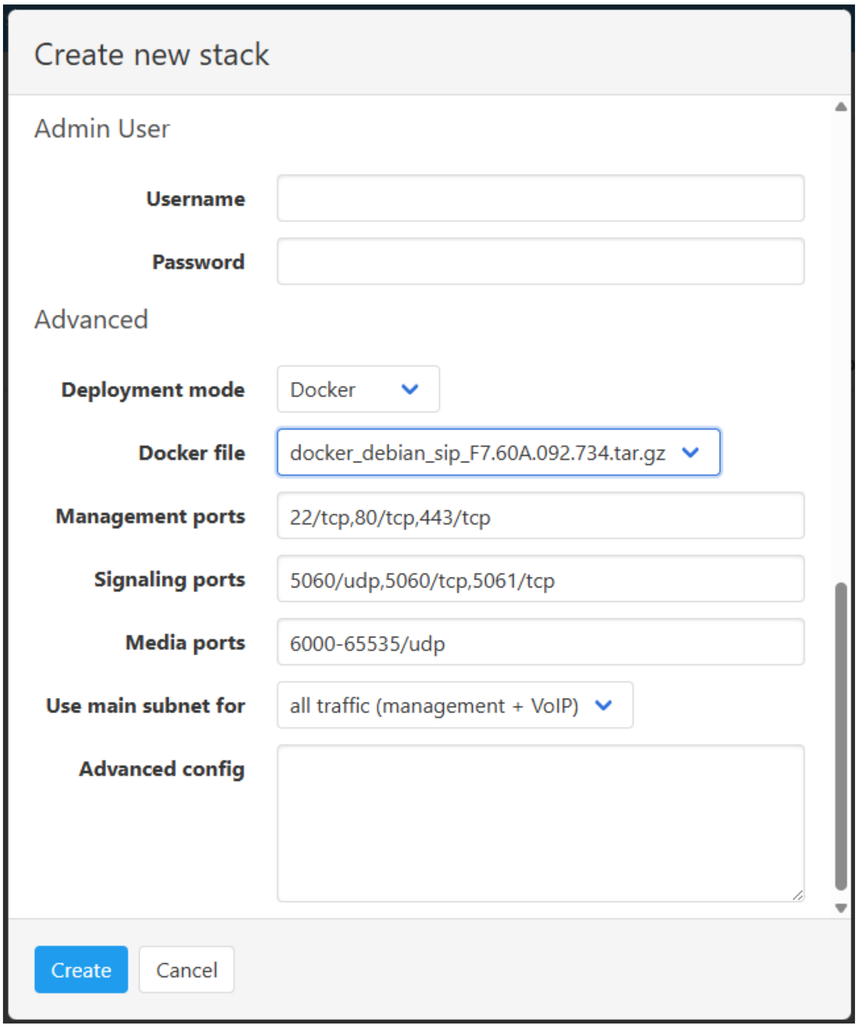
15. From the ‘HA subnet’ drop-down list, select the **HA** subnet for internal communication between the two SBC instances. The subnet is applicable to HA topology only and is connected through the second network interface (eth1). It is possible to skip this subnet for HA topology, by selecting **-- none--**. In this case, the “main” subnet is used for internal communication between the two SBC instances.

16. From the ‘Main subnet’ drop-down list, 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).

17. From the ‘1st Additional subnets’ and ‘2nd Additional subnets’ drop-down lists, select “additional” subnets for signaling (SIP) and media (RTP/RTCP) traffic. The subnets are connected to the virtual machine(s) through additional network interfaces: eth1/eth2 for standalone topology or for HA topology without an HA subnet, eth2/eth3 for HA topology with an HA subnet. If you don’t need these additional network interfaces, leave them at **-- none --**.

18. From the ‘Public IPs’ drop-down list, select the subnets (and corresponding network interfaces) that must be assigned with public IP addresses.

19. 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 selecting the ‘Use private IP address for management’ check box. In this case, Stack Manager creates additional (secondary) private IP address on “main” interface and uses it to communicate with the deployed stack.



Create new stack

Admin User

Username:

Password:

Advanced

Deployment mode: Docker

Docker file: docker_debian_sip_F7.60A.092.734.tar.gz

Management ports: 22/tcp,80/tcp,443/tcp

Signaling ports: 5060/udp,5060/tcp,5061/tcp

Media ports: 6000-65535/udp

Use main subnet for: all traffic (management + VoIP)

Advanced config:

20. Under the **Admin User** group, enter the default credentials for Mediant VE Container management interfaces (Web, SSH, and serial console). The same credentials are configured for the host machine and for Mediant SBC software.

> 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.

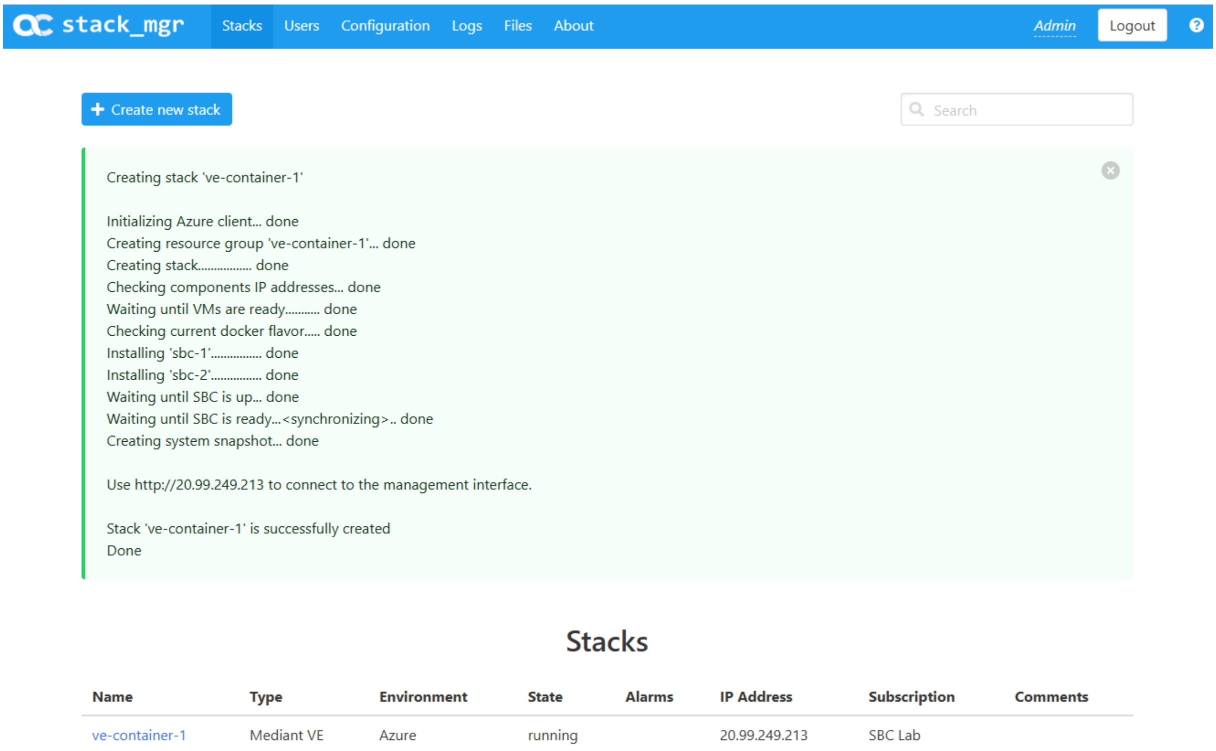
21. From the ‘Deployment mode’ drop-down list, select **Docker**.

22. From the ‘Docker file’ drop-down list, select the Mediant VE Container image file that you added in the **Files** screen.

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

24. Click **Create** to start stack creation.

25. Wait until the stack creation process completes:



```
Creating stack 've-container-1'

Initializing Azure client... done
Creating resource group 've-container-1'... done
Creating stack................ done
Checking components IP addresses... done
Waiting until VMs are ready.......... done
Checking current docker flavor..... done
Installing 'sbc-1'............... done
Installing 'sbc-2'............... done
Waiting until SBC is up... done
Waiting until SBC is ready...<synchronizing>.. done
Creating system snapshot... done

Use http://20.99.249.213 to connect to the management interface.

Stack 've-container-1' is successfully created
Done
```

Stacks

Name	Type	Environment	State	Alarms	IP Address	Subscription	Comments
ve-container-1	Mediant VE	Azure	running		20.99.249.213	SBC Lab	

4.3 Resources Created by Stack Manager

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

Resource	Description
Resource Group	Resource group into which all stack resources are deployed. This may be overridden by the `resource_group` advanced config parameter.
Virtual Machines	Virtual machines that run the SBC software.
Network Security Groups	Network security groups for different traffic types. This may be overridden by the `ha_nsg_id`, `main_nsg_id`, `voip_nsg_id` and `nsg_id_ethX` advanced config parameters.
Storage Account	Storage account for storing VM diagnostics data. This may be overridden by the `diag_account` 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. Stack Manager creates Load Balancer and all nested configuration – back-end pools, probes and forwarding rules. For Mediant VE Container deployments, NAT rules are also created to enable access to each host machine (VM) via the 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 that are assigned to virtual machines and load balancers. This may be overridden by the `public_ip_*` or `public_ip_prefix` 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 `use_availability_set` advanced config parameter. If the `availability_zones` 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 `use_proximity_placement_group` advanced config parameter. If the `availability_zones` advanced config parameter is specified, virtual machines are deployed into two availability zones and a proximity placement group is not created.

4.4 Network Ports and Security Groups

Stack Manager uses **Management ports**, **Signaling ports** and **Media ports** parameters, specified in **Create stack** and **Modify** dialogs, to configure Network Security Groups and Load Balancing Rules.

The following table lists the default configuration for these parameters and explains the purpose of each included port:

Group	Ports	Description
Management ports	22/tcp	Provides access to the SSH interface of the host machine. For HA configuration, use stack's "IP address" (as shown in stack details screen, under **General** section) to access this port. The address is assigned to the Load Balancer and the traffic is "routed" to the currently active VM.
	80/tcp 443/tcp	Provides access to the Web interface. For HA configuration, use stack's "IP address" (as shown in stack details screen, under **General** section) to access these ports. The address is assigned to the Load Balancer and the traffic is "routed" to the currently active VM.
	1122/tcp	Provides access to the CLI of the Mediant VE Container. For HA configuration, use stack's "IP address" (as shown in stack details screen, under **General** section) to access this port. The address is assigned to the Load Balancer and the traffic is "routed" to the currently active VM.
	1022/tcp	For HA configuration, which use public IP addresses on the "main" subnet (for management traffic) two NAT rules are created on the Load Balancer to provide direct access to the SSH interface on host machines (VMs). Each rule has its own "frontend" IP address and "listens" on port 22. Received traffic is routed to the backend instance via port 1022. Therefore, this port must be defined in the Network Security Group attached to the VM. Use the **More > Show IP Addresses** command in Stack Manager to find IP addresses of these NAT rules.
Signaling Ports	5060/udp 5060/tcp 5061/tcp	Provides access to the SIP interface of Mediant VE Container. For HA configuration, use IP addresses assigned to the Load Balancer to access these ports. The traffic is "passed" to the currently active VM. Use the **More > Show IP Addresses** command in Stack Manager to find IP addresses assigned to the Load Balancer.
Media Ports	6000-65535/udp	Media streams handled by Mediant VE Container. Use dedicated IP addresses that do **NOT** reside behind the Load Balancer (e.g., eth0:1) to access these ports.

4.4.1 Adjusting Network Security Groups

All ports mentioned in the previous section (4.4, Network Ports and Security Groups) are configured by default to accept traffic from any source. This is very useful for lab environments, but doesn't meet security requirements of production environments.

Therefore, for production deployments, it's recommended to limit some of the ports – especially Management ports – to accept traffic only from specified source addresses / subnets.

To limit ports, add an IP address or CIDR after a specific port/protocol element. Multiple addresses or CIDRs can be specified using a semicolon (;). For example:

```
Management ports: 22/tcp/10.1.1.12;10.1.1.13,443/tcp/10.1.1.0/24
```

Refer to the *Stack Manager User Manual* for more information.

> If you specify IP addresses / subnets for management ports 443/tcp, 22/tcp, and 1022/tcp make sure to always include Stack Manager's IP address in the provided list. This is needed because Stack Manager uses these ports to communicate with the deployed Mediant VE Container stacks and perform maintenance operations (e.g. "upgrade").
>
> Use Stack Manager's internal IP address if Mediant VE Container's management interface uses internal IP addresses. Use Stack Manager's public IP address if Mediant VE Container's management interface uses public IP addresses.

4.5 Connecting to Host Machine through SSH

For HA deployments, use one of the following methods to connect to the host machines:

1. In the Stack Manager **Stacks** screen, click your stack; the Stack details screen appears.
2. Under the **General** group**,** locate the 'IP Address' parameter. This is the stack's management IP address. It's assigned to Azure Load Balancer and at any given time is "routed" to the active Mediant VE Container instance. Use this address to access Linux shell of the host machine of the currently active Mediant VE Container. Connect to it using an SSH client on the standard port 22.
3. Use the **More > Show IP Addresses** command in Stack Manager to view all the stack's IP addresses. Locate the addresses that can be used to access Linux shell of the specific host machine, regardless of its role (active / standby) and state of the Mediant VE Container software:
   - For configurations that use public IP addresses on the "main" interface, two NAT rules on the public Load Balancer are displayed – "nat-sbc-1" and "nat-sbc-2". These rules pass the traffic from standard SSH port 22 to the specific host machine using port 1022.
   - For other configurations, use "eth0" of the corresponding VM – "sbc-1" or "sbc-2".
4. Connect to the specific IP address using an SSH client on the standard port 22.
5. If you want to connect to the Mediant VE Container CLI, configure your SSH client to connect to port 1122. Alternatively, you can access Mediant VE Container's CLI from the host machine's Linux shell, as described in Section 3.6, Connecting to Mediant VE Container's CLI through Console.

4.6 Login Credentials

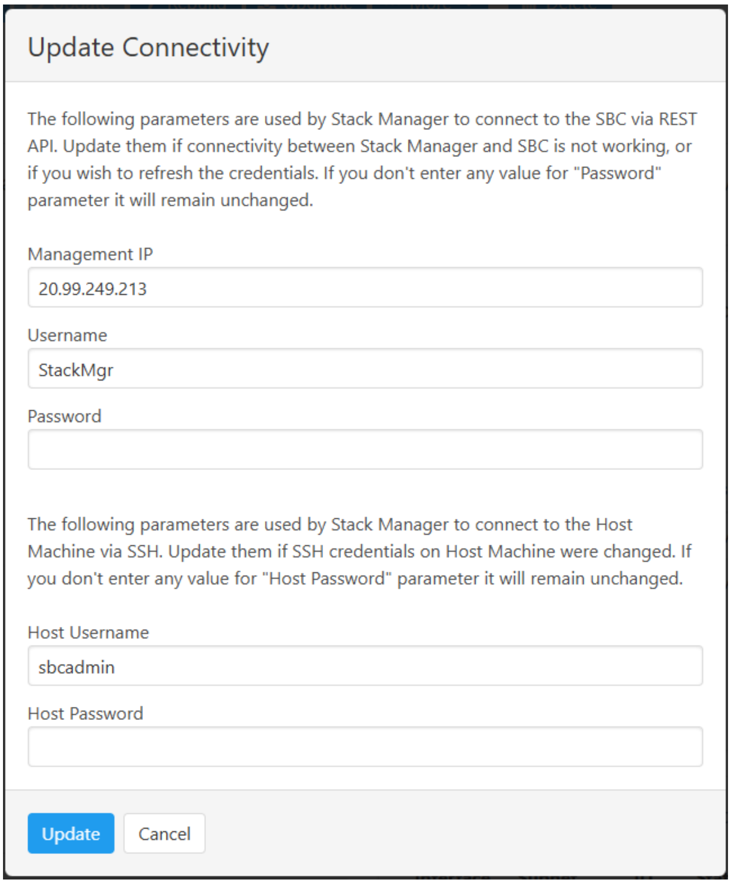
During Mediant VE Container creation, Stack Manager configures the same credentials – specified in **Create stack** dialog – for both host machine and Mediant VE Container software. This means that after stack creation, you can use the same username and password to log in to the host machine's SSH interface and to the SBC's Web and CLI interfaces.

Stack Manager creates a user named "StackMgr" with a random password on Mediant VE Container and uses it to access the SBC's REST API. However, it uses the regular username and password to access the host machine's SSH interface.

If the host machine is reconfigured and its login credentials are changed, you need to update Stack Manager with the new credentials so that it will be able to log in to the host machine, for example, during the "upgrade" operation. Do this as follows:

To update host machine credentials on Stack Manager:

1. Log in to Stack Manager's Web interface.
2. In the **Stacks** screen, click your stack.
3. Click **More > Update Connectivity**; the update connectivity screen appears.



Update Connectivity

The following parameters are used by Stack Manager to connect to the SBC via REST API. Update them if connectivity between Stack Manager and SBC is not working, or if you wish to refresh the credentials. If you don't enter any value for "Password" parameter it will remain unchanged.

Management IP: 20.99.249.213

Username: StackMgr

Password:

The following parameters are used by Stack Manager to connect to the Host Machine via SSH. Update them if SSH credentials on Host Machine were changed. If you don't enter any value for "Host Password" parameter it will remain unchanged.

Host Username: sbcadmin

Host Password:

4. In the 'Host Username' and 'Host Password' fields, enter the new values.
5. Click **Update** to apply the new configuration.

4.6.1 Logging in to Host Machine using SSH Key

If the host machine is reconfigured to allow login using the SSH key only, update Stack Manager's configuration accordingly, as described in the following procedure.

To configure Stack Manager to use SSH key when connecting to host machine:

1. Connect to the Stack Manager's virtual machine through SSH.
2. Upload the SSH private key file in PEM format.
3. Change the file and directory permissions using the **chown** and **chmod** commands so that the uploaded file is readable by the **stack_mgr** user.
4. Access the Stack Manager's CLI:

   ```
   stack_mgr_cli
   ```

5. Verify that the uploaded file is readable by the **stack_mgr** user. Replace example below to use the actual file location. If file content is displayed, the file is readable by the **stack_mgr** user.

   ```
   cat /home/admin/ssh-key.pem
   ```

6. Configure the location of the uploaded SSH key file. Replace the example below to use the actual stack name and file location.

   ```
   stack_mgr modify container-ve-1 \
     --docker-host-ssh-key-file /home/admin/ssh-key.pem
   ```

7. You can also use CLI to change the host machine's username:

   ```
   stack_mgr modify container-ve-1 \
     --docker-host-username <username>
   ```

8. If you disable SSH key authentication and switch back to password authentication, use the following command:

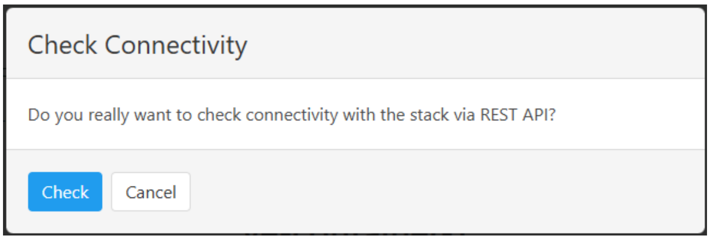
   ```
   stack_mgr modify container-ve-1 --docker-host-ssh-key-file ""
   ```

9. You can then change the password:

   ```
   stack_mgr modify container-ve-1 --docker-host-password
   <password>
   ```

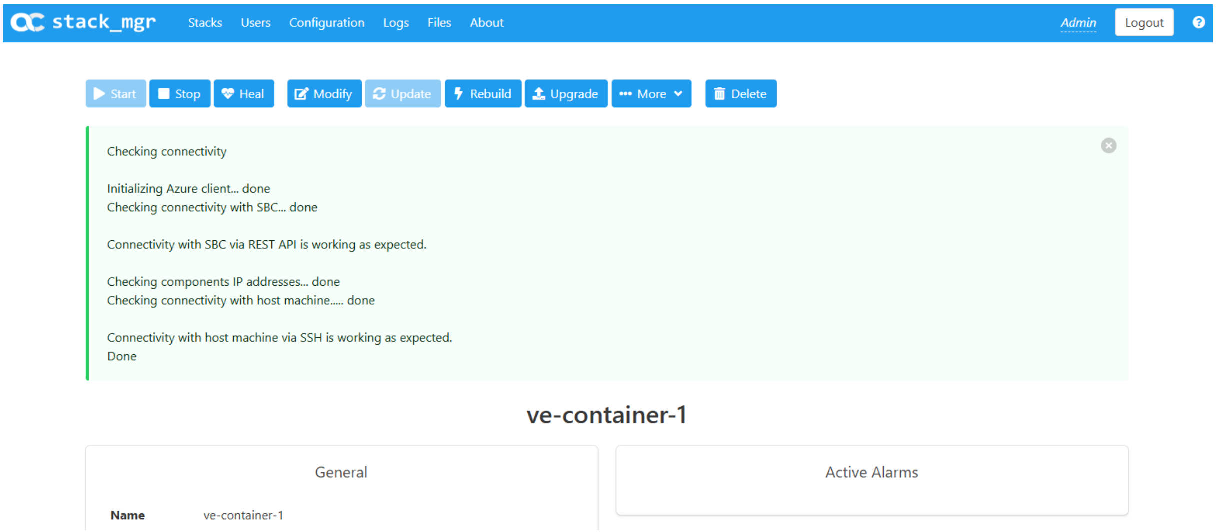
4.6.2 Checking Connectivity

Use the **More > Check Connectivity** operation in Stack Manager's Web interface to check connectivity between Stack Manager and Mediant VE Container. Both REST API connection with the SBC application and the SSH connection with host machines (via NAT rules) are checked.


Check Connectivity

Do you really want to check connectivity with the stack via REST API?

If errors are detected, recommended mitigation instructions are provided in the output.



```
Checking connectivity

Initializing Azure client... done
Checking connectivity with SBC... done

Connectivity with SBC via REST API is working as expected.

Checking components IP addresses... done
Checking connectivity with host machine..... done

Connectivity with host machine via SSH is working as expected.
Done
```

ve-container-1

General

Active Alarms

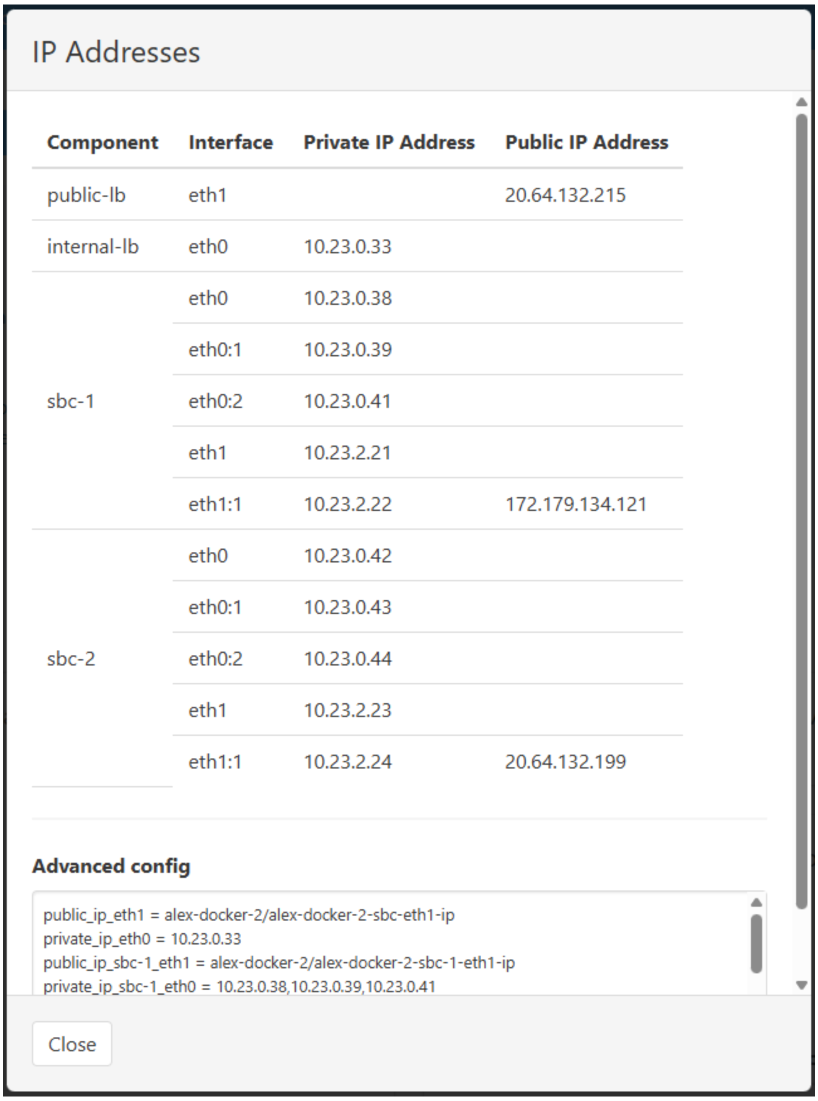
Name	ve-container-1

4.6.3 Mediant VE Container IP Addresses

HA deployment of Mediant VE Container creates “pairs” of IP addresses on each network interface. The first IP address in the “pair” is placed behind Azure Load Balancer and is used for management and signaling traffic. The second IP address in the “pair” is not placed behind Azure Load Balancer and is used for media traffic. During a switchover, existing calls and sessions are “relatched” to the new IP addresses using SIP re-INVITE messages to ensure service continuity.

Use the **More** > **Show IP Addresses** command in Stack Manager to display all stack’s IP addresses.

For example, consider the following output:



IP Addresses

Component	Interface	Private IP Address	Public IP Address
public-lb	eth1		20.64.132.215
internal-lb	eth0	10.23.0.33	
sbc-1	eth0	10.23.0.38	
	eth0:1	10.23.0.39	
	eth0:2	10.23.0.41	
	eth1	10.23.2.21	
	eth1:1	10.23.2.22	172.179.134.121
sbc-2	eth0	10.23.0.42	
	eth0:1	10.23.0.43	
	eth0:2	10.23.0.44	
	eth1	10.23.2.23	
	eth1:1	10.23.2.24	20.64.132.199

Advanced config

```
public_ip_eth1 = alex-docker-2/alex-docker-2-sbc-eth1-ip
private_ip_eth0 = 10.23.0.33
public_ip_sbc-1_eth1 = alex-docker-2/alex-docker-2-sbc-1-eth1-ip
private_ip_sbc-1_eth0 = 10.23.0.38,10.23.0.39,10.23.0.41
```

In this example:

- Each host machine (**sbc-1** and **sbc-2**) has two network interfaces (vNICs).
- The first vNIC (**eth0**) uses internal IP addresses, while the second vNIC (**eth1**) uses public IP addresses.
- The first vNIC of each host machine (VM) has three IP addresses:
  - **eth0** – this address is placed behind the internal Azure Load Balancer and is used for management and signaling traffic.
  - **eth0:1** – this address doesn’t reside behind Azure Load Balancer and is used for media traffic.

  - **eth0:2** – this address is used for internal communication between two Mediant VE Container instances.

- The second vNIC of each host machine (VM) has two IP addresses:
  - **eth1** – this address is placed behind the public Azure Load Balancer and is used for signaling traffic.
  - **eth1:1** – this address doesn't reside behind Azure Load Balancer and is used for media traffic. Each host machine has a dedicated public IP address assigned to this address.
- Two Azure Load Balancers are created:
  - **internal-lb** – internal Azure Load Balancer, which has the **eth0** internal IP address that is "routed" to the corresponding address on the first vNIC of the currently active Mediant VE Container instance.
  - **public-lb** – public Azure Load Balancer, which has the **eth1** public IP address that is "routed" to the corresponding address on the second vNIC of the currently active Mediant VE Container instance.
- Mediant VE Container should be configured to use correct IP addresses (network interfaces) per application:
  - **SIP Interfaces** should be attached to the primary addresses – **eth0** and **eth1** – that reside behind Azure Load Balancer.
  - **Media Realms** should be attached to the secondary addresses – **eth0:1** and **eth1:1** – that don't reside behind Azure Load Balancer.
- VoIP equipment that communicates with Mediant VE Container (e.g., SIP Trunk or Contact Center) should be configured to use Load Balancer's IP addresses – **internal-lb:eth0** and **public-lb:eth1** – to send signaling (SIP) traffic to Mediant VE Container. It should expect to receive media (RTP) traffic from the local IP address of the active Mediant VE Container instance – **eth0:1** and **eth1:1**. For the **eth0** network interface, it should also expect receiving signaling (SIP) traffic from the local IP address of the active Mediant VE Container instance – **eth0** – because the internal Azure Load Balancer doesn't perform NAT translation.
- Management clients (e.g., Web browser or SSH client) should use Load Balancer's IP address – **internal-lb:eth0** – to access the management interface on the active Mediant VE Container instance. They may also use internal IP addresses of each instance – **sbc-1:eth0** and **sbc-2:eth0** – to access a specific Mediant VE Container instance.
  - For HA deployments that use public IP addresses on the network interface connected to the "main" subnet (**eth0**), NAT rules are created on the public Azure Load Balancer – **sbc-1-nat** and **sbc-2-nat** – that can be used to access the host machine's Linux shell on each Mediant VE Container instance.

5 Upgrading Mediant VE Container

Mediant VE Container doesn't support software upgrade using the firmware (.cmp) file. Instead, software upgrade is done using the installation package (**docker_debian_sip_F7.60A.xxx.yyy.tar.gz**).

The upgrade procedure depends on the Mediant VE Container deployment method:

- Installation Script – see Section 5.1, Upgrade Procedure if Deployed using Installation Script
- Stack Manager – see Section 5.2, Upgrade Procedure if Deployed using Stack Manager

5.1 Upgrade Procedure if Deployed using Installation Script

If you deployed Mediant VE Container using the installation script (see Section 2, Deploying Mediant VE Container using Installation Script), follow the procedure below to upgrade Mediant VE Container.

To upgrade Mediant VE Container if deployed using installation script:

1. Connect to the Linux virtual host through SSH or a serial console.
2. Log in as a user who has administrative privileges to run the **sudo** command.
3. Use an SCP or SFTP client (e.g., WinSCP) to transfer the Mediant VE Container installation package (**docker_debian_sip_F7.60A.xxx.yyy.tar.gz**) to the host.
4. Run the **load-image.sh** script to load and run the Mediant VE Container image from the installation package:

```
./load-image.sh docker_debian_sip_F7.60A.xxx.yyy.tar.gz
```

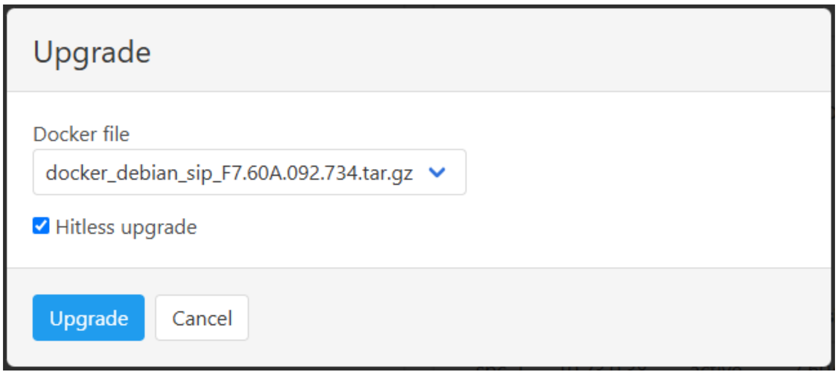
5.2 Upgrade Procedure if Deployed using Stack Manager

If you deployed Mediant VE Container using Stack Manager (see Section 4, Deploying Mediant VE Container using Stack Manager), follow the procedure below to upgrade Mediant VE Container.

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

To upgrade Mediant VE Container if deployed using Stack Manager:

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



Upgrade

Docker file: docker_debian_sip_F7.60A.092.734.tar.gz

☑ Hitless upgrade

5. Select the new Mediant VE Container image (**Docker file**).
6. If you want to perform the upgrade without service interruption (HA deployments only), select the **Hitless upgrade** check box; clear it if you want a faster upgrade.
7. Click **Upgrade** to start the upgrade.

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

Document #: LTRT-11032