

Stack Manager

Mediant Cloud Edition (CE) SBC |
Mediant Virtual Edition (VE) SBC

Version 7.6

Table of Contents

Notice	viii
Security Vulnerabilities.....	viii
WEEE EU Directive.....	viii
Customer Support.....	viii
Stay in the Loop with AudioCodes.....	viii
Abbreviations and Terminology.....	viii
Document Revision Record.....	viii
Documentation Feedback.....	ix
1 Introduction	2
2 Deployment	3
2.1 Operational Environment.....	3
2.2 Network Topology.....	3
2.3 Installation Prerequisites.....	4
2.3.1 Installation Prerequisites for Amazon Web Services (AWS) Environment.....	4
2.3.1.1 IAM Role for Stack Manager.....	4
2.3.1.2 Subnet and Elastic IP Addresses.....	7
2.3.2 Installation Prerequisites for Microsoft Azure Environment.....	8
2.3.2.1 Subnet and Public IP Addresses.....	8
2.3.3 Installation Prerequisites for Google Cloud Environment.....	9
2.3.3.1 Subnet and External IP Addresses.....	9
2.3.4 Installation Prerequisites for OpenStack Environment.....	10
2.3.4.1 Provider Versus Self-Service Networks.....	10
2.3.4.2 Subnet and Floating IP Addresses.....	10
2.4 Installation.....	11
2.4.1 Overview.....	11
2.4.2 Creating Amazon Web Services (AWS) Instance.....	12
2.4.3 Deploying Stack Manager on Microsoft Azure.....	15
2.4.4 Creating Google Cloud Virtual Machine.....	20
2.4.5 Creating OpenStack Instance.....	22
2.4.6 Installing Stack Manager Application.....	25
2.5 Accessing the Web Interface.....	26
2.6 Accessing the CLI.....	27
2.7 Upgrading Stack Manager.....	27
2.8 Post-installation Configuration.....	29
2.8.1 Post-installation Configuration on Amazon Web Services (AWS).....	29
2.8.1.1 Enabling Access to AWS API via IAM Role (Recommended Method).....	29
2.8.1.2 Enabling Access to AWS API via AWS Access Key (Alternative Method).....	29
2.8.2 Post-Installation Configuration on Microsoft Azure.....	30

2.8.2.1	Configuring the Azure Subscription ID	30
2.8.2.2	Enabling Access to Azure APIs via Managed Service Identity (Recommended Method)	31
2.8.2.3	Enabling Access to Azure APIs via Service Principal (Alternative Method)	38
2.8.3	Post-Installation Configuration on Google Cloud	39
2.8.3.1	Configuring Google Project ID	39
2.8.3.2	Enabling APIs in Project	40
2.8.3.3	Creating a Service Account	40
2.8.3.4	Enabling Access to Google Cloud APIs via Service Account (Recommended Method)	41
2.8.3.5	Enabling Access to Google Cloud APIs via Configuration File (Alternative Method)	41
2.8.4	Post-installation Configuration on OpenStack	42
2.8.5	Verifying Configuration	43
2.9	Runtime Data	44
2.9.1	Storing Runtime Data on AWS S3	44
2.9.2	Storing Runtime Data on Azure Storage Service	46
2.9.3	Storing Runtime Data on Google Cloud Storage Service	46
2.9.4	Storing Runtime Data on OpenStack Object Storage Service	47
2.9.5	Migrating Runtime Data from Local Disk to Storage Service	47
2.10	Resource Naming	47
2.11	Backup and Restore	48
2.12	Migrating to a New Virtual Machine	48
2.13	Providing Debug File for Troubleshooting	49
3	Web Interface	50
3.1	Accessing the Web Interface	50
3.2	Access Levels	51
3.3	Managing Users	52
3.3.1	Changing Your Own Password	54
3.3.2	Password Complexity	54
3.3.3	Password Reuse	55
3.4	Global Configuration	56
3.4.1	General Configuration Parameters	57
3.4.2	Change Password Block	57
3.4.3	Security Configuration Parameters	57
3.4.4	Microsoft Azure Parameters	59
3.4.5	Amazon AWS Parameters	60
3.4.6	Google Cloud Parameters	60
3.4.7	Openstack Parameters	61
3.4.8	Debug File Parameters	61
3.4.9	Advanced Parameters	62
3.5	Securing Connection to Web Interface	63

3.5.1	Configuring Hostname for Stack Manager Virtual Machine.....	63
3.5.2	Acquiring Certificate from Certificate Authority	65
3.5.3	Installing Let's Encrypt Certificates	66
3.5.4	Enforcing Secure Connection	66
3.6	Login via Azure Entra ID.....	67
3.7	Resetting Web Interface Credentials	70
3.8	Creating a New Stack.....	71
3.8.1	Creating Mediant CE in Amazon Web Services (AWS) Environment	72
3.8.1.1	Troubleshooting	75
3.8.2	Creating Mediant CE in Azure Environment.....	76
3.8.2.1	Troubleshooting	79
3.8.3	Creating Mediant CE in Google Cloud Environment	80
3.8.4	Creating Mediant CE in OpenStack Environment.....	83
3.8.5	Creating Mediant VE in Amazon Web Services (AWS) Environment	85
3.8.5.1	Troubleshooting	87
3.8.6	Creating Mediant VE in Azure Environment.....	88
3.8.6.1	Troubleshooting	90
3.8.7	Creating Mediant VE in Google Cloud Environment	91
3.8.8	Creating VoiceAI Connect in Amazon Web Services (AWS) Environment.....	93
3.8.9	Creating VoiceAI Connect in Azure Environment	96
3.8.10	Creating VoiceAI Connect in Google Cloud Environment.....	99
3.8.11	Advanced Configuration.....	101
3.8.11.1	Advanced Configuration for Mediant CE.....	101
3.8.11.2	Advanced Configuration for Mediant VE.....	121
3.8.11.3	Advanced Configuration for VoiceAI Connect	134
3.9	Checking Stack State and Configuration	145
3.9.1	Viewing IP Addresses of Stack Components	146
3.9.2	Checking Deployment Environment.....	147
3.9.3	Checking Connectivity	147
3.9.4	Updating Connectivity.....	148
3.10	Active Alarms.....	148
3.11	Performing Operations on Stack	149
3.12	Scaling Mediant CE Stack.....	150
3.12.1	Scale Out Operation	150
3.12.2	Scale In Operation	151
3.12.3	Scale To Operation	151
3.13	Automatic Scaling	152
3.13.1	Cool Down Period.....	153
3.13.2	Auto Scale Step.....	153
3.13.3	Changing Cluster Size at Specific Time of Day	153
3.14	Modifying Stack Configuration	154
3.14.1	Update Operation	156

3.14.2	Modifiable Parameters for Mediant CE.....	157
3.14.3	Modifiable Parameters for Mediant VE	160
3.15	Stopping and Starting Stack	161
3.16	Rebooting Stack Components	161
3.17	Healing Stack	162
3.17.1	Automatic Healing.....	162
3.18	Deleting Stack.....	163
3.19	Rebuilding Stack	163
3.20	Managing Files.....	164
3.21	Upgrading Stack.....	164
3.21.1	Hosting Software Load (CMP) Files on Stack Manager	166
3.21.2	Upgrading Software on Idle Media Components	167
3.22	Shelving and Unshelving Stack	167
3.23	Resetting Stack Password.....	168
3.24	Sending INI File	168
3.25	Stack Deployment Details	169
3.25.1	Deployment Method.....	169
3.25.2	Adjusting Security Groups	170
3.25.2.1	Modifying Security Groups Created by Stack Manager in Azure Environment	170
3.25.3	Using Pre-Defined Public IP Addresses.....	171
3.25.4	Using Pre-Defined Private IP Addresses	173
3.25.5	Using Pre-Defined Virtual IP Addresses.....	177
3.25.6	Using Pre-Defined IPv6 Addresses	178
4	CLI Interface.....	179
4.1	Accessing CLI Interface	179
4.2	Invocation.....	179
4.3	Usage Information.....	179
4.4	Managing Users.....	180
4.5	Global Configuration	182
4.6	Listing Available Stacks.....	182
4.7	Creating a New Stack.....	183
4.7.1	Creating Stack Configuration File via SBC Cluster Configuration Tool (Recommended Method).....	183
4.7.2	Creating Stack Configuration File Manually (Alternative Method)	189
4.7.2.1	Sample Configuration File	190
4.7.3	Creating a New Stack.....	197
4.8	Checking Stack State and Configuration	198
4.8.1	Checking Idle Media Components.....	201
4.8.2	Viewing IP Addresses of Stack Components	201
4.8.3	Checking Deployment Environment.....	202
4.8.4	Checking Connectivity	203

4.8.5	Updating Connectivity	204
4.9	Scaling Mediant CE Stack.....	204
4.9.1	Scale Out Operation	204
4.9.2	Scale In Operation	205
4.9.3	Scale To Operation	205
4.10	Modifying Stack Configuration	206
4.10.1	Update Operation	207
4.11	Stopping and Starting the Stack	209
4.11.1	Stopping Stack	209
4.11.2	Starting Stack.....	209
4.12	Rebooting Stack Components	210
4.13	Deleting Stack.....	210
4.13.1	Purging Deleted Stack	211
4.14	Healing Stack	211
4.15	Rebuilding Stack	212
4.16	Managing Files.....	213
4.17	Upgrading Stack.....	214
4.17.1	Hosting Software Load (CMP) Files on Stack Manager	215
4.17.2	Upgrading Software on Idle Media Components	215
4.18	Shelving and Unshelving the Stack.....	216
4.18.1	Shelving Stack.....	216
4.18.2	Unshelving Stack	216
4.19	Resetting Stack Password.....	217
4.20	Sending INI File	217
4.21	Multiple Operations	218
5	REST API.....	219
5.1	Overview.....	219
5.2	Asynchronous Tasks	220
5.3	Authorization.....	221
5.3.1	Authorization via Azure Entra ID	221
5.4	Discovery	223
5.5	Managing Users.....	223
5.5.1	Listing Users	223
5.5.2	Adding User	224
5.5.3	Modifying User	225
5.5.4	Deleting User	225
5.6	Global Configuration	226
5.6.1	Updating Global Configuration.....	227
5.7	Listing Available Stacks.....	227
5.8	Creating New Stack	228

5.8.1	Getting Stack Template	229
5.9	Checking Stack State and Configuration	231
5.9.1	Viewing IP Addresses of Stack Components	234
5.9.2	Checking Deployment Environment.....	236
5.9.3	Checking Connectivity	236
5.9.4	Updating Connectivity.....	237
5.10	Scaling Mediant CE Stack.....	238
5.10.1	Scale Out Operation	238
5.10.2	Scale In Operation	238
5.10.3	Scale To Operation	239
5.11	Modifying Stack Configuration	240
5.11.1	Update Operation	241
5.12	Stopping and Starting Stack	242
5.12.1	Stopping Stack.....	242
5.12.2	Starting Stack.....	242
5.13	Rebooting Stack Components	243
5.14	Deleting Stack.....	244
5.14.1	Purging Deleted Stack	244
5.15	Healing Stack	245
5.16	Rebuilding Stack	245
5.17	Managing Files.....	246
5.17.1	Listing Files	246
5.17.2	Adding File.....	246
5.17.3	Deleting File.....	247
5.18	Upgrading Stack.....	248
5.18.1	Hosting Software Load (CMP) Files on Stack Manager	249
5.18.2	Upgrading Software on Idle Media Components	250
5.19	Shelving and Unshelving Stack	250
5.19.1	Shelving Stack.....	250
5.19.2	Unshelving Stack	250
5.20	Resetting Stack Password.....	251
5.21	Sending INI File	252
6	Operational Logs	253
6.1	Web Server Logs.....	254
7	Stacks Management	255
7.1	Automatic Stop / Start / Shelve.....	255
7.2	Tagging Stack Resources	256
7.3	Integration with Azure Application Insights.....	256
8	Secure Deployment.....	258

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-22-2025

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.

Abbreviation	Description
MC	Media Component
SC	Signaling Component

Document Revision Record

LTRT	Description
28967	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

Stack Manager is used for managing 'software stacks' deployed in virtual environments. It implements the complete stack lifecycle, including:

- Stack deployment
- Stack termination
- Manual stack size adjustment – using user-initiated scale-in / scale-out
- Automatic stack size adjustment – using automatic scaling
- Stack configuration update

Current implementation supports Mediant CE (Cloud Edition) and Mediant VE (Virtual Edition) SBC in the following environments:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud
- OpenStack

Stack Manager implements VNFM (Virtual Network Function Manager) functionality as defined in the NFV Management and Organization (MANO) architectural framework.

The following management interfaces are provided:

- Web interface
- Command line interface (CLI)
- REST API

2 Deployment

2.1 Operational Environment

Stack Manager is mostly written in Python and can be installed on one of the following operating systems:

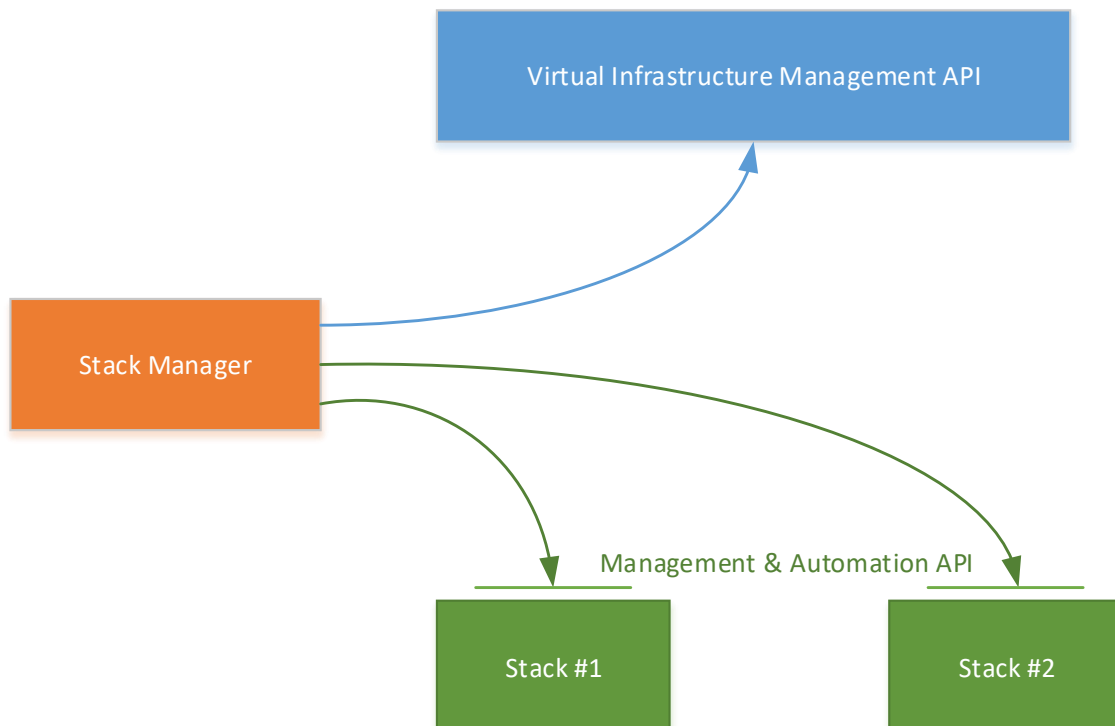
Operating System	Supported Versions
Ubuntu	18.04, 20.04, 22.04, 24.04
Debian	10, 11, 12
Red Hat Enterprise Linux (RHEL)	8, 9
CentOS / CentOS Stream	8, 9
Amazon Linux	2, 2023
Rocky Linux	8, 9
Alma Linux	8, 9

2.2 Network Topology

Stack Manager needs to have access to the following APIs for correct operation:

- Virtual Infrastructure Management API (e.g., AWS API) for deploying stack components and managing their lifecycle.
- Management API of the deployed stack (e.g., REST API of Mediant CE) for assessing operational status of deployed stack instances and managing their configuration and state.

Figure 2-1: Stack Manager Deployment Topology



2.3 Installation Prerequisites

2.3.1 Installation Prerequisites for Amazon Web Services (AWS) Environment

Prior to installing Stack Manager in the Amazon Web Services (AWS) environment, make sure that you meet the following prerequisites:

- You have an AWS account. If you don't have one, you can sign up for one on Amazon's website at <http://aws.amazon.com/>.
- You have created an IAM Role that enables Stack Manager to access all needed AWS APIs. For more information, see Section 2.3.1.1.
- Security groups of the "Main Subnet", where Stack Manager will be deployed, allow Stack Manager to communicate with both the AWS APIs and the deployed Mediant VE/CE stack instances, using the HTTPS protocol (Port 443).

2.3.1.1 IAM Role for Stack Manager

The following IAM role ensures that Stack Manager can access all needed AWS APIs for successful stack deployment and management. This role must be attached to the Stack Manager's virtual instances, as described in Section 2.4.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:*",
        "cloudformation:*",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:PutMetricAlarm",
        "iam:PassRole",
        "iam:ListInstanceProfiles",
        "iam:CreateServiceLinkedRole"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

For multi-zone Mediant VE and CE deployments that use Network Load Balancer, add the following additional action:

```
"elasticloadbalancing:*"
```

To create an IAM Role

1. Open the AWS IAM console (<https://console.aws.amazon.com/iam>).
2. Navigate to the **Policies** screen:
 - a. Click **Create**.
 - b. Select the **JSON** tab, copy-and-paste the IAM policy rules listed above, and then click **Review policy**.
 - c. Enter the IAM policy name (e.g., "STACK_MGR"), and then click **Create policy**.
3. Navigate to the **Roles** screen:
 - a. Click **Create role**.
 - b. Choose **EC2** use case, and then click **Next: permissions**.
 - c. Search for the IAM policy created in the previous step, select it, and then click **Next: tags**.
 - d. Click **Next: review**.
 - e. Enter the IAM role name (e.g., "STACK_MGR"), and then click **Create role**.

For multi-zone Mediant VE and CE deployments that use Network Load Balancer, if you receive the following error during stack creation:

```
User is not authorized to perform iam:CreateServiceLinkedRole on
resource arn:aws:iam::<account-id>:role/aws-service-
role/elasticloadbalancing.amazonaws.com/...
```

create the corresponding service linked role through the AWS CLI:

```
aws iam create-service-linked-role \
  --aws-service-name elasticloadbalancing.amazonaws.com
```

or add the following to the IAM role assigned to the Stack Manager:

```
{
  "Effect": "Allow",
  "Action": "iam:CreateServiceLinkedRole",
  "Resource": "arn:aws:iam::*:role/aws-service-
role/elasticloadbalancing.amazonaws.com/*",
  "Condition": {"StringLike": {
    "iam:AWSServiceName": "elasticloadbalancing.amazonaws.com"
  }}
}
```

The IAM role specified above grants access to all EC2 and CloudFormation APIs. Stack Manager currently uses the following specific services from these APIs:

```
"ec2:AllocateAddress",
"ec2:AssociateAddress",
"ec2:AssignPrivateIpAddresses",
"ec2:AttachNetworkInterface",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:AuthorizeSecurityGroupIngress",
"ec2:CreateNetworkInterface",
"ec2:CreatePlacementGroup",
"ec2:CreateSecurityGroup",
"ec2:CreateTags",
"ec2>DeleteNetworkInterface",
```

```
"ec2:DeletePlacementGroup",
"ec2:DeleteSecurityGroup",
"ec2:DeleteTags",
"ec2:DescribeAddresses",
"ec2:DescribeImages",
"ec2:DescribeInstances",
"ec2:DescribeInstanceStatus",
"ec2:DescribeKeyPairs",
"ec2:DescribeNetworkInterfaces",
"ec2:DescribePlacementGroups",
"ec2:DescribeRegions",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeVpcs",
"ec2:DetachNetworkInterface",
"ec2:DisassociateAddress",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ReleaseAddress",
"ec2:RevokeSecurityGroupEgress",
"ec2:RevokeSecurityGroupIngress",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:StopInstances",
"ec2:TerminateInstances",
"ec2:UnassignPrivateIpAddresses",
"cloudformation:CreateStack",
"cloudformation:DeleteStack",
"cloudformation:DescribeStackEvents",
"cloudformation:DescribeStackResources",
"cloudformation:DescribeStacks",
"cloudformation:GetTemplate",
"cloudformation:ListStacks",
"cloudformation:UpdateStack",
"cloudformation:CreateChangeSet",
"cloudformation>DeleteChangeSet",
"cloudformation:DescribeChangeSet",
"cloudformation:ExecuteChangeSet"
```



The above list might change as Stack Manager implementation is updated and new functionality is added.

For multi-zone Mediant VE and CE deployments that use Virtual IP addresses, the following additional services are consumed by Stack Manager:

```
"ec2:CreateRoute",
"ec2>DeleteRoute",
"ec2:DescribeRouteTables",
"ec2:ReplaceRoute"
```

For multi-zone Mediant VE and CE deployments that use Network Load Balancer, the following additional services are consumed by Stack Manager:

```
"elasticloadbalancing:CreateListener",  
"elasticloadbalancing:CreateLoadBalancer",  
"elasticloadbalancing:CreateRule",  
"elasticloadbalancing:CreateTargetGroup",  
"elasticloadbalancing>DeleteListener",  
"elasticloadbalancing>DeleteLoadBalancer",  
"elasticloadbalancing>DeleteRule",  
"elasticloadbalancing>DeleteTargetGroup",  
"elasticloadbalancing:ModifyListener",  
"elasticloadbalancing:ModifyLoadBalancerAttributes",  
"elasticloadbalancing:ModifyRule",  
"elasticloadbalancing:ModifyTargetGroup",  
"elasticloadbalancing:RegisterTargets",  
"elasticloadbalancing:DescribeListeners",  
"elasticloadbalancing:DescribeLoadBalancerAttributes",  
"elasticloadbalancing:DescribeLoadBalancers",  
"elasticloadbalancing:DescribeRules",  
"elasticloadbalancing:DescribeTargetGroups",  
"elasticloadbalancing:DescribeTargetHealth",  
"elasticloadbalancing:AddTags",  
"elasticloadbalancing:RemoveTags",
```

2.3.1.2 Subnet and Elastic IP Addresses

Stack Manager uses the following IP addresses when communicating with Mediant VE/CE stack instances that it deploys:

- If the stack instance has a public IP address (Elastic IP) assigned to its management interface, Stack Manager uses this public IP address to access the stack instance's management REST API.
- Otherwise, Stack Manager uses the private IP address of the stack instance's management interface.

To enable Stack Manager's access to the deployed Mediant VE/CE stack's management APIs, it's recommended to deploy Stack Manager to the same "Main Subnet" that is used for carrying management traffic of the deployed Mediant VE/CE stack(s). Alternatively, you should ensure connectivity between the subnet where Stack Manager is deployed and the "Main Subnet" of the deployed Mediant VE/CE stack(s), for example, through VPC peering.

Stack Manager also needs to communicate with AWS APIs, which are accessible via public IP addresses. Therefore, it should either be assigned with an Elastic IP address or placed behind a NAT Gateway.

2.3.2 Installation Prerequisites for Microsoft Azure Environment

Prior to installing Stack Manager in the Microsoft Azure environment, make sure that you meet the following prerequisites:

- You have an Azure account. If you don't have one, you can sign up for one on Microsoft's website at <http://azure.microsoft.com>.
- Security groups of the "Main Subnet", where Stack Manager will be deployed, allow Stack Manager to communicate with both the Azure API and the deployed Mediant VE/CE stack instances, using the HTTPS protocol (Port 443).

2.3.2.1 Subnet and Public IP Addresses

Stack Manager uses the following IP addresses when communicating with Mediant VE/CE stack instances that it deploys:

- If the stack instance has a public IP address assigned to its management interface, Stack Manager uses this public IP address to access the stack instance's management REST API.
- Otherwise, Stack Manager uses the private IP address of the stack instance's management interface.

To enable Stack Manager's access to the deployed Mediant VE/CE stack's management APIs, it's recommended to deploy Stack Manager to the same "Main Subnet" that is used for carrying management traffic of the deployed Mediant VE/CE stack(s). Alternatively, you should ensure connectivity between the subnet where Stack Manager is deployed and the "Main Subnet" of the deployed Mediant VE/CE stack(s), for example, through Vnet peering.

Stack Manager also needs to communicate with Azure APIs, which are accessible via public IP addresses. Therefore, it should either be assigned with a public IP address or placed behind a NAT Gateway.

2.3.3 Installation Prerequisites for Google Cloud Environment

Prior to installing Stack Manager in the Google Cloud environment, make sure that you meet the following prerequisites:

- You have a Google Cloud account. If you don't have one, you can sign up for one on Google's website at <http://cloud.google.com>.
- Firewall Rules of the "Main Subnet", where Stack Manager will be deployed, allow Stack Manager to communicate with both the Google Cloud API and the deployed Mediant VE/CE stack instances, using the HTTPS protocol (Port 443).

2.3.3.1 Subnet and External IP Addresses

Stack Manager uses the following IP addresses when communicating with Mediant VE/CE stack instances that it deploys:

- If the stack instance has an external IP address assigned to its management interface, Stack Manager uses this external IP address to access the stack instance's management REST API.
- Otherwise, Stack Manager uses the internal IP address of the stack instance's management interface.

To enable Stack Manager's access to the deployed Mediant VE/CE stack's management APIs, it's recommended to deploy Stack Manager to the same "Main Subnet" that is used for carrying management traffic of the deployed Mediant VE/CE stack(s). Alternatively, you should ensure connectivity between the subnet where Stack Manager is deployed and the "Main Subnet" of the deployed Mediant VE/CE stack(s), for example, through VPC Network peering.

Stack Manager also needs to communicate with Google Cloud APIs, which are accessible via public IP addresses. Therefore, it should either be assigned with an External IP address or placed behind a NAT Gateway.

2.3.4 Installation Prerequisites for OpenStack Environment

Prior to installing Stack Manager in the OpenStack environment, make sure that you meet the following prerequisites:

- The OpenStack environment contains the following components:
 - Nova
 - Neutron
 - Cinder
 - Glance
 - Heat
- Security groups of the "Main Subnet", where Stack Manager will be deployed, allow Stack Manager to communicate with both the OpenStack API and the deployed Mediant CE stack instances, using the HTTPS protocol (Port 443).

2.3.4.1 Provider Versus Self-Service Networks

Stack Manager supports deployment both in provider (flat) and self-service networks.

2.3.4.2 Subnet and Floating IP Addresses

Stack Manager uses the following IP addresses when communicating with Mediant VE/CE stack instances that it deploys:

- If the stack instance has a Floating IP address assigned to its management interface, Stack Manager uses this Floating IP address to access the stack instance's management REST API.
- Otherwise, Stack Manager uses the private IP address of the stack instance's management interface.

To enable Stack Manager's access to the deployed Mediant VE/CE stack's management APIs, it's recommended to deploy Stack Manager to the same "Main Subnet" that is used for carrying management traffic of the deployed Mediant VE/CE stack(s).

Stack Manager also needs to communicate with OpenStack automation APIs. Make sure that your network topology enables such communication.

2.4 Installation

2.4.1 Overview

For Microsoft Azure, Stack Manager is available in the Azure Marketplace. Therefore, its deployment consists of a single step, as described in Section 2.4.3, Deploying Stack Manager on Microsoft Azure.

For other cloud environments, Stack Manager installation consists of two steps:

1. Creating the Instance / Virtual Machine: This step differs, depending on the virtual environment. For detailed instructions, see the following sections:
 - Section 2.4.2, Creating Amazon Web Services (AWS) Instance
 - Section 2.4.4, Creating Google Cloud Virtual Machine
 - Section 2.4.5, Creating OpenStack Instance
2. Installing the Stack Manager application: For detailed instructions, see Section 2.4.6, Installing Stack Manager Application



It's also possible to install Stack Manager in Azure on a VM of your choice. To do this, create a VM with the supported OS flavor/version, as described in Section 2.1, Operational Environment, and then proceed with Stack Manager application installation, as described in Section 2.4.6, Installing Stack Manager Application.

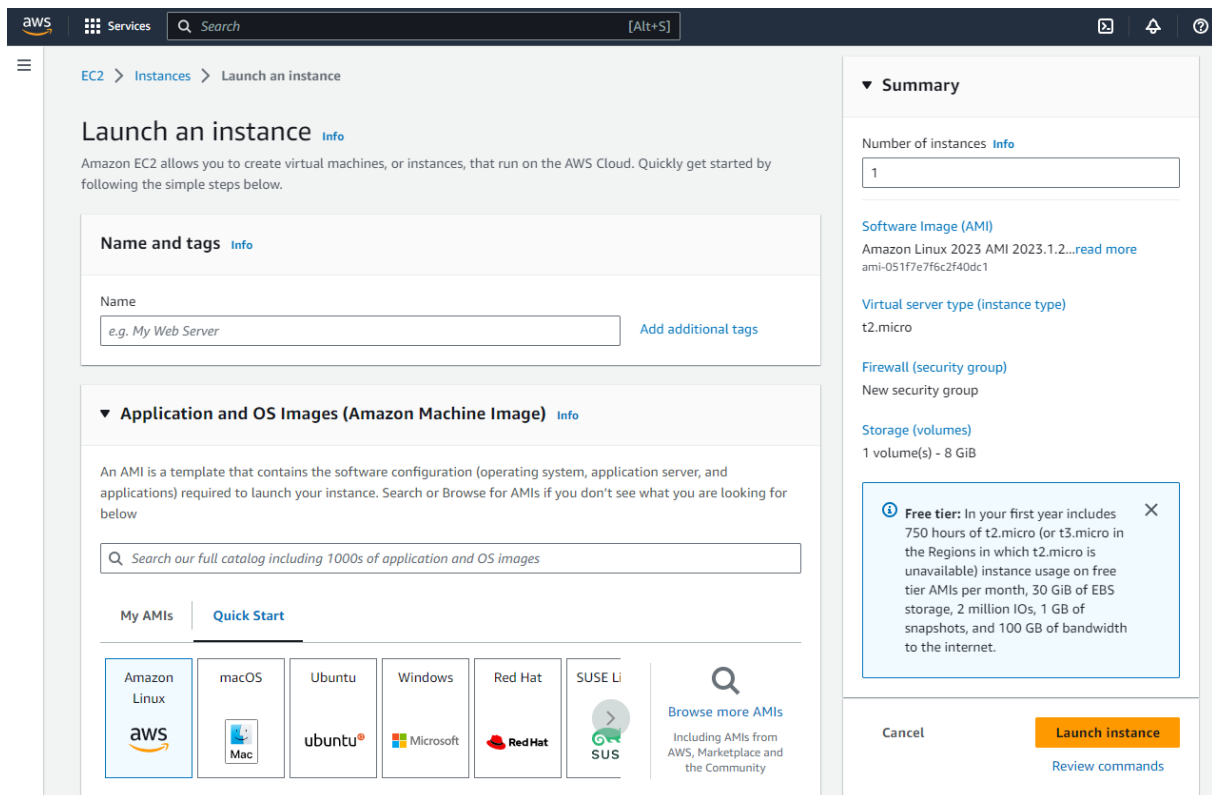
2.4.2 Creating Amazon Web Services (AWS) Instance

The following procedure describes how to create a new AWS instance for running the Stack Manager application.

To create a new AWS instance for running Stack Manager application:

1. Open the AWS EC2 Console at <http://console.aws.amazon.com/ec2>.
2. In the Instances screen, click **Launch Instance**.

Figure 2-2: Launching EC2 Instance



3. In the 'Name' field, type the name of the EC2 Instance (e.g., "STACK-MGR").
4. Under the **Application and OS Images (Amazon Machine Image)** group, choose one of the supported Linux distributions as listed in Section Operational Environment (e.g., "Amazon Linux").
5. From the 'Instance Type' drop-down list, select **t2.small**.
6. From the 'Key pair (login)' drop-down list, select the SSH key that is used to log into the created EC2 instance.
7. For **Network settings**, click **Edit**, and then in the 'VPC' and 'Subnet' drop-down list, select where the EC2 Instance will be deployed. If you plan to deploy a single Mediant VE/CE stack, it's recommended to deploy Stack Manager in the same VPC and the "main subnet" that is used for connecting the management interface of the deployed Mediant VE/CE stack.

- From the 'Auto-assign public IP' drop-down list, select whether you want to assign a public IP address to the deployed EC2 instance. Note that if you select **Disable** you can access the deployed EC2 instance only via a private IP address (from some other instance / machine connected to the same VPC).

Figure 2-3: Launching EC2 Instance – Configuring Network Settings

▼ Network settings [Info](#)

VPC - *required* [Info](#)

vpc-f8b7159d (default) (default) ↕

172.31.0.0/16

Subnet [Info](#)

subnet-a704e9fe main ↕

VPC: vpc-f8b7159d Owner: 516086831279 Availability Zone: us-east-1c
IP addresses available: 4077 CIDR: 172.31.0.0/20

↻ Create new subnet [↗](#)

Auto-assign public IP [Info](#)

Enable

- For **Inbound Security Group Rules**, keep the default rule that allows access to the deployed EC2 instance via SSH protocol, and add two new rules that allow access via HTTP and HTTPS protocols.

Figure 2-4: Launching EC2 Instance – Configuring Inbound Security Rules

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

Type [Info](#) ssh

Protocol [Info](#) TCP

Port range [Info](#) 22

Source type [Info](#) Anywhere

Source [Info](#) [Add CIDR, prefix list or security](#)

0.0.0.0/0 ✕

Description - optional [Info](#) e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0) [Remove](#)

Type [Info](#) HTTP

Protocol [Info](#) TCP

Port range [Info](#) 80

Source type [Info](#) Anywhere

Source [Info](#) [Add CIDR, prefix list or security](#)

0.0.0.0/0 ✕

Description - optional [Info](#) e.g. SSH for admin desktop

▼ Security group rule 3 (TCP, 443, 0.0.0.0/0) [Remove](#)

Type [Info](#) HTTPS

Protocol [Info](#) TCP

Port range [Info](#) 443

Source type [Info](#) Anywhere

Source [Info](#) [Add CIDR, prefix list or security](#)

0.0.0.0/0 ✕

Description - optional [Info](#) e.g. SSH for admin desktop

- For **Configure Storage**, leave it at the default values (8 GiB, gp3).

- Click **Advanced Details**, and then from the 'IAM instance profile' drop-down list, select the IAM role that you created for Stack Manager in Section IAM Role for Stack Manager.

Figure 2-5: Launching EC2 Instance – Specifying IAM Role

The screenshot shows the 'Advanced details' section of the AWS Management Console. It includes the following elements:

- Purchasing option:** A checkbox for 'Request Spot Instances' is currently unchecked.
- Domain join directory:** A dropdown menu is set to 'Select', with a 'Create new directory' link and icon to the right.
- IAM instance profile:** A dropdown menu is set to 'STACK_MGR' with the ARN 'arn:aws:iam::516086831279:instance-profile/STACK_MGR' visible below it. A 'Create new IAM profile' link and icon are also present to the right.

- Click **Launch instance**.
- Wait until the instance is successfully launched.
- Connect to the instance through SSH using the default username and configured SSH key. The default username depends on the operating system:

Operating System	Default Username
Debian	admin
Ubuntu	ubuntu
Amazon Linux	ec2-user
RHEL	ec2-user
CentOS	centos
Rocky Linux	rocky
Alma Linux	ec2-user

- If you have chosen to auto-assign a public IP address during EC2 instance creation, your instance has been assigned a Public IP address that changes when the instance is stopped or started. If you want Stack Manager's Public IP address to remain unchanged, create an Elastic IP address and attach it to the instance.
- Continue with Stack Manager installation, as described in Section 2.4.6, Installing Stack Manager Application.

2.4.3 Deploying Stack Manager on Microsoft Azure

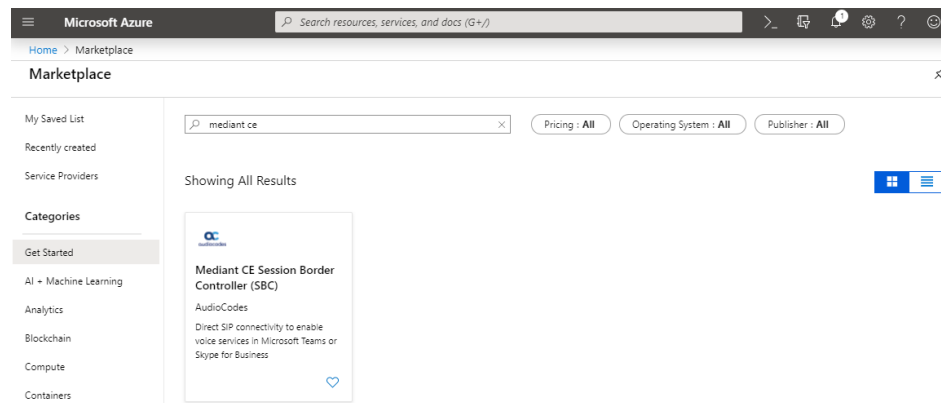
Stack Manager is available in Microsoft Azure Marketplace. Therefore, it's recommended that you deploy it from there, as described below. The default Stack Manager image is based on Debian 11 (Bullseye) Linux distribution.

Alternatively, you can create a virtual machine with one of the supported Linux distributions, as specified in Section 2.1, Operational Environment, and continue with Stack Manager installation, as described in Section 2.4.6, Installing Stack Manager Application.

To deploy Stack Manager from Microsoft Azure Marketplace:

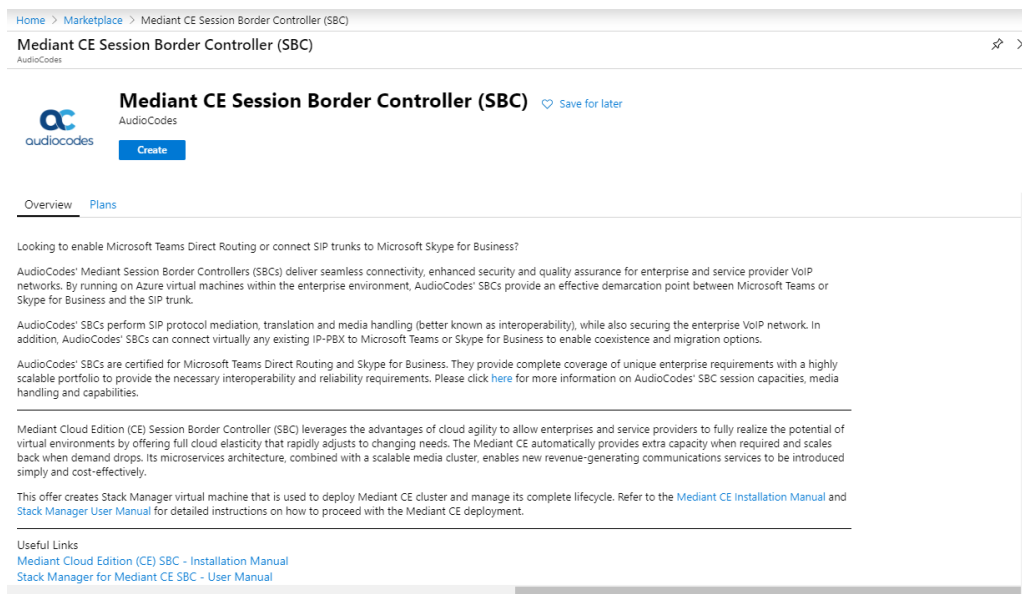
1. Open the Azure portal at <https://portal.azure.com/>.
2. Navigate to Azure Marketplace (**All services > Marketplace**).
3. Search for the product "Mediant CE Session Border Controller (SBC)" published by AudioCodes.

Figure 2-6: Azure Marketplace



4. Click the "Mediant CE Session Border Controller (SBC)" product; the Mediant CE Product overview screen appears.

Figure 2-7: Mediant CE SBC Product Offer



5. Click **Create**; a configuration wizard starts with the Basics page (Step 1).

6. In the **Basics** step, do the following:

Figure 2-8: Basics – Step 1

The screenshot shows the 'Basics' configuration page for a Mediant CE Session Border Controller (SBC). On the left, a navigation pane lists four steps: 1. Basics (selected), 2. Virtual Machine Settings, 3. Summary, and 4. Buy. The main area contains the following fields and options:

- Virtual Machine name ***: Text input field containing 'stack-mgr'.
- Username ***: Text input field containing 'stackmgr' with a green checkmark.
- Authentication type ***: Radio buttons for 'Password' (selected) and 'SSH public key'.
- Password ***: Password input field with masked characters and a green checkmark.
- Confirm password ***: Password input field with masked characters and a green checkmark.
- Subscription**: Drop-down menu showing 'SBC Lab'.
- Resource group ***: Drop-down menu showing '(New) StackMgrRG' with a 'Create new' link below it.
- Location ***: Drop-down menu showing '(Europe) West Europe'.

An 'OK' button is located at the bottom of the form.

- In the 'Virtual Machine name' field, enter a unique name for the new virtual machine.
- In the 'Username' field, enter a username.
- In the 'Authentication type' field, choose an appropriate authentication type, and then enter the 'Password' or 'SSH public key' accordingly. These credentials are used to connect to the deployed Stack Manager's CLI interface through SSH.



Azure imposes some limitations on the username and password. For example, it prohibits the use of "Admin" for the 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.

- From the 'Subscription' drop-down list, select a proper subscription for your deployment.
 - Under 'Resource group', click **Create new**, and then enter a new Resource Group name for your deployment.
 - From the 'Location' drop-down list, select a proper location for your deployment.
 - Click **OK**; the Virtual Machine Settings page (Step 2) appears.
7. In the Virtual Machine Settings step, do the following:

Figure 2-9: Virtual Machine Settings – Step 2

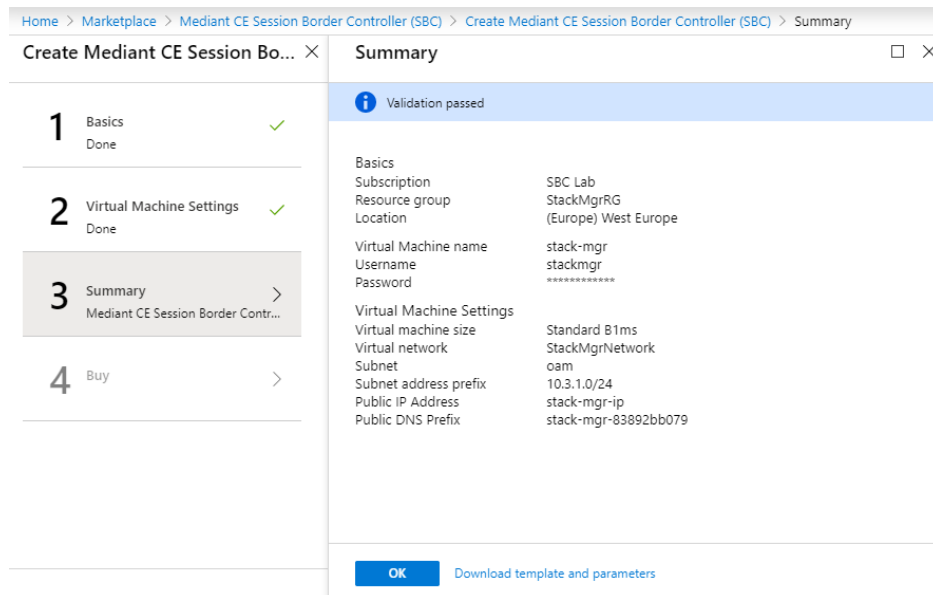
- a. Choose the Virtual machine size. Standard_B1ms instance is recommended for most deployments.
- b. Choose the virtual network where Stack Manager will be deployed. Specify the same network where you intend to deploy the Mediant VE/CE stack(s).
- c. Configure the subnet that Stack Manager will be connected to. Specify the same subnet that will be used for carrying management traffic for the deployed Mediant VE/CE stack(s).
- d. Configure a Public IP address to use Standard SKU:

Figure 2-10: Virtual Machine Settings Step – Creating Public IP Address

- e. Click **OK.**; the Summary page (Step 3) appears.

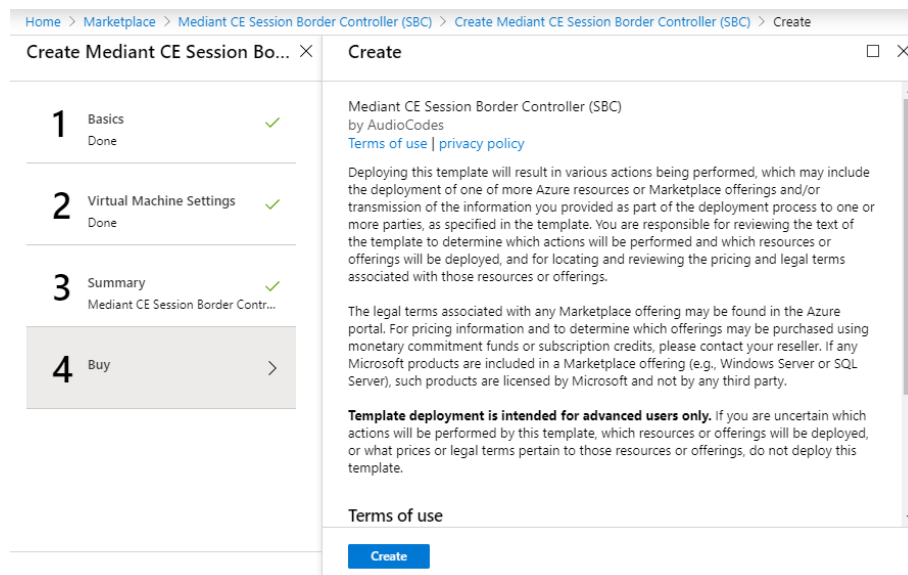
- In the Summary step, review your virtual machine configuration.

Figure 2-11: Summary – Step 3



- Click **OK**; the Buy page (Step 4) appears.
- Review the Mediant CE SBC terms of use.

Figure 2-12: Buy – Step 4



- Click **Create** to start the virtual machine deployment.
- Wait until the virtual machine deployment is complete, and then open the Virtual Machines screen (All services > Virtual Machines).
- Select the Stack Manager virtual machine.

14. In the Overview screen, view the public IP address assigned to it.

Figure 2-13: Determining Public IP Address

The screenshot shows the Azure portal interface for a virtual machine named 'stack-mgr'. The 'Overview' tab is selected, displaying various system details. The 'Public IP address' is highlighted with a red box and is '51.105.185.252'. Other details include: Resource group: StackMgrRG, Status: Running, Location: West Europe, Subscription: SBC Lab, Computer name: stack-mgr, Operating system: Linux (ubuntu 18.04), and Size: Standard B1ms (1 vcpu, 2 GiB memory).

15. In the Networking screen, verify that the following ports are open for inbound traffic:

Port	Protocol	Purpose
22	TCP	SSH connection to Stack Manager's CLI interface.
80	TCP	HTTP connection to Stack Manager's Web interface.
443	TCP	HTTPS connection to Stack Manager's Web interface.

16. If any port is missing, click **Add inbound port rule** and then add the port.

Figure 2-14: Checking Inbound Port Rules

The screenshot shows the Azure portal interface for the 'Networking' tab of the virtual machine 'stack-mgr'. It displays the network interface 'stack-mgr-nic' with a public IP of 51.105.185.252. Under 'Inbound port rules', a table lists the configured rules:

Priority	Name	Port	Protocol	Source	Destination	Action
500	ssh	22	TCP	Any	Any	Allow
510	http	80	TCP	Any	Any	Allow
520	https	443	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalanc...	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

17. Continue with post-installation configuration, as described in Section 2.8.2, Post-Installation Configuration on Microsoft Azure.

2.4.4 Creating Google Cloud Virtual Machine

The following procedure describes how to create a new Google Cloud virtual machine (VM) for running the Stack Manager application.

To create a new Google Cloud virtual machine for running Stack Manager application:

1. Open the Google Cloud Console at <https://console.cloud.google.com/compute>.
2. On the VM Instances page, click **Create Instance**.
3. In the 'Name' field, enter a unique name for the new virtual machine.
4. Choose the Region and Zone where Stack Manager will be deployed.
5. Under the 'Machine Type' group, choose **e2-small** (2 shared vCPUs, 2-GB memory).
6. Under the 'Boot disk' group, choose one of the supported Linux distributions, as specified in Section 2.1, Operational Environment, for example, **Debian GNU/Linux 11 (bullseye)**.
7. Under the 'Firewall' group, select the **Allow HTTP traffic** and **Allow HTTPS traffic** check boxes.
8. Click **Management, security, disks, networking, sole tenancy**.
9. In the **Networking** tab for the 'Network interface', choose the "Main Network" for connecting to the management interface of the deployed Mediant VE/CE stack(s).
10. If you want to be able to connect to Stack Manager's CLI interface through a regular SSH client (e.g., PuTTY) and not through the Google Cloud dashboard, configure the SSH keys under the **Security** tab. Note that the username is provided as the last part of the encoded key. For example, in the following SSH key, "admin" is the username:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAA...0Sknr admin
```
11. Click **Create**.

Figure 2-15: Create Google Cloud Instance

← Create an instance HELP ASSISTANT

To create a VM instance, select one of the options:

- New VM instance**
Create a single VM instance from scratch
- New VM instance from template
Create a single VM instance from an existing template
- New VM instance from machine image
Create a single VM instance from an existing machine image
- Marketplace
Deploy a ready-to-go solution onto a VM instance

Name *
stackmgr

Labels
+ ADD LABELS

Region *
us-central1 (Iowa) Region is permanent

Zone *
us-central1-a Zone is permanent

Machine configuration

Machine family
GENERAL-PURPOSE COMPUTE-OPTIMIZED MEMORY-OPTIMIZED GPU
Machine types for common workloads, optimized for cost and flexibility

Series
E2

CPU platform selection based on availability

Machine type
e2-small (2 vCPU, 2 GB memory)

	vCPU	Memory
	1 shared core	2 GB

✓ CPU PLATFORM AND GPU

Display device
Enable to use screen capturing and recording tools.
 Enable display device

Confidential VM service
 Enable the Confidential Computing service on this VM instance.

Container
Deploy a container image to this VM instance
[DEPLOY CONTAINER](#)

Boot disk

Name	instance-1
Type	New balanced persistent disk
Size	10 GB
Image	Debian GNU/Linux 11 (bullseye)

[CHANGE](#)

Identity and API access

Service accounts
Service account
Compute Engine default service account

Access scopes
 Allow default access
 Allow full access to all Cloud APIs
 Set access for each API

Firewall
Add tags and firewall rules to allow specific network traffic from the Internet
 Allow HTTP traffic
 Allow HTTPS traffic

✓ NETWORKING, DISKS, SECURITY, MANAGEMENT, SOLE-TENANCY

You will be billed for this instance. [Compute Engine pricing](#)

[CREATE](#) [CANCEL](#) [EQUIVALENT COMMAND LINE](#)

Monthly estimate
\$13.23
That's about \$0.02 hourly
Pay for what you use: No upfront costs and per second billing
[DETAILS](#)

- By default, new Google Cloud virtual machines are assigned with ephemeral External IP addresses that change when the instance is stopped or started. If you wish Stack Manager's External IP address to remain unchanged, allocate an External IP address and attach it to the virtual machine.
- Continue with Stack Manager installation, as described in Section 2.4.6, Installing Stack Manager Application.

2.4.5 Creating OpenStack Instance

The following procedure describes how to create a new OpenStack instance for running the Stack Manager application.

To create an OpenStack instance for running Stack Manager application:

1. Open the OpenStack dashboard.
2. On the Instances page, click **Launch Instance**; the Launch Instance wizard starts with the Details page.
3. In the 'Instance Name' field, enter a unique name for the new instance.

Figure 2-16: Launch Instance Wizard - Details Page

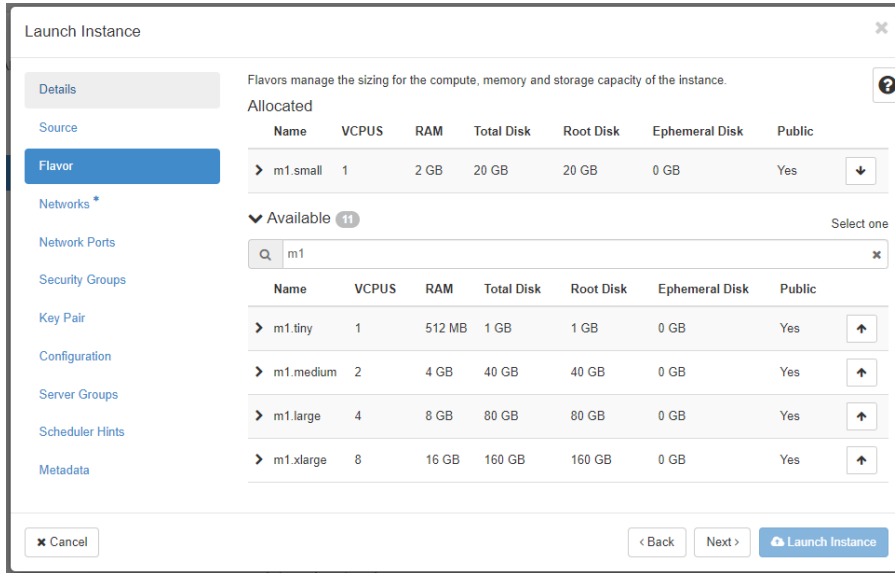
4. Click **Next**; the Source wizard page appears.
5. Select one of the supported Linux distributions, as specified in Section 2.1, Operational Environment, for example, **Debian 11**.

Figure 2-17: Launch Instance Wizard - Source Page

Name	Updated	Size	Type	Visibility
> debian-11	1/25/22 10:02 AM	310.25 MB	qcow2	Public

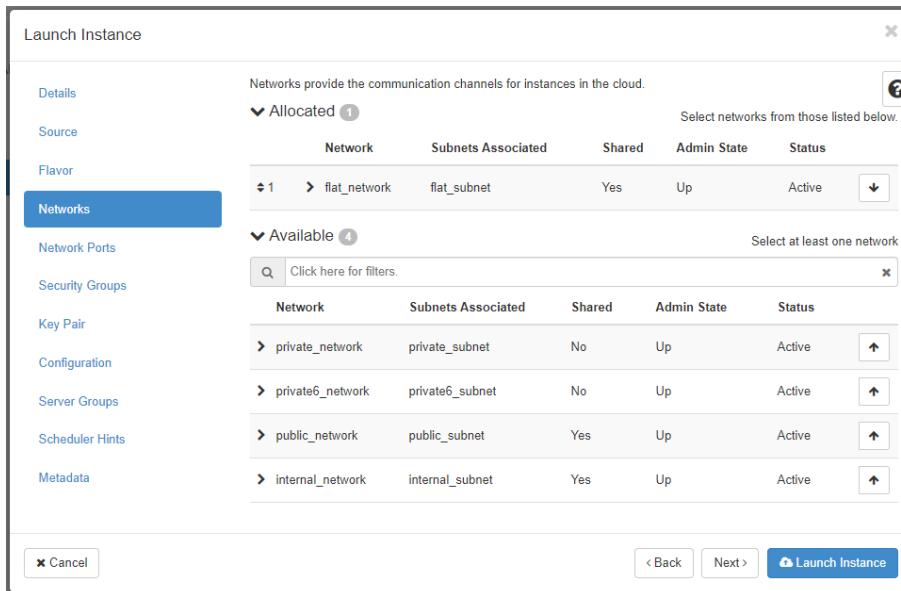
6. Click **Next**; the Flavor wizard page appears.
7. Select the flavor that provides 1 vCPU and 2 GB of RAM.

Figure 2-18: Launch Instance Wizard - Flavor Page



8. Click **Next**; the Networks wizard page appears.
9. Select the "Main Network" that will be used for connecting to the management interface of the deployed Mediant VE/CE stack(s).

Figure 2-19: Launch Instance Wizard - Networks Page



10. Click **Next**; the Network Ports wizard page appears.
11. Click **Next**; the Security Groups wizard page appears.
12. Select a security group that enables the following ports and protocols to communicate with the Stack Manager instance:

Port	Protocol	Purpose
22	TCP	SSH connection to Stack Manager's CLI interface.
80	TCP	HTTP connection to Stack Manager's Web interface.
443	TCP	HTTPS connection to Stack Manager's Web interface.

Figure 2-20: Launch Instance Wizard - Security Groups Page

13. Click **Next**; the Key Pair wizard page appears.
14. Select an existing key pair or create a new one. Make sure that you have private key that matches the selected pair because you will need it to connect the deployed instance through SSH.

Figure 2-21: Launch Instance Wizard - Key Pair Page

15. Click **Launch Instance**.
16. Continue with Stack Manager installation, as described in Section 2.4.6, Installing Stack Manager Application.

2.4.6 Installing Stack Manager Application

The following procedure describes how to install the Stack Manager application after successfully creating the instance / virtual machine.



This step is not needed if you have deployed Stack Manager in Microsoft Azure environment from Azure Marketplace.

To install Stack Manager application:

1. Log in to the launched virtual instance / machine through SSH, using the credentials obtained during the launch.

2. Run the following command to download the latest installation package:

```
$ curl
https://tools.audiocodes.com/install/stack_mgr/stack_mgr.zip -
-output stack_mgr.zip
```

Alternatively, you can download the installation package manually from <https://tools.audiocodes.com/install/index.html> and then transfer it to the virtual instance / machine through an SCP/SFTP client (e.g., WinSCP).

3. Type **unzip** to check if unzip package is installed. If you get the “command not found” response, then the unzip package is missing.

```
$ unzip
-bash: unzip: command not found
```

If the unzip package is missing, install it using the distribution-specific package manager:

- For Debian / Ubuntu, type the following:

```
$ sudo apt update
$ sudo apt install unzip
```

- For RHEL, CentOS, Rocky Linux, Alma Linux, and Amazon Linux, type the following:

```
$ sudo yum install unzip
```

4. Run the following commands to start the installation:

```
$ unzip stack_mgr.zip
$ sudo bash stack_mgr/install.sh
```

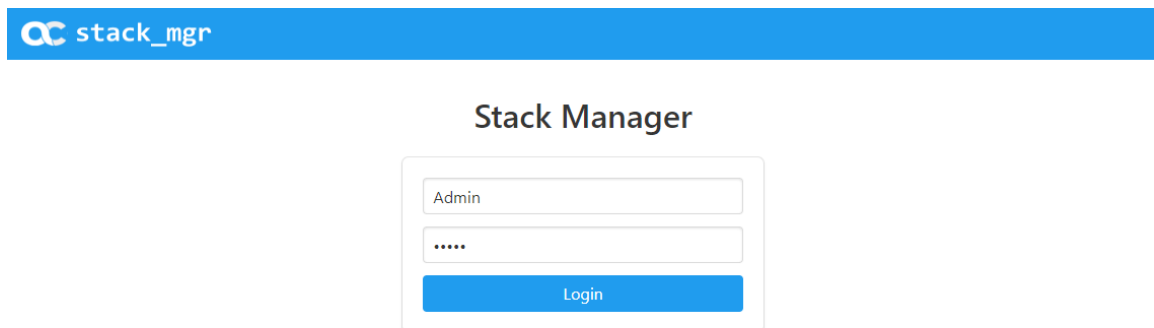
5. Continue with post-installation configuration, as described in Section 2.8, Post-installation Configuration.

2.5 Accessing the Web Interface

Stack Manager's Web interface is accessed by connecting to the virtual machine through HTTP/HTTPS, using one of the supported web browsers:

- Google Chrome
- Firefox
- Microsoft Edge

Figure 2-22: Web Interface of Stack Manager



The default login credentials of the Web Interface are:

- Username: **Admin**
- Password: **Admin**

It's recommended to change the login credentials on first login.

If you deployed Stack Manager in a Microsoft Azure environment from Azure Marketplace, you were prompted to specify login credentials during the deployment. Use these credentials instead of the default ones to log in to the Web Interface.



You can use Azure Entra ID to control access to the Stack Manager's Web interface. See Section 3.6 for details.

To change default Web credentials:

1. Log in to the Web interface.
2. If you are using Stack Manager version 3.5.0 or later:
 - a. Open the **Users** page.
 - b. Click **Modify** corresponding to the Admin user.
 - c. Type the new username and password.
 - d. Click **Modify** to close the dialog.

Figure 2-23: Changing Web Login Credentials

3. If you are using Stack Manager version earlier than 3.5.0:
 - a. Open the **Configuration** page.
 - b. In the 'Admin Username' field, enter the new username.
 - c. In the 'Admin Password' field, enter the new password.
 - d. Click **Update**.

2.6 Accessing the CLI

Stack Manager's CLI interface is accessed by switching to the `stack_mgr` user, using the following command:

```
$ stack_mgr_cli
```

If the above command doesn't function, close the current SSH session and then open a new one. If the problem persists, use the following alternative syntax:

```
$ sudo su - stack_mgr
```

2.7 Upgrading Stack Manager

To upgrade the Stack Manager application to the latest version, log in to the virtual instance (machine) through SSH as a regular user (e.g., `ubuntu`), and then run the following command:

```
$ sudo /opt/stack_mgr/update.sh
```

The command 1) checks if a new Stack Manager application version was published on AudioCodes website, 2) if yes, downloads it, and then 3) updates the current installation. All configuration and created stacks are preserved. The upgrade operation has no effect on Mediant VE/CE stacks service.

The `update.sh` script supports the following optional parameters:

- **--force:** Performs an upgrade even if the current Stack Manager version is later or equal to the one published on AudioCodes website. (This can be useful if the upgrade operation failed and needs to be re-run.)
- **--test:** Checks if a new version is available, but doesn't perform an upgrade.
- **--verify:** Similar to **--test**, but also outputs the change log for the new version.

Alternatively, you can upgrade Stack Manager by installing a new version using the regular installation procedure (see Section 2.4.6, Installing Stack Manager Application for details). All existing configuration and stacks are preserved.

If you are using Stack Manager version 2.5.2 or later, you can also perform an upgrade through the Web interface:

1. Navigate to the “About” screen.
2. If a new version is available, the **Upgrade** button will be displayed.
3. Click the “Upgrade” button and wait for the upgrade to complete.



When upgrading Stack Manager through the regular installation procedure, make sure that you log in as a regular user (e.g., "debian") and that you don't enter Stack Manager's CLI (via the "stack_mgr_cli" command).

2.8 Post-installation Configuration

The following procedures describe post-installation configuration that ensures that Stack Manager is able to properly access cloud / virtual infrastructure APIs.

The instructions depend on the cloud / virtual environment.

After performing the configuration, verify that Stack Manager is able to operate normally, as described in Section 2.8.5, Verifying Configuration.

For production environments, it's also recommended to configure Stack Manager to store its run-time data on cloud storage services, as described in Section 2.9, Runtime Data.



The instructions described in this section use the Web interface to configure Stack Manager. The same tasks can be performed through CLI, using the `configure` command, as described in Section 3.3, Managing Users.

2.8.1 Post-installation Configuration on Amazon Web Services (AWS)

The following procedure describes post-installation configuration of the Stack Manager application in the Amazon Web Services (AWS) environment, which consists of the following step:

- Enabling Stack Manager virtual machine access to AWS APIs

2.8.1.1 Enabling Access to AWS API via IAM Role (Recommended Method)

Before using Stack Manager, you need to ensure that it has access to the AWS API. The recommended method for achieving this is to create an IAM role, as described in Section 2.3.1.1, IAM Role for Stack Manager, and then to attach it to the Stack Manager's virtual instance during its creation, as described in Section 2.4.2, Creating Amazon Web Services (AWS) Instance.

2.8.1.2 Enabling Access to AWS API via AWS Access Key (Alternative Method)

This section describes an alternative method for enabling Stack Manager access to AWS APIs. For typical deployments, please use the recommended method instead, as described in Section 2.8.1.1, Enabling Access to AWS API via IAM Role (Recommended Method).

To configure Stack Manager access to AWS API using access key:

1. Obtain the AWS access key with permissions listed in Section 2.3.1.1, IAM Role for Stack Manager. For more information on how to do this, refer to [AWS documentation](#).
2. Log in to the Stack Manager Web interface.
3. Open the Configuration page.
4. Enter the access key values in the 'AWS Access Key' and 'AWS Secret Key' fields.
5. Click **Update**.

2.8.2 Post-Installation Configuration on Microsoft Azure

The following procedure describes post-installation configuration of the Stack Manager application in Microsoft Azure environment, which includes the following steps:

1. Configure the Azure Subscription ID.
2. Enable the Stack Manager virtual machine access to Azure APIs.
3. (Optional) Enable login via Azure Entra ID.

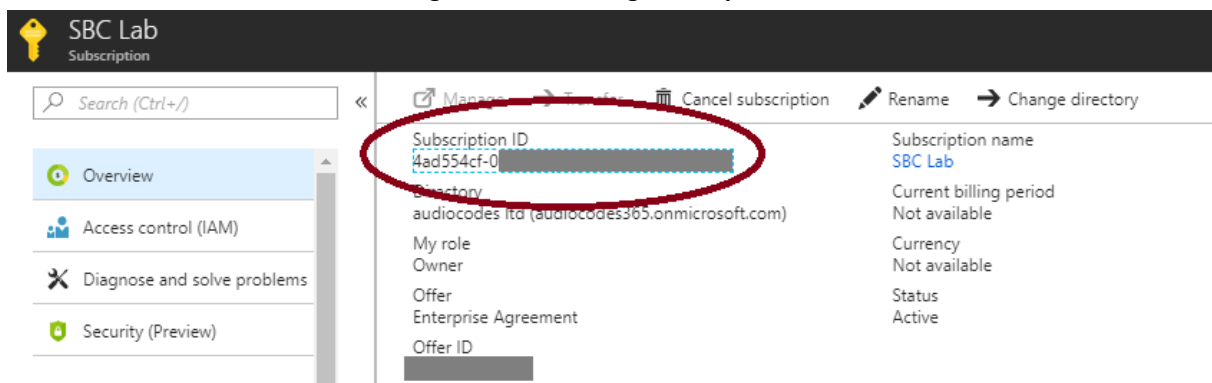
2.8.2.1 Configuring the Azure Subscription ID

After installing Stack Manager, you need to configure the Subscription ID where it will operate.

To configure Azure Subscription ID:

1. Open the Azure portal at <https://portal.azure.com/>.
2. Navigate to Subscriptions (**All services** > **Subscriptions**).
3. Locate your Azure Subscription ID.

Figure 2-24: Locating Subscription ID



4. Log in to the Stack Manager Web interface.
5. Open the Configuration page.
6. Enter the Azure subscription ID in the 'Azure Subscription ID' field.

Figure 2-25: Configuring Azure Subscription ID

The screenshot shows the 'Configuration' page in Stack Manager. The page has a blue header with the 'stack_mgr' logo and navigation links for 'Stacks', 'Configuration', 'Logs', and 'About'. A 'Logout' button is in the top right. The main content area is titled 'Configuration' and contains five panels:

- General:** Fields for 'Name Prefix', 'Admin Username' (pre-filled with 'Admin'), and 'Admin Password'.
- Amazon Web Services:** Fields for 'Access Key', 'Secret Key', 'S3 Bucket', and 'S3 Prefix'.
- Openstack:** Fields for 'Cloud Name' and 'Container'.
- Microsoft Azure:** Fields for 'Tenant ID', 'Client ID', 'Secret', 'Subscription ID' (pre-filled with '4ad554cf-8b4e-4a65-8a14-2b4951a3d1d3'), 'Blob Account Name', 'Blob Account Key', and 'Blob Container'.
- Google:** Fields for 'Project', 'Credentials', 'Storage Bucket', and 'Storage Prefix'.

At the bottom of the configuration area are two buttons: 'Update' (with a checkmark icon) and 'Verify' (with a question mark icon).

7. Click **Update**.

2.8.2.2 Enabling Access to Azure APIs via Managed Service Identity (Recommended Method)

Before using Stack Manager, you need to ensure that it has access to Azure APIs. This section describes the recommended method for achieving this through the Managed Service Identity. The method consist of two steps:

1. Enabling Managed Service Identity for the Stack Manager virtual machine.
2. Assigning a proper IAM role to the Stack Manager virtual machine.

An alternative method is to use the service principal, as described in Section 2.8.2.3, Enabling Access to Azure APIs via Service Principal (Alternative Method).

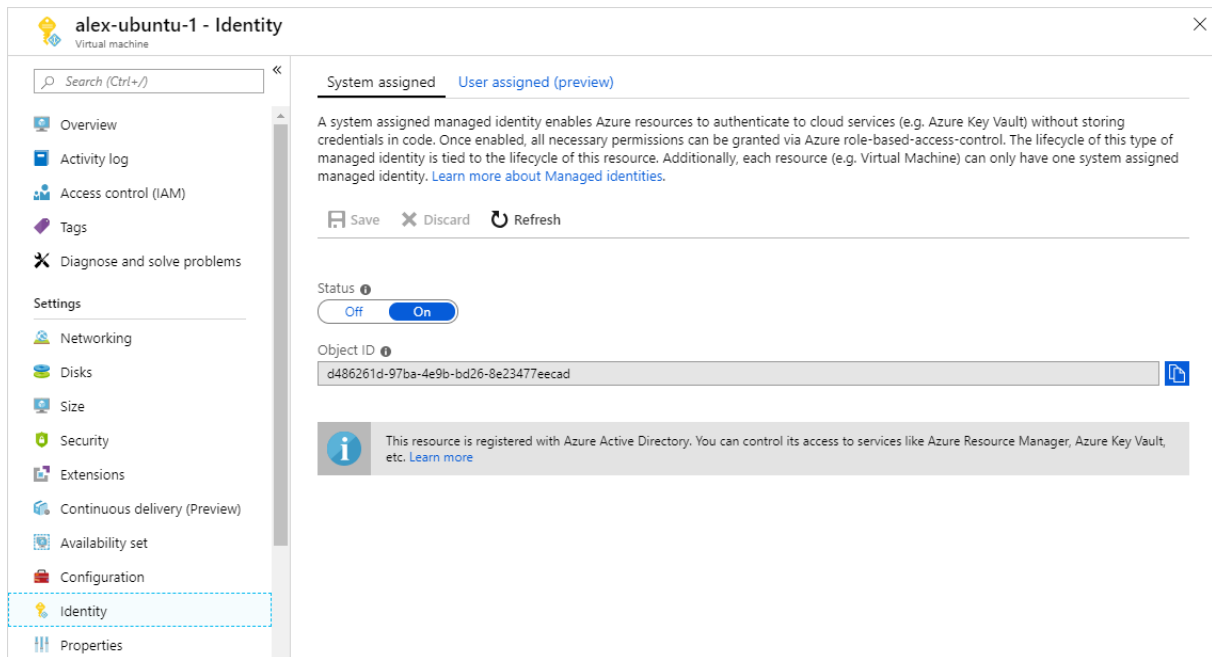
Managed Service Identity (MSI) enables the assignment of access control (IAM) roles to a specific Azure virtual machine deployed in Azure.

To enable Managed Service Identity:

1. Open the Azure portal at <http://portal.azure.com>.
2. Navigate to the Virtual Machines page.
3. Select the Stack Manager virtual machine.

- In the Navigation menu, click **Identity**, and then enable Managed Service Identity.

Figure 2-26: Configuring Virtual Machine's Managed Service Identity



Once you have performed the above procedure, you should grant the Stack Manager virtual machine permissions to access all needed Azure APIs for successful stack deployment and management. There are several ways to achieve this:

- Option 1 (recommended): Assign Stack Manager with the "Contributor" role at the Subscription level.
- Option 2: Assign Stack Manager with custom IAM roles at Subscription, Network and Resource Group levels.

2.8.2.2.1 Option 1: "Contributor" Role at Subscription Level

This method provides Stack Manager with complete access to Subscription resources, including the ability to create new Resource Groups. This method is recommended for most users, as it's simple to provision and doesn't impose any restrictions on Stack Manager functionality.

To assign Stack Manager with "Contributor" role at Subscription level:

- Open the Azure portal at <http://portal.azure.com>.
- Navigate to the Subscriptions page.
- Select your subscription.
- In the Navigation menu, click **Access Control (IAM)**, and then click **Add a role assignment**:
 - From the 'Role' drop-down list, select **Contributor**.
 - From the 'Assign access to' drop-down list, select **Virtual Machine**.

- c. From the 'Select' drop-down list, select the name of Stack Manager's virtual machine.
- d. Click **Save**.

Figure 2-27: Adding Role Assignment

The screenshot shows a dialog box titled "Add role assignment". It contains the following fields and elements:

- Role:** A dropdown menu set to "Contributor".
- Assign access to:** A dropdown menu set to "Virtual Machine".
- Subscription:** A dropdown menu set to "SBC Lab".
- Select:** A section with a "Search by name" input field and a list of virtual machines. One VM, "alex-ubuntu-1", is visible in the list.
- Selected members:** A section showing "alex-stack-mgr-2" as the selected member, with a "Remove" link next to it.
- Buttons:** "Save" and "Discard" buttons at the bottom.

2.8.2.2.2 Option 2: Custom IAM Roles at Subscription, Network and Resource Group Levels

This method limits Stack Manager administrative access to the specific pre-defined Resource Group(s). It's more complicated to provision and slightly complicates stack creation. Therefore, this method is recommended for advanced users who want to minimize IAM permissions granted to the Stack Manager.

With this method, Stack Manager is assigned the following IAM roles:

Scope	IAM Role
Subscription	Custom IAM role that includes read-only access for specific resources only (e.g., virtual networks and subnets). This is needed for displaying "Create new stack" Web UI dialog and validating stack configuration during create, modify, update, and heal operations.
Virtual Network	Custom IAM role that grants Stack Manager the ability to deploy new virtual machines into the specific Virtual Network(s). The role is assigned only for specific Virtual Networks where new stacks will be deployed.
Resource Group	Custom IAM role that grants Stack Manager the ability to create, modify and delete stack resources (e.g., virtual machines, network interfaces, load balancers). The role is assigned only for specific Resource Group(s) that must be pre-created prior to stack deployment.



When using this method, an empty Resource Group must be manually created prior to stack deployment. The name of this Resource Group must be specified during new stack creation through the Advanced Config parameter **resource_group**.

To assign Stack Manager with custom IAM roles at Subscription, Network and Resource Group levels:

1. Create the following three custom IAM roles:

- **Custom IAM Role 'Stack Manager Subscription Role':**

```
{
  "properties": {
    "roleName": "Stack Manager Subscription Role",
    "description": "Subscription role for AudioCodes Stack Manager.",
    "assignableScopes": [
      "/subscriptions/{subscriptionId}"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Network/publicIPAddresses/read",
          "Microsoft.Compute/images/read",
          "Microsoft.Compute/skus/read",
          "Microsoft.Compute/virtualMachines/vmSizes/read",
          "Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/read",
          "Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/write"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
      }
    ]
  }
}
```

- **Custom IAM Role 'Stack Manager Network Role':**

```
{
  "properties": {
    "roleName": "Stack Manager Network Role",
    "description": "Network role for AudioCodes Stack Manager.",
    "assignableScopes": [
      "/subscriptions/{subscriptionId}"
    ],
    "permissions": [
      {
```

```

        "actions": [
            "Microsoft.Network/virtualNetworks/subnets/join/action"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
    }
}
}
}

```

- **Custom IAM Role 'Stack Manager Resource Group Role':**

```

{
  "properties": {
    "roleName": "Stack Manager Resource Group Role",
    "description": "",
    "assignableScopes": [
      "/subscriptions/{subscriptionId}/resourcegroups/{rgName}"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.Compute/availabilitySets/*",
          "Microsoft.Compute/proximityPlacementGroups/*",
          "Microsoft.Compute/locations/*",
          "Microsoft.Compute/virtualMachines/*",
          "Microsoft.Compute/disks/write",
          "Microsoft.Compute/disks/read",
          "Microsoft.Compute/disks/delete",
          "Microsoft.Network/networkInterfaces/*",
          "Microsoft.Network/networkSecurityGroups/*",
          "Microsoft.Network/publicIPAddresses/*",
          "Microsoft.Resources/deployments/*",
          "Microsoft.Storage/storageAccounts/*",
          "Microsoft.Network/loadBalancers/*",
          "Microsoft.Network/loadBalancers/backendAddressPools/*",
          "Microsoft.Network/loadBalancers/probes/*",
          "Microsoft.Network/loadBalancers/outboundRules/*",
          "Microsoft.Network/loadBalancers/loadBalancingRules/*",
          "Microsoft.Network/loadBalancers/frontendIPConfigurations/*",
          "Microsoft.Resources/subscriptions/resourceGroups/read",
          "Microsoft.ManagedIdentity/userAssignedIdentities/*"
        ]
      }
    ]
  }
}

```

```
    ],  
    "notActions": [],  
    "dataActions": [],  
    "notDataActions": []  
  }  
]  
}  
}
```

Refer to [Azure documentation](#) for detailed instructions on how to create custom IAM roles.

2. Open the Azure portal at <http://portal.azure.com>.
3. Navigate to the Subscriptions page.
4. Select your subscription.
5. In the Navigation menu, click **Access Control (IAM)**, and then click **Add a role assignment**:
 - a. From the 'Role' drop-down list, select **Stack Manager Subscription Role**.
 - b. From the 'Assign access to' drop-down list, select **Virtual Machine**.
 - c. From the 'Select' drop-down list, select the name of Stack Manager's virtual machine.
 - d. Click **Save**.
6. Navigate to the Virtual Networks page.
7. Select the network where new stacks will be deployed.
8. In the Navigation menu, click **Access Control (IAM)**, and then click **Add a role assignment**:
 - a. From the 'Role' drop-down list, select **Stack Manager Network Role**.
 - b. From the 'Assign access to' drop-down list, select **Virtual Machine**.
 - c. From the 'Select' drop-down list, select the name of Stack Manager's virtual machine.
 - d. Click **Save**.
9. Navigate to the Resource Groups page.
10. Click **Add** to create a new Resource Group(s) where new stacks will be deployed. Each stack will require a dedicated Resource Group that must be empty prior to stack creation.
 - a. Enter the Resource Group name.
 - b. From the 'Region' drop-down list, select the region where the new stack will be deployed.
 - c. Click **Create**.
11. Select the created Resource Group(s).
12. In the Navigation menu, click **Access Control (IAM)**, and then click **Add a role assignment**:
 - a. From the 'Role' drop-down list, select **Stack Manager Resource Group Role**.
 - b. From the 'Assign access to' drop-down list, select **Virtual Machine**.
 - c. From the 'Select' drop-down list, select the name of Stack Manager's virtual machine.
 - d. Click **Save**.
13. Restart the Stack Manager virtual machine to apply the new IAM credentials.

2.8.2.2.1 Advanced Restriction of Custom IAM Roles

Custom IAM roles, described in the previous section, can further be restricted if you choose to pre-create some Azure resources (e.g., public IP addresses) and/or are willing to deploy the new stack via the CLI interface.

The following permissions can be dropped from the Stack Manager Subscription Role:

Permission	What happens when it's dropped
Microsoft.Network/ virtualNetworks/read	<p>You will be unable to create the new stack through the Web interface. Use the CLI or REST interface to create the new stack. After initial creation, further stack management can be performed through all management interfaces, including Web interface.</p> <p>Stack Manager will not be able to validate the Virtual Network name prior to stack deployment. If the wrong name is provided, stack deployment will fail.</p>
Microsoft.Network/ virtualNetworks/subnets/read	<p>You will be unable to create the new stack through the Web interface. Use the CLI or REST interfaces to create the new stack. After initial creation, further stack management can be performed through all management interfaces, including Web interface.</p> <p>Stack Manager will not be able to validate Subnet names prior to stack deployment. If wrong names are provided, stack deployment will fail.</p> <p>You must specify the CIDR for each subnet through the Advanced Config parameters: cluster_subnet_cidr, main_subnet_cidr, additional1_subnet_cidr, additional2_subnet_cidr. Otherwise, Stack Manager will not be able to properly configure network interfaces for deployed stack components.</p>
Microsoft.Network/ publicIPAddresses/read	<p>Stack Manager will not be able to validate predefined Public IP addresses provided via public_ip_* Advanced Config parameters. If wrong names are provided, stack deployment will fail.</p>
Microsoft.Compute/ images/read	<p>Stack Manager will not be able to validate custom VM images provided via sc_image_id / mc_image_id Advanced Config parameters. If wrong names are provided, stack deployment will fail.</p>
Microsoft.Compute/ skus/read	<p>Stack Manager will not be able to validate instance types (VM sizes) provided via sc_instance_type / mc_instance_type Advanced Config parameters. If wrong names are provided, stack deployment will fail.</p>
Microsoft.MarketplaceOrdering/ offertypes/publishers/offers/ plans/agreements/read Microsoft.MarketplaceOrdering/ offertypes/publishers/offers/ plans/agreements/write	<p>Stack Manager will not be able to automatically accept publisher agreement for Mediant VE/CE Marketplace offer. You need to manually accept the agreement prior to new stack deployment through the following CLI command:</p> <pre>az vm image terms accept \ --publisher audiocodes \ --offer mediantsessionbordercontroller \ --sku mediantvesbcazure</pre> <p>If you are deploying SBC version based on CentOS 6, use --sku mediantvirtualsbcazure in the above command.</p>

Permission	What happens when it's dropped
	If agreement is not accepted stack deployment will fail.

The following permissions can be dropped from the Stack Manager Resource Group Role:

Permission	What happens when it's dropped
Microsoft.Network/ networkSecurityGroups/*	You must precreate network security groups and provide them via cluster_nsg_id / oam_nsg_id / signaling_nsg_id / media_nsg_id Advanced Config parameters during the new stack creation. Stack Manager VM must be granted with Microsoft.Network/networkSecurityGroups/actions/join permission in the Resource Group where network security groups reside.
Microsoft.Network/ publicIPAddresses/*	You must precreate public IP addresses and provide them via public_ip_* Advanced Config parameters during the new stack creation. Stack Manager VM must be granted with Microsoft.Network/publicIPAddresses/read and Microsoft.Network/publicIPAddresses/actions/join permissions in the Resource Group where public IP addresses reside.
Microsoft.Storage/ storageAccounts/*	You must precreate diagnostics Storage Account and provide it via diag_account Advanced Config parameters during the new stack creation. Stack Manager VM must be granted with Microsoft.Storage/storageAccounts/read permission in the Resource Group where Storage Account resides.

2.8.2.3 Enabling Access to Azure APIs via Service Principal (Alternative Method)

This section describes an alternative method for enabling Stack Manager access to Azure APIs. For typical deployments, please use the recommended method instead, as described in Section 2.8.2.2, Enabling Access to Azure APIs via Managed Service Identity (Recommended Method).

To configure Stack Manager access to Azure API using Service Principal:

1. Create an Azure Service Principal, as described in the [Azure documentation](#). Assign an appropriate IAM role(s) to the created Azure Service Principal, as described in the previous section.
2. Log in to the Stack Manager Web interface.
3. Open the Configuration page.
4. Enter the values in the 'Azure Tenant ID', 'Azure Client ID' and 'Azure Secret' fields.
5. Click **Update**.

2.8.3 Post-Installation Configuration on Google Cloud

The following procedure describes post-installation configuration of the Stack Manager application in Google Cloud environment, which includes the following steps:

1. Configuring Google Project ID.
2. Enabling Google Cloud APIs in the Project.
3. Enabling Stack Manager virtual machine access to Google Cloud APIs.

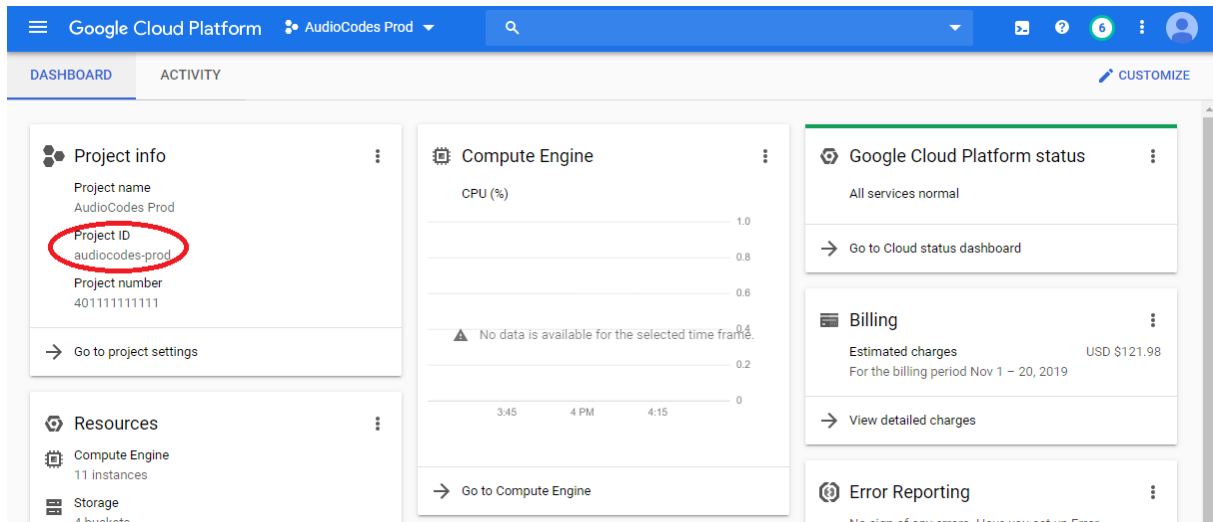
2.8.3.1 Configuring Google Project ID

After installing Stack Manager, you need to configure the Project ID where it will operate.

To configure Google Project ID:

1. In Google Cloud Platform Console, go to the **Home > Dashboard** (<https://console.cloud.google.com/home/dashboard>), and then determine your project ID.

Figure 2-28: Determining Google Project ID



2. Log in to the Stack Manager Web interface.
3. Open the Configuration page.
4. In the 'Google Project' field, enter the Project ID.
5. Click **Update**.

2.8.3.2 Enabling APIs in Project

The following Google Cloud APIs must be enabled in the Project for normal Stack Manager operation:

- Compute Engine API
- Cloud Deployment Manager V2 API
- Cloud Resource Manager API

To enable APIs in the project:

1. In the Google Cloud Platform Console, go to the **API & Services > Dashboard** page (<https://console.cloud.google.com/apis/dashboard>).
2. Click **Enable APIs And Services**.
3. Type the API name, and then select it from the list.
4. Click **Enable** to enable the API.
5. Repeat the above steps for all APIs required by the Stack Manager.

2.8.3.3 Creating a Service Account

Service Accounts are used to manage application permissions.

To create a Service Account:

1. In the Google Cloud Platform Console, go to the **IAM & admin > Service Accounts** page (<https://console.cloud.google.com/iam-admin/serviceaccounts>).
2. Click **Create service account**.
3. Enter the service account name, for example, "stack-mgr", and provide a description.
4. Click **Create** to create the account.
5. On the **Service account permissions (optional)** page displayed immediately afterwards, assign the following IAM roles to the service account, and then click **Continue**.
 - a. Compute Engine > Compute Admin.
 - b. Deployment Manager > Deployment Manager Editor.
6. On the **Grant users access to this service account (optional)** page displayed immediately afterwards, click **Done**.
7. Go to the **IAM & admin > IAM** page (<https://console.cloud.google.com/iam-admin/iam>).
8. Verify that the service account has been successfully created and is assigned with Compute Admin and Deployment Manager Editor roles.

2.8.3.4 Enabling Access to Google Cloud APIs via Service Account (Recommended Method)

Before using Stack Manager, you need to ensure that it has access to Google Cloud API. This section describes the recommended method for achieving this through the Service Account assigned to the Stack Manager virtual machine.

An alternative method is to use the configuration file, as described in Section 2.8.3.5, Enabling Access to Google Cloud APIs via Configuration File (Alternative Method).

To assign Service Account to Stack Manager virtual machine:

1. In the Google Cloud Platform Console, go to the **Compute Engine > VM Instances** page (<https://console.cloud.google.com/compute/instances>).
2. Click the Stack Manager VM.
3. On the VM instance details page, click **Edit**.
4. For **Service account**, select the Service Account that you created in Section 2.8.3.3, Creating a Service Account.
5. Click **Save**.

2.8.3.5 Enabling Access to Google Cloud APIs via Configuration File (Alternative Method)

This section describes an alternative method for enabling Stack Manager access to Google Cloud APIs. For typical deployments, please use the recommended method instead, as described in Section 2.8.3.4, Enabling Access to Google Cloud APIs via Service Account (Recommended Method).

To enable access to Google Cloud APIs via configuration file:

1. In the Google Cloud Platform Console, go to the **IAM & admin > Service Accounts** page (<https://console.cloud.google.com/iam-admin/serviceaccounts>).
2. Click the Service Account that you created in Section 2.8.3.3, Creating a Service Account.
3. Click **Edit**.
4. Click **Create Key**.
5. Choose the JSON key type, and then click **Create**.
6. The credentials file, which contains the generated key, is downloaded and saved to your computer. Move the file to a permanent location and write down its complete name and path.
7. Log in to the Stack Manager Web interface.
8. Open the Configuration page.
9. In the 'Google Credentials' field, enter the complete path to the credentials file.
10. Click **Update**.

2.8.4 Post-installation Configuration on OpenStack

The following procedure describes post-installation configuration of the Stack Manager application in the OpenStack environment.

To perform post-installation configuration of Stack Manager in OpenStack environment:

1. Obtain credentials for application access to your OpenStack installation.
2. Create the configuration file **clouds.yaml**, which will be used by Stack Manager to access OpenStack APIs. Below shows an example OpenStack configuration file:

```
clouds:
  openstack-se2:
    region_name: RegionOne
    auth:
      auth_url: http://10.4.220.50:5000/v3
      username: admin
      password: 123456
      project_name: admin
      project_domain_name: Default
      user_domain_name: Default
```

Change the configuration parameters to match your OpenStack installation. Refer to the **openstacksdk** documentation at <http://docs.openstack.org/openstacksdk> for more information.

3. Place the file in one of the following locations:
 - /var/stack_mgr/.config/openstack
 - /etc/openstackMake sure that the file is readable by user **stack_mgr**.
4. Log in to the Stack Manager Web interface.
5. Open the Configuration page.
6. In the 'OpenStack Cloud Name' field, enter the value ("openstack-se2" in the example above).
7. Click **Update**.

2.8.5 Verifying Configuration

After completing post-installation configuration, perform the following steps to verify that Stack Manager can operate normally.

To verify Stack Manager configuration:

1. Log in to the Stack Manager Web interface.
2. Open the Configuration page.
3. Click **Verify**.
4. Wait until the operation completes, and then check its output.

Figure 2-29: Verifying Stack Manager Configuration

The screenshot displays the 'Configuration' page in the Stack Manager web interface. The page has a blue header with the 'stack_mgr' logo and navigation links for 'Stacks', 'Configuration', 'Logs', and 'About'. A 'Logout' button is in the top right corner. The main content area is titled 'Configuration' and contains several panels for different cloud providers:

- General:** Name Prefix (alex-), Admin Username (Admin), Admin Password.
- Amazon Web Services:** Access Key, Secret Key, S3 Bucket, S3 Prefix.
- Openstack:** Cloud Name, Container.
- Microsoft Azure:** Tenant ID, Client ID, Secret, Subscription ID (4ad554cf-104e-4a65-8a14-2b8951a3d1d3), Blob Account Name (sbc1), Blob Account Key (MqknD2G0PYUhh9PfygCK3C2hgqHaaemr5R8D95awtUJhsg), Blob Container (stackmgr).
- Google:** Project, Credentials, Storage Bucket, Storage Prefix.

At the bottom of the configuration panels are two buttons: 'Update' (with a checkmark icon) and 'Verify' (with a question mark icon). Below these buttons is a light green notification box with the following text:

```
Verifying configuration
Verify access to Azure API... success
Done
```

2.9 Runtime Data

Stack Manager uses *stack descriptors* to keep information about created stacks, including their configuration and references to all corresponding resources. By default, Stack Manager stores this information on the local file system in the `/opt/stack_mgr/data` directory.

However, you can configure Stack Manager to store the *stack descriptors* in the cloud storage services, namely:

- AWS Simple Cloud Storage Service (S3)
- Microsoft Azure Storage Service
- Google Cloud Storage Service
- OpenStack Object Storage Service (swift)

Doing so significantly improves runtime data availability and provides service continuity if the Stack Manager instance must be rebuilt.



Stack descriptors are for internal Stack Manager use and should **not** be manipulated by the user.

2.9.1 Storing Runtime Data on AWS S3

The procedure below describes how to configure Stack Manager to store its runtime data on AWS S3.

To configure Stack Manager to store runtime data on AWS S3:

1. Open the AWS S3 Console at <http://console.aws.amazon.com/s3>.
2. Create a new S3 bucket in the same region where the Stack Manager instance is deployed. Enter the bucket name (e.g., "stack-mgr").

Figure 2-30: Create Bucket

The screenshot shows the 'Create bucket' dialog in the AWS S3 console. The dialog is titled 'Create bucket' and has a close button (X) in the top right. It features a progress bar with four steps: 1 Name and region (active), 2 Set properties, 3 Set permissions, and 4 Review. The 'Name and region' section includes a 'Bucket name' field with the value 'stack-mgr', a 'Region' dropdown menu set to 'EU (Frankfurt)', and a 'Copy settings from an existing bucket' section with a dropdown menu showing 'Select bucket (optional)' and '33 Buckets'. At the bottom, there are three buttons: 'Create', 'Cancel', and 'Next'.

3. Create a new IAM policy that allows the Stack Manager instance to access data in the created S3 bucket. In the 'Bucket name' field, replace **stack-mgr** with the actual name of the bucket that you created.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::stack-mgr"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3>DeleteObject"
      ],
      "Resource": "arn:aws:s3:::stack-mgr/*"
    }
  ]
}
```

4. Attach the created IAM policy to the Stack Manager instance (*in addition* to the policy created in Section 2.3.1.1, IAM Role for Stack Manager).
5. Log in to the Stack Manager Web interface.
6. Open the Configuration page.
7. In the 'AWS S3 Bucket' field, enter the value ("stack-mgr" in the example above).
8. If you want Stack Manager runtime data to be stored in some folder(s), configure the 'AWS S3 Prefix' field to some value that ends with "/" (e.g., "stack-mgr/").
9. Click **Update**.
10. Click **Verify** to verify configuration.

2.9.2 Storing Runtime Data on Azure Storage Service

The procedure below describes how to configure Stack Manager to store its runtime data on Microsoft Azure Storage Service.

To configure Stack Manager to store runtime data on Azure Storage Service:

1. Open the Azure portal at <https://portal.azure.com/>.
2. Navigate to the Storage Accounts page (**All services > Storage Accounts**).
3. Create a new Storage Account in the same location where the Stack Manager virtual machine is deployed.
4. Locate the access key for the Storage Account under the **Access keys** tab.
5. Go to the **Blobs service**, and then create a new container.
6. Log in to the Stack Manager Web interface.
7. Open the Configuration page.
8. In the 'Azure Blob Account Name', 'Azure Blob Account Key', and 'Azure Blob Container' fields, enter the values.
9. Click **Update**.
10. Click **Verify** to verify configuration.



Instead of using the Access Key as described above, Stack Manager can be configured to access Azure Storage Service using a shared access signature (SAS) token. For this you need to use the 'Azure Blob SAS token' configuration parameter.

2.9.3 Storing Runtime Data on Google Cloud Storage Service

The procedure below describes how to configure Stack Manager to store its runtime data on Google Cloud Storage Service.

To configure Stack Manager to store runtime data on Google Cloud Storage Service:

1. In the Google Cloud Platform Console, go to the **Storage > Browser** page (<https://console.cloud.google.com/storage/browser>).
2. Create a bucket where Stack Manager runtime data will be stored.
3. Create folder(s) inside the bucket, if needed.
4. Go to the **IAM & admin > IAM** page (<https://console.cloud.google.com/iam-admin/iam>).
5. Assign the following IAM role to the Stack Manager service account: **Storage > Storage Admin**.
6. Log in to the Stack Manager Web interface.
7. Open the Configuration page.
8. In the 'Google Storage Bucket' field, enter the value.
9. If you want Stack Manager runtime data to be stored in some folder(s), configure the 'Google Storage Prefix' field to some value that ends with "/" (e.g., "stack-mgr/").
10. Click **Update**.
11. Click **Verify** to verify configuration.

2.9.4 Storing Runtime Data on OpenStack Object Storage Service

The procedure below describes how to configure Stack Manager to store its runtime data on OpenStack Object Storage Service (swift).

To configure Stack Manager to store runtime data on OpenStack Object Storage Service (swift):

1. Open the OpenStack dashboard.
2. Navigate to **Object Store > Containers** page.
3. Create a new Object Storage (swift) container.
4. Log in to the Stack Manager Web interface.
5. Open the Configuration page.
6. In the 'Openstack Container' field, enter the value.
7. Click **Update**.
8. Click **Verify** to verify configuration.

2.9.5 Migrating Runtime Data from Local Disk to Storage Service

If you started working with Stack Manager while it was configured to store run-time data on local disk and later decided to migrate to the cloud-specific storage service, use the following procedure to migrate the data:

1. Download all .json files from the `/opt/stack_mgr/data` folder to your computer.
2. Remove the .json extension from all the downloaded files.
3. Upload all the files to the proper container / folder on the storage service.

2.10 Resource Naming

By default, resources created by Stack Manager (e.g., virtual machines) use the following naming convention:

```
<stack name>-<resource name>
```

For example, for stack 'stack1', the corresponding resources are named "stack1-sc-1", "stack1-mc-1" and so on.

It's possible to define additional prefixes that will be added to created resources. The prefix would typically end with a dash "-". For example, if you configure it as "lab1-", the corresponding resources are named "lab1-stack1-sc-1", and so on.

To configure a name prefix:

1. Log in to the Stack Manager Web interface.
2. Open the Configuration page.
3. In the 'Name Prefix' field, enter the value (e.g., "lab1-").
4. Click **Update**.



The 'Name Prefix' field should be configured *prior* to any Mediant VE/CE stack creation. **Don't** change it if some stacks already exist.

2.11 Backup and Restore

To create a backup of Stack Manager installation, use the `/opt/stack_mgr/backup.sh` script as described below.

To create a backup of Stack Manager installation:

1. Connect to the Stack Manger virtual machine via an SSH client (e.g., PuTTY).
2. Log in as a regular user (e.g., "debian").
3. Run the following command:

```
sudo /opt/stack_mgr/backup.sh <filename>
```

Specify the name of the backup file instead of <filename> (e.g., **backup.tgz**).
4. Download the created backup file via an SCP/SFTP client (e.g., WinSCP).

To restore a backup of Stack Manager installation, use the `/opt/stack_mgr/restore.sh` script as described below.

To restore a backup of Stack Manager installation:

1. Upload the backup file to the Stack Manager virtual machine via an SCP/SFTP client (e.g., WinSCP).
2. Connect to the Stack Manger virtual machine via an SSH client (e.g., PuTTY).
3. Log in as a regular user (e.g., "debian").
4. Run the following command:

```
sudo /opt/stack_mgr/restore.sh <filename>
```

Specify the name of the backup file instead of <filename> (e.g., **backup.tgz**).

2.12 Migrating to a New Virtual Machine

If you need to replace the operating system on the Stack Manager virtual machine, migrate the existing Stack Manager installation as follows:

To migrate Stack Manager to a new virtual machine:

1. Create a backup of the current Stack Manager installation, as described in Section 2.11, Backup and Restore, and then download it to your PC.
2. Shut down the current virtual machine.
3. Create a new virtual machine and install Stack Manager on it, as described in Section 2.4, Installation.
4. Upload the backup file to the new virtual machine and restore it, as described in Section 2.11, Backup and Restore.
5. Verify that the new Stack Manager instance works correctly.
6. Terminate the old virtual machine.



Never run two copies of Stack Manager software that manage the same stacks on two different virtual machines. This might result in corrupted stack configuration.

2.13 Providing Debug File for Troubleshooting

If you experience any issues with the Stack Manager application and need to open a service request at AudioCodes [Services Portal](#), generate a debug file as described below and then attach it to your service request.

To generate Debug File:

1. Connect to the Stack Manger virtual machine through an SSH client (e.g., PuTTY).
2. Log in as a regular user (e.g., "debian").
3. Run the following command:

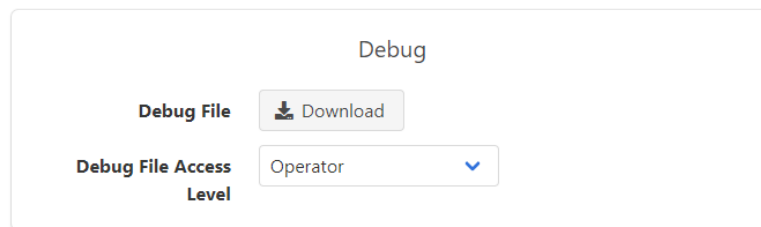
```
sudo /opt/stack_mgr/debug.sh
```

A debug file is generated with the name **stack_mgr_debug.tgz**.

Once you have generated the file, download it using an SCP / SFTP client (e.g., WinSCP) and then attach it to your service request.

If you are using Stack Manager version 3.2.4 or later, you can also download the Debug File through the Web interface's Configuration screen.

Figure 2-31: Downloading Debug File via Web Interface



If you are using Stack Manager version 3.4.7 or later, you can control minimum access level (e.g., Operator) that's allowed to download the Debug File through the Web interface. This is done using the 'Debug File Access Level' parameter in the Configuration screen. The parameter can be changed by the Security Administrator user only.

If you are using Stack Manager version 3.5.0 or later, you can control whether OS information is included in the Debug File that's downloaded through the Web interface. By default, this information is included for efficient troubleshooting. Use the following CLI command to change this behavior:

```
stack_mgr configure --debug-file-include-os-data disable
```

The above configuration applies to debug files downloaded through the Web interface only. OS information is always included in debug files created through the CLI.

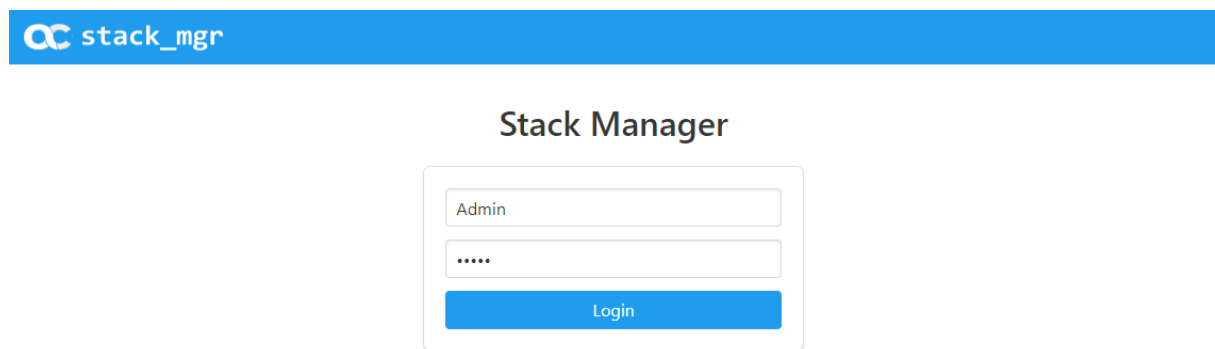
3 Web Interface

3.1 Accessing the Web Interface

Stack Manager's Web interface is accessed by connecting to the virtual machine through HTTP/HTTPS, using one of the supported web browsers:

- Google Chrome
- Firefox
- Microsoft Edge

Figure 3-1: Accessing Web Interface



If you installed Stack Manager from the .zip file, as described in Section 2.4.6, Installing Stack Manager Application, the default login credentials of the Web interface are:

- Username: **Admin**
- Password: **Admin**

It's recommended to change the default login credentials on first login, as described in Section 2.5, Accessing the Web Interface.

If you deployed Stack Manager in a Microsoft Azure environment from Azure Marketplace, you were prompted to specify login credentials during the deployment. Use these credentials instead of the default ones to log in to the Web Interface.



You can use Azure Entra ID to control access to the Stack Manager's Web interface. For more information, see Section 3.6, Login via Azure Entra ID.

3.2 Access Levels

Stack Manager's Web interface supports the following access levels:

- Security Administrator
- Administrator
- Operator
- Monitor

The following table shows operations accessible to each access level:

	Security Administrator	Administrator	Operator	Monitor
Change global configuration parameters	✓			
Manage users	✓			
Perform upgrade via Web interface	✓			
Create new stacks	✓	✓		
Manage existing stacks	✓	✓	✓	
Monitor existing stacks	✓	✓	✓	✓

The default user (Admin) is created with Security Administrator access level.



Security Administrator access level was introduced in Stack Manager version 3.5.0. In earlier versions, Administrator access level had access to global configuration parameters, upgrade through the Web interface, and user management.

3.3 Managing Users

In Stack Manager version 3.5.0 and later, you can manage users allowed to access the Stack Manager Web interface through the **Users** screen.

The screen is accessible to users with Security Administrator access level.

It supports the following operations:

- Creating a new user.
- Modifying an existing user's properties, for example, username, password, or access level.
- Deleting a user.

Figure 3-2: Users Screen

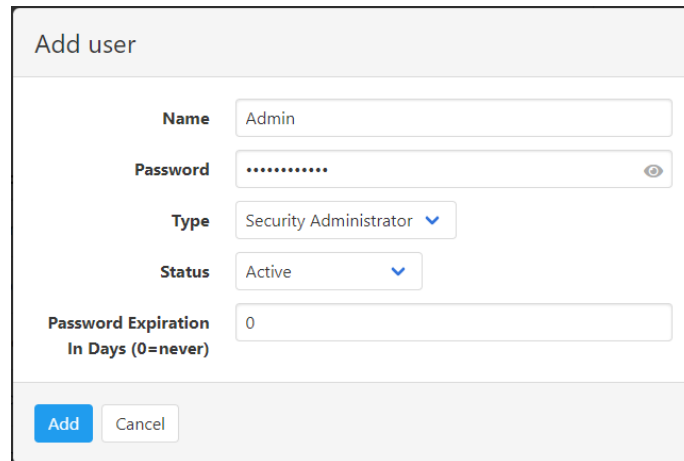
Name	Type	Status	Password Expiration	Actions
admin	Administrator	Active	Never	Modify Delete
monitor	Monitor	Active	Never	Modify Delete
operator	Operator	Active	After 30 days (28 days remaining)	Modify Delete
secadmin	Security Administrator	Active	Never	Modify Delete

The following statuses are supported:

- "Active": User can access the Stack Manager
- "Change Password": User must change password on next login
- "Locked": User is temporarily blocked from accessing Stack Manager

In addition, you can define the password expiration time (in days), after which the user is required to change the password.

Figure 3-3: Add User Dialog



The screenshot shows a web-based dialog box titled "Add user". It contains several input fields and dropdown menus. The "Name" field is filled with "Admin". The "Password" field is masked with dots and has a small eye icon to its right. The "Type" dropdown menu is set to "Security Administrator". The "Status" dropdown menu is set to "Active". The "Password Expiration In Days (0=never)" field is filled with "0". At the bottom of the dialog, there are two buttons: "Add" (highlighted in blue) and "Cancel".

You may also use the following CLI commands to manage Web users:

```
$ stack_mgr user-list
$ stack_mgr user-add <username> --password <password> ...
$ stack_mgr user-modify <username> ...
$ stack_mgr user-delete <username>
```

For Stack Manager versions earlier than 3.5.0, you could define a single set of credentials for each supported access level: Administrator, Operator and Monitor. This was done via the **Configuration** screen.

3.3.1 Changing Your Own Password

Every user may change his own password via the **Configuration** screen.

Figure 3-4: Changing Your Own Password

The screenshot shows the Stack Manager web interface. The top navigation bar includes the Stack Manager logo, links for 'Stacks', 'Configuration', 'Logs', 'Files', and 'About', the user role 'operator', and a 'Logout' button. The main content area is titled 'Configuration' and contains two sections:

- Change Password:** This section has three input fields: 'Current Password', 'New Password', and 'New Password (One More Time)'.
- Security:** This section has one input field labeled 'Number Of Other Sessions' with the value '0'.

At the bottom of the configuration area is a blue button with a checkmark icon and the text 'Update'.

3.3.2 Password Complexity

Starting with version 3.4.7, Stack Manager enforces the following password complexity rules:

- at least 12 characters long
- must include 3 of the following 4 character types: lowercase, uppercase, digit, special symbol

If you want to disable these rules, use the following CLI command:

```
$ stack_mgr configure --password-complexity disable
```

Starting with version 3.5.0 you may override default username and password complexity rules by providing your own regular expressions via the following CLI commands:

```
$ stack_mgr configure --password-complexity-regex '<regex>'
$ stack_mgr configure --username-complexity-regex '<regex>'
```

It is recommended to use single quotes for <regex> to prevent character escaping by shell.

For example use the following <regex> to enforce passwords that are at least 14 characters long and include 4 different character types - lowercase, uppercase, digit and special symbols:

```
^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*() ?]).{14,}$
```

3.3.3 Password Reuse

Starting with version 3.5.0 Stack Manager prevents users from re-using password that they used before. Up to 5 previous passwords are stored for each user.

Security Administrators, who modify user password via **Users** screen are exempt from password history check by default. If you want to change this behavior, use the following CLI command:

```
$ stack_mgr configure -user-modify-password-history enable
```

3.4 Global Configuration

The Configuration page contains global configuration parameters of the Stack Manager application.

After changing the value of a parameter, click **Update** (located at the bottom of the page) to apply the changes.

To verify current configuration, click **Verify**. See Section 2.8.5, Verifying Configuration for more information.

Figure 3-5: Configuration Page

The screenshot shows the 'Configuration' page in the Stack Manager application. The page is organized into several panels:

- General:** Includes fields for 'Name Prefix', a dropdown for 'Show Welcome Screen' (set to 'Enable'), and a text input for 'Welcome Screen Custom Text'.
- Change Password:** Features a text input for 'Current Password'.
- Microsoft Azure:** Contains fields for 'Subscription ID', a dropdown for 'Cloud' (set to 'Public'), 'Tenant ID', 'Client ID', 'Secret', 'Blob Account Name', and 'Blob Account Key'.
- Obfuscation Algorithm:** A dropdown menu set to 'Universal Key'.
- Summary Format:** A dropdown menu set to 'Regular'.
- Use Configuration Package:** A dropdown menu set to 'Enable'.
- Docker Deployment Mode:** A dropdown menu set to 'Disable'.
- Debug:** Includes a 'Download' button for the 'Debug File' and a dropdown for 'Debug File Access Level' set to 'Administrator'.

At the bottom of the configuration area, there are two buttons: a blue 'Update' button with a checkmark icon and a blue 'Verify' button with a question mark icon.

3.4.1 General Configuration Parameters

The following table describes general global configuration parameters.

Parameter	Description
Name Prefix	Defines prefix for all cloud resources created during stack creation – e.g. virtual machines, network security groups etc. See Section 2.10, Resource Naming for more information.
Show Welcome Screen	Defines whether welcome screen is displayed after the login to Web interface. Supported values: “Enable”, “Disable” Welcome screen contains information about last login time for specific user, IP address from where login was performed and number of unsuccessful login attempts (if there were such). It may also contain custom text defined via Welcome Screen Custom Text parameter.
Welcome Screen Custom Text	Defines custom text to be displayed on welcome screen. Use “\n” delimiter to specify multi-line text block, for example: “Welcome to the Stack Manager!\nPlease note that all your activity will be recorded.”

3.4.2 Change Password Block

Change password block enables users to change their passwords and consists of the following parameters.

Parameter	Description
Current Password	Current user's password
New Password	New password
New Password (One More Time)	New password again (to prevent typing errors)

3.4.3 Security Configuration Parameters

The following table describes security-related global configuration parameters.

Parameter	Description
Hostname	Defines hostname of Stack Manager's VM. When hostname is defined, access to the Stack Manager's Web interface is allowed only via it, and not via the corresponding IP address. See Section 3.5.4, Enforcing Secure Connection for more information.

Parameter	Description
Enforce HTTPS	Enforces secure connection (via the HTTPS protocol) for accessing the Stack Manager's Web interface. Supported values: "Enable", "Disable". See Section 3.5.4, Enforcing Secure Connection for more information.
Session Expiration	Defines expiration timeout for Web sessions. Supported values: "Disabled", "5 min", "10 min", "15 min", "30 min", "1 hour", "2 hours", "3 hours", "6 hours", "12 hours", "24 hours". Sessions that extend the expiration time without any activity will be closed and users will be forced to re-login.
Max Sessions Per Account	Defines maximum number of simultaneous sessions per user / account. Supported values: "Unlimited", 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 Note that sessions that are not explicitly closed by clicking a Logout button will remain active until the timeout defined via the Session Expiration parameter.
Number of Other Sessions	Read-only parameter that displays number of other active sessions for the current user / account. If number is greater than zero, Cleanup button is also displayed, allowing to "cleanup" other sessions. You may also clear Web sessions via the following CLI command: <code>stack_mgr clear-web-sessions <username></code>
Failed Login IP Block Duration	Stack Manager automatically blocks source IP addresses after 5 failed login attempts withing the 5 minute interval. This is done to prevent brute force attacks on the Web interface's authentication mechanism. This parameter defines duration of this block. Supported values: "1 min", "3 min", "5 min", "10 min", "15 min", "30 min", "1 hour".
Failed Login Account Block	Stack Manager may also block specific account / user after pre-defined number of failed login attempts that use corresponding username. Note that if you enable this behavior it may prevent legitimate users from logging into the system in case of attack by malicious users. Only new logins are blocked – existing already logged in sessions remain unaffected. Supported values: "Disable", "After 5 attempts", "After 10 attempts", "After 15 attempts", "After 20 attempts", "After 30 attempts", "After 40 attempts", "After 50 attempts".
Failed Login Account Block Duration	Defines duration of a per account / user block. Supported values: "1 min", "3 min", "5 min", "10 min", "15 min", "30 min", "1 hour"
REST API Mode	Defines availability and authorization scheme for REST API. Supported values: "Enable", "Disable", "Enable with Basic auth", "Enable with Azure auth", "Enable with any Auth". See Section 5.3, Authorization for details.

3.4.4 Microsoft Azure Parameters

The following table describes global configuration parameters applicable to Azure environment.

Parameter	Description
Subscription ID	<p>Defines default Azure Subscription ID for deployed stacks.</p> <p>For Stack Manager versions 3.2.5 and later you can choose a different Azure Subscription ID as part of stack creation dialog. So this global configuration parameter only defines the default subscription value.</p> <p>For Stack Manager versions prior to 3.2.5 all stacks were deployed to the Subscription ID defined via this global configuration parameter.</p>
Cloud	<p>Defines Azure cloud where deployments are done.</p> <p>Supported values: "Public", "US Government".</p>
Tenant ID Client ID Secret	<p>Leave these parameters blank if you are using the recommended deployment method, where Stack Manager VM is granted access to Azure APIs via managed system identity – as described in Section 2.8.2.2, Enabling Access to Azure APIs via Managed Service Identity (Recommended Method).</p> <p>However if you choose to use Service Principal instead, as described in Section 2.8.2.3, Enabling Access to Azure APIs via Service Principal (Alternative Method), you need to configure these parameter with values that match your Service Principal.</p>
Blob Account Name Blob Account Key Blob SAS Token Blob Container	<p>Configure these parameters if you want Stack Manager to store its runtime data on Microsoft Azure Storage Service.</p> <p>See Section 2.9.2, Storing Runtime Data on Azure Storage Service for detailed instructions.</p>
Application Insights Connection String	<p>Configure this parameter if you want Stack Manager to send metrics and alarms data to Azure Application Insights.</p> <p>See Section 7.3, Integration with Azure Application Insights for more information.</p>
Application Insights Mode	<p>Defines data that will be sent to Azure Application Insights.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ "all" – both metrics and alarms data will be sent for all stacks ■ "specific" – data will be sent only for stacks that have app_insights advanced configuration parameter; the later also determines what data will be sent – metrics / alarms / all; refer to Section 3.8.11, Advanced Configuration for detailed description.
Keep-Alive Alarm	<p>Defines whether Stack Manager periodically sends "keep-alive" alarm to Azure Application Insights. The alarm may be used to determine that Stack Manager application is alive.</p> <p>Supported values: "Enable", "Disable"</p>

Parameter	Description
Azure Login	Defines whether Stack Manager's Web interface uses Azure Entra ID for user authorization. Supported values: "Enable", "Optional", "Disable". See Section 3.6, Login via Azure Entra ID for detailed description.
Login Authority	Defines login authority for authorization via Azure Entra ID. See Section 3.6, Login via Azure Entra ID for detailed description.
INI Repository	Defines Azure Storage Account that contains INI files repository. When repository is defined, Stack Manager allows choosing files from it as part of "Send INI File" dialog for Mediant VE and Mediant CE stacks. Syntax: "account=<account-name>;container=<container-name>;key=<account-key>" It is possible to specify "token" (SAS token) instead of "key".

3.4.5 Amazon AWS Parameters

The following table describes global configuration parameters applicable to AWS environment.

Parameter	Description
Access Key Secret Key	Leave these parameters blank if you are using the recommended deployment method, where Stack Manager VM is granted access to AWS APIs via IAM role – as described in Section 2.8.1.1, Enabling Access to AWS API via IAM Role (Recommended Method). However if you choose to use AWS Secret Key instead, as described in Section 2.8.1.2, Enabling Access to AWS API via AWS Access Key (Alternative Method) you need to configure these parameter with proper values.
S3 Bucket S3 Prefix	Configure these parameters if you want Stack Manager to store its runtime data on Amazon S3 Service. See Section 2.9.1, Storing Runtime Data on AWS S3 for detailed instructions.

3.4.6 Google Cloud Parameters

The following table describes global configuration parameters applicable to Google Cloud environment.

Parameter	Description
Project	Defines project where all stacks are deployed. See Section 2.8.3.1, Configuring Google Project ID for more information.

Parameter	Description
Credentials	<p>Leave these parameters blank if you are using the recommended deployment method, where Stack Manager VM is granted access to Google Cloud APIs via Service Account – as described in Section 2.8.3.4, Enabling Access to Google Cloud APIs via Service Account (Recommended Method).</p> <p>However if you choose to use Configuration File instead, as described in Section 2.8.3.5, Enabling Access to Google Cloud APIs via Configuration File (Alternative Method), you need to configure these parameter with corresponding value.</p>
Storage Bucket Storage Prefix	<p>Configure these parameters if you want Stack Manager to store its runtime data on Google Cloud Storage Service.</p> <p>See Section 2.9.3, Storing Runtime Data on Google Cloud Storage Service for detailed instructions.</p>

3.4.7 Openstack Parameters

The following table describes global configuration parameters applicable to Openstack environment.

Parameter	Description
Cloud Name	<p>Defines cloud name where all stacks are deployed.</p> <p>See Section 2.8.4, Post-installation Configuration on OpenStack for more information.</p>
Container	<p>Configure this parameter if you want Stack Manager to store its runtime data on Openstack Object Storage Service.</p> <p>See Section 2.9.4, Storing Runtime Data on OpenStack Object Storage Service for detailed instructions.</p>

3.4.8 Debug File Parameters

The following table describes global configuration parameters applicable to Debug File.

Parameter	Description
Debug File	<p>Downloads Stack Manager Debug File used for bug reporting and troubleshooting.</p> <p>See Section 2.13, Providing Debug File for Troubleshooting for more information.</p>
Debug File Access Level	<p>Defines minimum access level that is allowed to download Debug File.</p> <p>Supported values: "Security Administrator", "Administrator", "Operator", "Monitor"</p>

3.4.9 Advanced Parameters

The following table describes global advanced configuration parameters.

Parameter	Description
Stack Manager Tag	<p>Defines global tag applied to all cloud resources (e.g. virtual machines and network security groups) created during stack creation.</p> <p>Syntax: "<tag_name>=<tag_value>"</p> <p>You may use %IP% element in <tag_value> that will be expanded to Stack Manager IP address.</p> <p>See Section 7.2, Tagging Stack Resources for more information.</p>
Auto Stop Time Auto Start Time Auto Shelve Time	<p>Define time for automatic stop / start / shelve operations.</p> <p>See Section 7.1, Automatic Stop / Start / Shelve for more information.</p>
Delete Public IPs During Shelve	<p>Defines whether public IPs are deleted during "shelve" operation.</p> <p>Supported valued: "Enable", "Disable"</p>
Stop / Start Individual Components	<p>Defines whether Stack Manager's "stop" and "start" operations allow user to choose specific component to be stopped / started.</p> <p>Supported values: "Enable", "Disable"</p>
Default Instance Types	<p>Defines default instance types used by Stack Manager in "stack create" dialog.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ Production – default instance type are suitable for production deployments ■ Development – minimal (and typically burstable) instance types are used by default; note that these instance types are not suitable for production deployment and software is not guaranteed to behave correctly on them
Update Protocol	<p>Defines transport protocol used during Stack Manager software update.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ HTTP or HTTPS – software automatically determines transport type based on current network conditions ■ HTTPS – use secure transport type only
Obfuscation Algorithm	<p>Defines algorithm used for sensitive data obfuscation in stack descriptors.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ Simple – simple transposition algorithm; this was the only algorithm supported in Stack Manager versions prior to 3.2.4 ■ Universal Key – data is encrypted using AES256 cipher with universal key; this is the default behavior for Stack Manager version 3.2.4 and later ■ Custom Key – data is encrypted using AES256 cipher with custom key. <p>Custom key must be created using the following CLI command:</p>

Parameter	Description
	<pre>stack_mgr obfuscate-key --generate</pre> <p>Use the following command to view the generated key:</p> <pre>stack_mgr obfuscate-key --show</pre> <p>And the following command to set it to specific value:</p> <pre>stack_mgr obfuscate-key --set <key></pre>
Summary Format	<p>Defines format for stack summary list in Web interface's Stacks screen.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ Regular – summary list includes regular fields ■ Verbose – summary list includes Subscription ID and Comments
Use Configuration Package	<p>Use configuration package for backup/restore of SBC configuration during “rebuild” and “update” operations on Mediant VE and Mediant CE stacks.</p> <p>Supported values: “Enable”, “Disable”</p>
Docker Deployment Mode	<p>Enables experimental Docker deployment mode for Mediant VE and CE stacks on Azure.</p>

3.5 Securing Connection to Web Interface

The Web interface is accessible by default through both an insecure (HTTP) and a secure (HTTPS) connection.

However, the default TLS certificates, which are installed during Stack Manager installation are self-signed. Therefore, when you attempt to connect to the Web interface through a secure (HTTPS) connection, your browser will display a security error. For most browsers, it's possible to ignore the error and proceed to the site, for example, in Google Chrome you can click **Advanced** and then **Proceed to <address> (unsafe)**.

To remove the security error screen when connecting to the Stack Manager's Web interface through a secure (HTTPS) connection, you must do the following:

4. Configure a hostname for the Stack Manager's virtual machine.
5. Acquire and install a certificate for the configured hostname.

3.5.1 Configuring Hostname for Stack Manager Virtual Machine

Certificates are typically issued for hostnames and not for IP addresses. Therefore, prior to acquiring a certificate, you must configure a hostname for the Stack Manager virtual machine.

For production deployments, contact your IT administrator and request a hostname for the Stack Manager virtual machine.

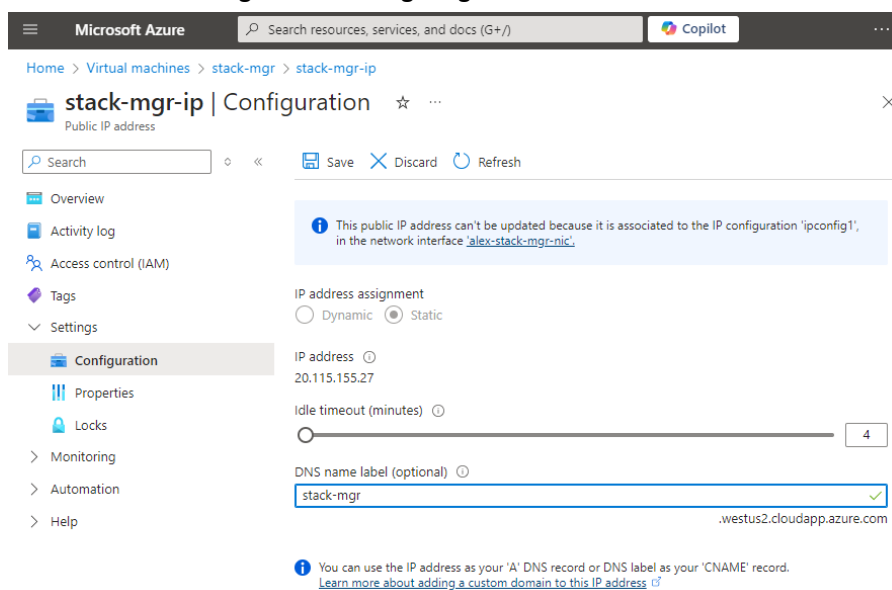
For lab deployments, you can use DNS names provided by your cloud provider. Note that these DNS names will lack your company branding (e.g., for Azure deployments they will belong to the cloudapp.azure.com domain). Refer to your cloud provider documentation for detailed instructions on how to configure such DNS names.

For example, if your Stack Manager is deployed in Azure and has a public IP address, you can allocate it a DNS name as follows:

1. Open the Azure portal at <https://portal.azure.com/>.
2. Navigate to **Virtual Machines**.
3. Select the Stack Manager virtual machine.
4. In the **Overview** screen, locate the **Public IP address** assigned to the VM and click it; the Public IP address configuration screen opens.
5. In the Public IP address configuration screen, under **Settings**, switch to the **Configuration** pane.
6. Under **DNS name label (optional)** enter the DNS name, and then click **Save**.

Note that the full DNS name assigned by Azure consists of the **DNS name label** entered by you, followed by a DNS suffix, automatically allocated by Azure depending on the VM's location. For example, if you enter "stack-mgr" for the **DNS name label** and your VM is located in WestUS2 region, Azure will assign it the following DNS name: "stack-mgr.westus2.cloudapp.azure.com".

Figure 3-6: Configuring DNS Name in Azure



Verify that you can access Stack Manager's Web interface via the configured hostname, by entering the `http://<hostname>` URL in the browser.

3.5.2 Acquiring Certificate from Certificate Authority

After successfully configuring a hostname for the Stack Manager's virtual machine, you should acquire a certificate for this hostname.

Contact a Certificate Authority (CA) of your choice and follow the provided instructions. You would typically be requested to generate a private key and submit a certificate signing request (CSR). The CA will generate a certificate based on the submitted CSR and provide it back to you as a PEM file. Make sure that the provided certificate file is in PEM format and perform a proper conversion if needed.

Install the private key and certificate files (in PEM format) on Stack Manager's virtual machine, as follows:

1. Upload the private key and certificate files to the Stack Manager virtual machine (e.g., through an SCP/SFTP) and place them in the following locations:
 - Place the private key in `/etc/ssl/nginx/server.key`.
 - Place the certificate in `/etc/ssl/nginx/server.crt`.
2. Restart the NGINX server by running the following command as a regular Linux user (e.g., "debian"):

```
sudo systemctl restart nginx
```

3.5.3 Installing Let's Encrypt Certificates

Let's Encrypt (<https://letsencrypt.org>) is a Certificate Authority that provides free certificates through the ACME protocol.

If your Stack Manager's Web interface is exposed via the public IP address, you can install Let's Encrypt certificates by doing the following:

To install Let's Encrypt certificates:

1. Open Certbot ACME client official site at <https://certbot.eff.org>
2. In **My HTTP website is running** section, select:
 - a. Software: **Nginx**
 - b. System: Choose the operating system installed on the Stack Manager virtual machine. If you are not sure, use the `cat /etc/os-release` command to find out the operating system.
3. Log in to the Stack Manager virtual machine as a regular user (e.g., "debian").
4. Install the Certbot ACME client per provided instructions.
5. Edit the NGINX server configuration file and add the `server_name` parameter to the `server` section, e.g.,:

```
server {  
    listen 80;  
    listen 443 ssl;  
    server_name stack-mgr.westeurope.cloudapp.azure.com  
    ...  
}
```

Replace `stack-mgr.westeurope.cloudapp.azure.com` with the hostname of the Stack Manager virtual machine.

6. The location of the NGINX server configuration file depends on the operating system being used:
 - For Ubuntu / Debian, edit the `/etc/nginx/sites-enabled/stack_mgr` file
 - For RHEL / CentOS / Amazon Linux, edit the `/etc/nginx/nginx.conf` file
7. Restart the NGINX server:

```
sudo systemctl restart nginx
```
8. Run the Certbot client:

```
sudo certbot --nginx
```

3.5.4 Enforcing Secure Connection

Verify that you can successfully access Stack Manager's Web interface through a secure (HTTPS) connection, by entering the `https://<hostname>` URL in the browser.

If the connection is successful, it's recommended to enforce secure connection, by configuring the following parameters in the global Configuration screen:

- **Hostname** – enter the hostname of the Stack Manager virtual machine
- **Enforce HTTPS** – set it to "enabled"

3.6 Login via Azure Entra ID


If you deployed Stack Manager in a Microsoft Azure environment, you may use Azure Entra ID (formerly Active Directory) to control access to Stack Manager's Web interface.

When such mode is enabled, users have to authenticate using their Azure credentials while logging into the Stack Manager's Web interface. Role-based access for specific users / groups is granted via the Azure portal.



Prior to enabling login via Azure Entra ID, you must secure connection to the Stack Manager's Web interface, as described in Section 3.5.

To enable login via Azure Entra ID:

1. Open the Azure portal at <https://portal.azure.com/>.
2. If you have access to multiple tenants, click the **Directory + subscription** filter  to select the tenant in which you want to register an application.
3. Navigate to the App registrations page.
4. Click **New registration**.
5. Enter a display **Name** for the new application (e.g., "Stack Manager").
6. Specify who can use the application:
 - If you want to allow only users from your organization to access the Stack Manager, choose **Accounts in this organizational directory only**.
 - If you want to allow users from any organization to access the Stack Manager, choose, **Accounts in any organizational directory**.
7. Under the Redirect URI group, select **Web** from the drop-down list, and then enter the following redirect URI:
8. **https://<hostname>/azureToken**
Replace <hostname> with the hostname of the Stack Manager virtual machine.
9. Click **Register** to register the new application.
10. In the app's Overview screen, find and note the **Application (client) ID** and **Directory (tenant) ID**.
11. Switch to the Authentication screen and under the **Implicit grant and hybrid flows** group, enable **ID tokens**. Click **Save** to apply the changes.
12. Open a new web browser tab.
13. Log in to the Stack Manager Web interface.
14. Open the Configuration page.
15. Under the Microsoft Azure group, enter the following values:
 - a. 'Client ID': Application (client) ID
 - b. 'Tenant ID': **Directory (tenant) ID**
 - c. 'Azure Login': **Enabled**
 - d. 'Login Authority': Configure the same value as was chosen during the application registration

16. Click **Update**.
17. Keep the browser tab open, in case you fail to login using Azure Entra ID and decide to revert the Azure Login configuration parameter back to 'Disabled'.
18. On the Azure app's registration page, switch to the App roles screen.
19. Click **Create app role**, enter the following values, and then click **Apply**:
 - a. 'Display name': **SecAdmin**
 - b. 'Allowed member types': **Users/Groups**
 - c. 'Value': **SecAdmin**
 - d. 'Description': **SecAdmin**
20. Click **Create app role**, enter the following values, and then click **Apply**:
 - a. 'Display name': **Admin**
 - b. 'Allowed member types': **Users/Groups**
 - c. 'Value': **Admin**
 - d. 'Description': **Admin**



For Stack Manager versions prior to 3.5.0 create **Administrator** role instead of **SecAdmin** and **Admin** roles. When you upgrade to version 3.5.0 or later you may keep using this Administrator role, as it will be mapped to "Security Administrator" access level (similar to SecAdmin role).

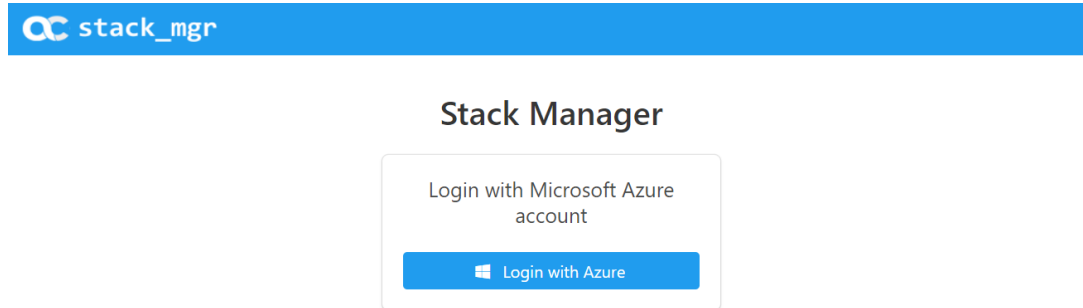
21. Click **Create app role**, enter the following values, and then click **Apply**:
 - a. 'Display name': **Operator**
 - b. 'Allowed member types': **Users/Groups**
 - c. 'Value': **Operator**
 - d. 'Description': **Operator**
22. Click **Create app role**, enter the following values, and then click **Apply**:
 - a. 'Display name': **Monitor**
 - b. 'Allowed member types': **Users/Groups**
 - c. 'Value': **Monitor**
 - d. 'Description': **Monitor**
23. Navigate to the Enterprise applications page.
24. Locate the application registered in the previous steps, by setting 'Application type' to **All applications** and typing the registered application name (e.g. "Stack Manager") into the search string.
25. Click the application.
26. Select **Users and groups**.
27. For each user or group that you want to grant access to the Stack Manager application:
 - a. Click Add user/group.
 - b. Select the user or group.
 - c. Select the role – **SecAdmin, Admin, Operator, or Monitor**.
 - d. Click **Assign**.



For Stack Manager versions prior to 3.5.0 use **Administrator** role instead of **SecAdmin** and **Admin** roles.

28. Open a new browser tab.
29. Navigate to the Stack Manager's Web interface.
30. Click the **Login with Azure** button.
31. Enter your Azure credentials and then verify that you can successfully log in.

Figure 3-7: Login via Azure Entra ID



3.7 Resetting Web Interface Credentials

If you can't access the Web interface because you forgot your username and/or password, use the following CLI commands to configure new credentials:

- **Stack Manager version 3.5.0 and later:**

- List existing users:

```
$ stack_mgr user-list
```

- Change the password for the existing user:

```
$ stack_mgr user-modify <username> --password <password>
```

Alternatively, you can create a new user:

```
$ stack_mgr user-add <username> --password <password>
```

- **Stack Manager versions earlier than 3.5.0:**

```
$ stack_mgr configure --rest-api-username <username>
```

```
$ stack_mgr configure --rest-api-password <password>
```

If your Stack Manager instance is configured to login through Azure Entra ID and you want to revert it back to local login, use the following CLI command:

```
$ stack_mgr configure --azure-login disable
```

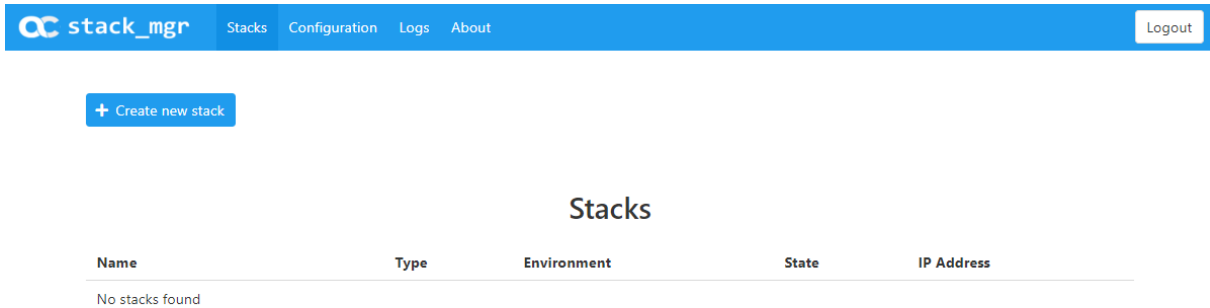
3.8 Creating a New Stack

The procedure below describes how to create a new stack.

To create a new Mediant VE/CE stack:

1. Open the Stacks page.

Figure 3-8: Creating a New Stack



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

Figure 3-9: Create New Stack Dialog

3. In the 'Name' field, enter the stack name.
4. From the 'Environment' drop-down list, select the public cloud / virtual environment; the dialog box is updated with the relevant parameters.
5. Refer to the following sections for detailed instructions for each public cloud / virtual environment.



Prior to creating a new Mediant CE stack, make sure that all pre-requisites specified in the *Mediant Cloud Edition Installation Manual* are met. The document can be downloaded from AudioCodes website at <https://www.audiocodes.com/library/technical-documents>.

3.8.1 Creating Mediant CE in Amazon Web Services (AWS) Environment

The following configuration parameters should be configured (in the Create new stack dialog) for Mediant CE stack in Amazon Web Services (AWS) environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant CE**.
- 'Environment': **AWS**.
- 'Region': Defines the region where Mediant CE is to be deployed.
- 'Key Pair': Defines the key pair for logging in to the Mediant CE CLI through SSH. Alternatively, you can log in using the password specified below. If not needed, leave it as **none**.
- 'IAM Role': Defines the name of the IAM role that enables Mediant CE access to AWS APIs for network reconfiguration in case of Signaling Component switchover. Refer to *Mediant Cloud Edition SBC for AWS Installation Manual* for detailed instructions on how to create it. Make sure that you use an IAM role that matches your deployment topology (see below).
- **Networking:**
 - 'Deployment Topology': Defines Mediant CE deployment topology.
 - ◆ 'single zone': All Mediant CE components are deployed in a single Availability Zone.
 - ◆ 'multiple zones': Mediant CE components are spread across two Availability Zones.
 - 'VPC': Defines the Virtual Private Cloud where Mediant CE is to be deployed.
 - 'Cluster Subnet': Defines the subnet within the VPC for internal communication between Mediant CE components. The subnet must have a private EC2 endpoint or NAT Gateway configured as the default route. Refer to *Mediant Cloud Edition Installation Manual* for detailed instructions on how to create it.
 - 'Main Subnet': Defines the subnet within the VPC for carrying management traffic (e.g., connecting to the Mediant CE Web or SSH interface). The subnet can also be used for carrying signaling and media traffic.
 - '1st and 2nd Additional Subnet': Defines additional subnets for carrying signaling and media traffic. If not needed, leave them as – **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant CE network interfaces) for which Elastic IPs are assigned.
 - 'Use private IP address for management': Applicable when the public IP address is assigned to the 'Main' interface. It creates an additional private IP address on the 'Main' interface and configures it as the management interface.



- For 'single zone' deployment topology, all specified subnets must reside in the same Availability Zone.
- For 'multiple zones' deployment topology, you need to choose two subnets – one in each Availability Zone.

- **Signaling Components:**
 - 'VM Type': Defines the instance type for Signaling Component instances.
- **Media Components:**
 - 'Profile': Defines the operational mode of Media Components:
 - ◆ 'forwarding' – Media Components lack DSP resources and can perform media traffic forwarding, including conversion between RTP and SRTP.

- ◆ 'transcoding' – Media Components have DSP resources and can perform conversion from one vocoder to another and DTMF detection.
 - 'VM Type': Defines the instance type for Media Component instances.
 - 'Max Number': Defines the total number of Media Components that will be created. It also defines the higher boundary for scale-out operation.
 - 'Min Number': Defines the number of Media Components that will be initially active after Mediant CE creation. It also defines the lower boundary for scale-in operation.
- **Admin User:**
- 'Username': Defines the username for logging in to the Mediant CE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant CE Web or SSH interface.
- **Advanced:**
- 'SBC version': Defines the deployed SBC software version. The displayed list corresponds to SBC software versions published in AWS Marketplace.
 - 'Management ports': Defines a list of inbound ports and corresponding transport protocols for management traffic. It's used for configuring AWS network security groups (NSGs) for the "main" network interface of Signaling Components. 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)
 You can also specify "icmp" or "icmpv6" instead of "<port>/<protocol>".
 For example: "22/tcp,80/tcp,443/tcp,161/udp".
 - 'Signaling ports': Defines a list of inbound ports and corresponding transport protocols for signaling traffic. It's used for configuring AWS NSGs for network interfaces of Signaling Components capable of handling signaling traffic. The syntax is similar to the 'management ports'. For example: "5060/udp,5060/tcp,5061/tcp".
 - 'Media ports': Defines a list of inbound ports and corresponding transport protocols for media traffic. It's used for configuring AWS NSGs for network interfaces of Media Components capable of handling media traffic. The syntax is similar to the 'management ports'. For example: "6000-65535/udp".
 - 'Use main subnet for!': Defines the type of traffic handled by the interface connected to the "main" subnet. It's used for configuring AWS NSGs, and the default SIP Interface and Media Realm.
 - ◆ 'all traffic (management + VoIP)' – all types of traffic are allowed.
 - ◆ 'management traffic only' – only management traffic is allowed.
 - 'Advanced config': Defines additional configuration parameters, as described in Section 3.8.11.1 Advanced Configuration for Mediant CE.

Figure 3-10: Configuring Mediant CE in AWS Environment

Create new stack

Name

Stack type Mediant CE ▾

Environment AWS ▾

Region EU (Frankfurt) ▾

Key Pair aws_ssh_frankfurt_1 ▾

IAM Role

Networking

VPC vpc-45f3152c (DefaultVPC) ▾

Cluster Subnet subnet-0496039603680f5a2 (cluster) ▾

Create
Cancel

Once you have configured all the above parameters, click **Create** to create the Mediant CE stack instance. The operation progress is displayed at the top of the page.

Figure 3-11: Creating Mediant CE in AWS environment

stack_mgr
Stacks Configuration Logs About Logout

+ Create new stack

Creating stack 'ce1'

Initializing AWS client... done

Creating network resources..... done

Creating media components.....

Stacks

Name	Type	Environment	State	IP Address
ce1	Mediant CE	AWS	creating	

3.8.1.1 Troubleshooting

The following table lists common problems during Mediant CE stack creation in the AWS environment and their corresponding solutions.

Table 3-1: Troubleshooting Mediant CE Stack Creation in AWS Environment

Problem	Reason	Solution
Mediant CE stack creation freezes at the "Creating media components" step for more than 10 minutes. No Media Component instances are shown in the AWS dashboard.	You haven't subscribed to the Mediant VE offer in AWS Marketplace.	Subscribe to Mediant VE offer in AWS Marketplace, as described in <i>Mediant Cloud Edition Installation Manual</i> .
	The IAM role specified during Mediant CE stack creation doesn't exist.	Create an IAM role for Mediant CE, as described in <i>Mediant Cloud Edition Installation Manual</i> and specify its name in the Mediant CE Create stack dialog box.

For other problems, go to the **Cloud Formation** service in the AWS dashboard, locate the stack that corresponds to the deployed Mediant CE name, switch to the **Events** tab, and then check for any additional errors.

3.8.2 Creating Mediant CE in Azure Environment

The following configuration parameters should be configured (in the **Create new stack** dialog) for Mediant CE stack in the Azure environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant CE**.
- 'Environment': **Azure**.
- 'Region': Defines the region where Mediant CE is to be deployed.
- **Compute:**
 - 'Deployment topology': Defines the topology for Mediant CE deployment:
 - ◆ 'availability set' - Mediant CE components are deployed in a single Proximity Placement Group to minimize network latency, with two Availability Sets for Signaling Components and Media Components, respectively. Each Availability Set is configured with two fault and update domains. This deployment topology provides 99.95% SLA at the infrastructure level.
 - ◆ 'availability zones' – Mediant CE components are spread across two Availability Zones. Each group of components are deployed in dedicated Proximity Placement Group to minimize network latency. This deployment topology provides 99.99% SLA at the infrastructure level, but has a slightly higher chance of suffering from temporary lack of resources in specific Azure datacenters.
 - 'Zones': comma-separated list of two availability zones, for example: **1,2**; applicable only when 'Deployment topology' is set to 'availability zones'.
- **Networking:**
 - 'Virtual Network': Defines the virtual network where Mediant CE is to be deployed.
 - 'Cluster Subnet': Defines the subnet used for internal communication between Mediant CE components.
 - 'Main Subnet': Defines the subnet for management traffic (Web, SSH, etc.). The subnet can also be used for signaling (SIP) and media (RTP) traffic.
 - 1st and 2nd Additional Subnet': Defines additional subnets for signaling (SIP) and media (RTP) traffic. If not needed, leave them as -- **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant CE network interfaces) for which Public IPs are assigned.
 - 'Use private IP address for management': Applicable when the public IP address is assigned to the 'Main' interface. It creates an additional private IP address on the 'Main' interface (behind the Internal Load Balancer) and configures it as the management interface.
- **Signaling Components:**
 - 'VM Type': Defines the VM size for Signaling Component instances.
- **Media Components:**
 - 'Profile': Defines the operational mode of Media Components:
 - ◆ 'forwarding' – Media Components lack DSP resources and can perform media traffic forwarding, including conversion between RTP and SRTP.
 - ◆ 'transcoding' – Media Components have DSP resources and can perform conversion from one vocoder to another and DTMF detection.
 - 'VM Type': Defines the VM size for Media Component instances.
 - 'Max Number': Defines the total number of Media Components that will be created. It also defines the higher boundary for scale-out operation.

- 'Min Number': Defines the number of Media Components that will be initially active after Mediant CE creation. It also defines the lower boundary for scale-in operation.

■ **Admin User:**

- 'Username': Defines the username for logging in to the Mediant CE Web or SSH interface.
- 'Password': Defines the password for logging in to the Mediant CE Web or SSH interface.



Azure imposes the following limitations on the username and password:

■ **Username:**

- A minimum of 4 and a maximum of 20 characters.
- May not be one of the commonly used usernames, as listed in <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-username-requirements-when-creating-a-vm> (comparison is case insensitive).
- May not end with dot (.).

■ **Password:**

- A minimum of 12 and a maximum of 123 characters.
- May not be one of the commonly used passwords, as listed in <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-password-requirements-when-creating-a-vm> .
- Must use three out of four of the following: lowercase characters, uppercase characters, numbers, and symbols.
- May not be the same as the username.

■ **Advanced:**

- 'SBC version': Defines the deployed SBC software version. The displayed list corresponds to the SBC software versions published in Azure Marketplace.
- 'Management ports': Defines a list of inbound ports and corresponding transport protocols for the management traffic. It's used for configuring Azure network security groups (NSGs) and Public Load Balancer routing rules for the "main" network interface of Signaling Components. 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,80/tcp,443/tcp,161/udp".

- 'Signaling ports': Defines a list of inbound ports and corresponding transport protocols for signaling traffic. It's used for configuring Azure NSGs and Public Load Balancer routing rules for network interfaces of Signaling Components capable of handling signaling traffic. The syntax is similar to the 'management ports'. For example: "5060/udp,5060/tcp,5061/tcp".
- 'Media ports': Defines a list of inbound ports and corresponding transport protocols for media traffic. It's used for configuring Azure NSGs for network interfaces of Media Components capable of handling media traffic. The syntax is similar to the 'management ports'. For example: "6000-65535/udp"

- ‘Use main subnet for’: Defines the type of traffic handled by the interface connected to the “main” subnet. It's used for configuring Azure NSGs, and the default SIP Interface and Media Realm.
 - ◆ ‘all traffic (management + VoIP)’ – All types of traffic are allowed.
 - ◆ ‘management traffic only’ – Only management traffic is allowed.
- ‘Advanced config’: Defines additional configuration parameters, as described in Section 3.8.11.1 Advanced Configuration for Mediant CE.

Figure 3-12: Configuring Mediant CE in Azure Environment

Once you have configured all the above parameters, click **Create** to create the Mediant CE stack instance. The operation progress is displayed at the top of the page.

Figure 3-13: Creating Mediant CE in Azure Environment

Name	Type	Environment	State	IP Address
ce1	Mediant CE	Azure	creating	



If Stack Manager is assigned with custom IAM roles at Subscription, Network and Resource Group levels, as described in Section 2.8.2.2.2, Option 2: Custom IAM Roles at Subscription, Network and Resource Group Levels, an empty Resource Group must be manually created prior to stack deployment and Stack Manager must be assigned with “Contributor” role in it. The name of this Resource Group must be specified during stack creation by the Advanced Config parameter **resource_group**.

3.8.2.1 Troubleshooting

The following table lists common problems during Mediant CE stack creation in the Azure environment and their corresponding solutions.

Table 3-2: Troubleshooting Mediant CE Stack Creation in Azure Environment

Problem	Reason	Solution
Mediant CE stack creation fails with error message "Legal terms have not been accepted for this item on this subscription"	You haven't subscribed to the Mediant VE offer in Azure Marketplace.	Subscribe to Mediant VE offer in Azure Marketplace, by deploying a demo instance of it. Refer to <i>Mediant Cloud Edition Installation Manual</i> for detailed description.
Mediant CE stack creation fails with error message "Creating resource group <stack_name> failed"	Stack Manager creates a new Resource Group for each stack with the same name as the stack name (unless resource_group advanced config parameter is used). If your subscription already has such a resource group, stack creation will fail.	Use a different stack name that doesn't match the name of any existing Resource Group in your subscription. Alternatively, you can configure the 'Name Prefix' parameter in the Stack Manager configuration screen.

For other problems, go to the **Resource Group** in the Azure portal that matches the deployed Mediant CE name, switch to the **Deployments** tab, and then check for any errors.

3.8.3 Creating Mediant CE in Google Cloud Environment

The following configuration parameters should be configured (in the Create new stack dialog) for Mediant CE stack in Google Cloud environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant CE**.
- 'Environment': **Google**.
- 'Region': Defines the region where Mediant CE is to be deployed.
- 'Zones': Defines a comma-separated list of two Availability Zones within the specified Region. Mediant CE components will be evenly spread across these two zones.
- 'Image': Defines the name of the Mediant VE/CE image. Refer to *Mediant Cloud Edition Installation Manual* for detailed instructions on how to upload it to your account.
- **Networking:**
 - 'Cluster Subnet': Defines the subnet used for internal communication between Mediant CE components.
 - 'Main Subnet': Defines the subnet for carrying management traffic (e.g. connecting to the Mediant CE Web or SSH interface) and signaling traffic. The subnet can also be used for carrying media traffic.
 - 1st and 2nd Additional Subnet': Defines additional subnets used for carrying media traffic. If not needed leave them as – **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant CE network interfaces) for which External IPs are assigned.
 - 'Use private IP address for management': Applicable when the public IP address is assigned to the 'Main' interface. It creates an additional private IP address on the 'Main' interface (behind the Internal Load Balancer) and configures it as the management interface.
- **Signaling Components:**
 - 'VM Type': Defines the machine type for Signaling Component instances.
- **Media Components:**
 - 'Profile': Defines the operational mode of Media Components:
 - ◆ 'forwarding' – Media Components lack DSP resources and can perform media traffic forwarding, including conversion between RTP and SRTP.
 - ◆ 'transcoding' – Media Components have DSP resources and can perform conversion from one vocoder to another and DTMF detection.
 - 'VM Type': Defines the machine type for Media Component instances.
 - 'Max Number': Defines the total number of Media Components that will be created. It also defines the higher boundary for scale-out operation.
 - 'Min Number': Defines the number of Media Components that will be initially active after Mediant CE creation. It also defines the lower boundary for scale-in operation.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant CE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant CE Web or SSH interface.

■ **Advanced:**

- 'Management ports': Defines a list of inbound ports and corresponding transport protocols for management traffic. It's used for configuring Google Cloud firewall rules for the "main" network interface of Signaling Components. 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,80/tcp,443/tcp,161/udp"

- 'Signaling ports': Defines a list of inbound ports and corresponding transport protocols for signaling traffic. It's used for configuring Google Cloud firewall rules for network interfaces of Signaling Components capable of handling signaling traffic. The syntax is similar to the 'management ports'. For example: "5060/udp,5060/tcp,5061/tcp".
- 'Media ports': Defines a list of inbound ports and corresponding transport protocols for media traffic. It's used for configuring Google Cloud firewall rules for network interfaces of Media Components capable of handling media traffic. The syntax is similar to the 'management ports'. For example: "6000-65535/udp".
- 'Advanced config': Defines additional configuration parameters, as described in Section 3.8.11.1 Advanced Configuration for Mediant CE.

Figure 3-14: Configuring Mediant CE in Google Cloud Environment

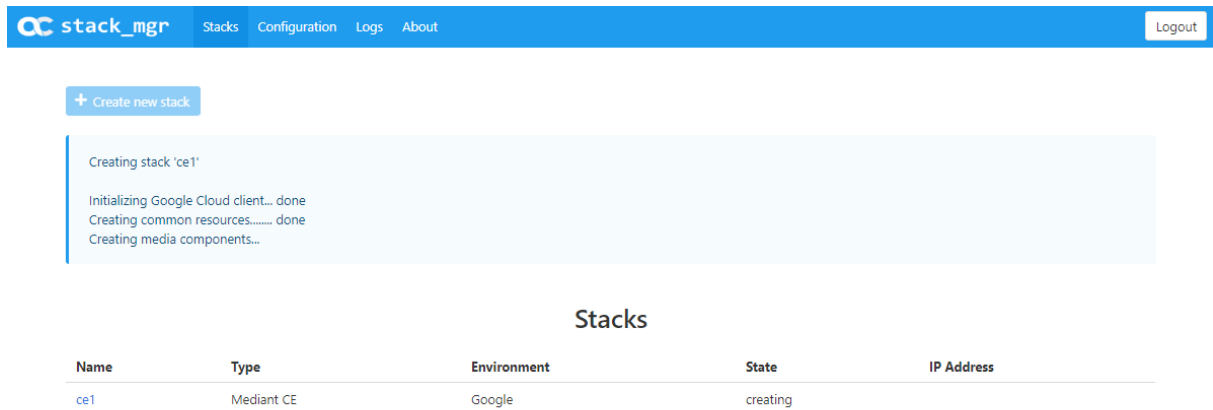
The screenshot shows a 'Create new stack' dialog box with the following configuration:

- Name:** ce1
- Stack type:** Mediant CE
- Environment:** Google
- Region:** us-central1
- Zones:** a,b
- Image:** sbc-7-20-256-110
- Networking:**
 - Cluster Subnet:** cluster
 - Main Subnet:** oam

Buttons: Create, Cancel

Once you have configured all the above parameters, click **Create** to create the Mediant CE stack instance. The operation progress is displayed at the top of the page.

Figure 3-15: Creating Mediant CE in Google Cloud Environment



The screenshot displays the Stack Manager web interface. At the top, there is a blue navigation bar with the logo 'stack_mgr' and menu items: 'Stacks', 'Configuration', 'Logs', and 'About'. A 'Logout' button is located on the right side of the bar. Below the navigation bar, there is a '+ Create new stack' button. A light blue box shows the progress of creating stack 'ce1':

- Creating stack 'ce1'
- Initializing Google Cloud client... done
- Creating common resources..... done
- Creating media components...

Below the progress box, the title 'Stacks' is centered. Underneath, a table lists the stack instances:

Name	Type	Environment	State	IP Address
ce1	Mediant CE	Google	creating	

3.8.4 Creating Mediant CE in OpenStack Environment

The following configuration parameters should be configured (in the Create new stack dialog) for Mediant CE stack in OpenStack environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant CE**.
- 'Environment': **OpenStack**.
- 'Image': Defines the name of the Mediant VE/CE image. Refer to *Mediant Cloud Edition Installation Manual* for detailed instructions on how to upload it to your account.
- 'Key Pair': Defines the key pair for logging in to the Mediant CE CLI through SSH. Alternatively, you can log in using the password specified below.
- **Networking:**
 - 'Cluster Subnet': Defines the subnet for internal communication between Mediant CE components.
 - 'Main Subnet': Defines the subnet for carrying management traffic (e.g., connecting to the Mediant CE Web or SSH interface). The subnet can also be used for carrying signaling and media traffic.
 - '1st and 2nd Additional Subnet': Defines additional subnets for carrying signaling and media traffic. If not needed, leave them as **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant CE network interfaces) for which Floating IPs are assigned.
- **Signaling Components:**
 - 'VM Type': Defines the flavor for Signaling Component instances. Refer to *Mediant Cloud Edition Installation Manual* for recommended flavors.
- **Media Components:**
 - 'Profile': Defines the operational mode of Media Components:
 - ◆ 'forwarding' – Media Components lack DSP resources and can perform media traffic forwarding, including conversion between RTP and SRTP.
 - ◆ 'transcoding' – Media Components have DSP resources and can perform conversion from one vocoder to another and DTMF detection.
 - 'VM Type': Defines the flavor for Media Component instances. Refer to *Mediant Cloud Edition Installation Manual* for recommended flavors.
 - 'Max Number': Defines the total number of Media Components that will be created. It also defines the higher boundary for scale-out operation.
 - 'Min Number': Defines the number of Media Components that will be initially active after Mediant CE creation. It also defines the lower boundary for scale-in operation.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant CE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant CE Web or SSH interface.
- **Advanced:**
 - 'Advanced config': Defines additional configuration parameters as described in Section 3.8.11.1 Advanced Configuration for Mediant CE.

Figure 3-16: Configuring Mediant CE in OpenStack Environment

Create new stack

Name

Stack type

Environment

Image

Key Pair

Networking

Cluster Subnet

Main Subnet

1st Additional Subnet

Once you have configured all the above parameters, click **Create** to create the Mediant CE stack instance. The operation progress is displayed at the top of the page.

Figure 3-17: Creating Mediant CE in OpenStack Environment

stack_mgr Stacks Configuration Logs About Logout

+ Create new stack

Creating stack 'ce1'

Initializing Openstack client... done

Creating network resources..... done

Creating media components...

Stacks

Name	Type	Environment	State	IP Address
ce1	Mediant CE	OpenStack	creating	

3.8.5 Creating Mediant VE in Amazon Web Services (AWS) Environment

The following configuration parameters should be configured (in the Create new stack dialog) for Mediant VE stack in Amazon Web Services (AWS) environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant VE**.
- 'Environment': **AWS**.
- 'Region': Defines the region where Mediant VE is to be deployed.
- 'Key Pair': Defines the key pair for logging in to the Mediant VE CLI through SSH. Alternatively, you can log in using the password specified below. If not needed, leave it as -- **none** --.
- 'IAM Role': Defines the name of the IAM role that enables Mediant VE access to AWS APIs for network reconfiguration in case of Signaling Component switchover. Refer to *Mediant Virtual Edition SBC for Amazon AWS Installation Manual* for detailed instructions on how to create it. Make sure that you use the IAM role that matches your deployment topology (see below).
- **Compute:**
 - 'HA Mode': Defines whether Mediant VE is deployed in HA mode (that includes two EC2 instances operating in Active/Standby mode) or as a single EC2 instance.
 - 'VM Type': Defines instance type used for Mediant VE deployment.
- **Networking:**
 - 'Deployment Topology': Defines Mediant VE deployment topology:
 - ◆ 'single zone': All Mediant VE components are deployed in a single Availability Zone.
 - ◆ 'multiple zones': Mediant VE components are spread across two Availability Zones.
 - 'VPC': Defines the Virtual Private Cloud where Mediant VE is to be deployed.
 - 'HA Subnet': (for HA deployment only) Defines the subnet within the VPC for internal communication between Mediant VE instances. The subnet must have a private EC2 endpoint or NAT Gateway configured as the default route. Refer to *Mediant Virtual Edition SBC for Amazon AWS Installation Manual* for detailed instructions on how to create it.
 - 'Main Subnet': Defines the subnet within the VPC for carrying management traffic (e.g., connecting to the Mediant VE Web or SSH interface). The subnet can also be used for carrying signaling and media traffic.
 - '1st and 2nd Additional Subnet': Defines additional subnets for carrying signaling and media traffic. If not needed, leave them as – **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant VE network interfaces) for which Elastic IPs are assigned.
 - 'Use private IP address for management': Applicable when the public IP address is assigned to the 'Main' interface. It creates an additional private IP address on the 'Main' interface and configures it as the management interface.



- For 'single zone' deployment topology, all specified subnets must reside in the same Availability Zone.
- For 'multiple zones' deployment topology, you need to choose two subnets – one in each Availability Zone.

- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant VE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant VE Web or SSH interface.
- **Advanced:**
 - 'SBC version': Defines the deployed SBC software version. The displayed list corresponds to the SBC software versions published in AWS Marketplace.
 - 'Management ports': Defines a list of inbound ports and corresponding transport protocols for management traffic. It's used for configuring AWS network security groups (NSGs) for the "main" network interface. 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)You can also specify "icmp" or "icmpv6" instead of "<port>/<protocol>".
For example: "22/tcp,80/tcp,443/tcp,161/udp"
 - 'Signaling ports': Defines a list of inbound ports and corresponding transport protocols for signaling traffic. It's used for configuring AWS NSGs and for network interfaces capable of handling signaling traffic. The syntax is similar to the 'management ports', for example: "5060/udp,5060/tcp,5061/tcp".
 - 'Media ports': Defines a list of inbound ports and corresponding transport protocols for media traffic. It's used for configuring AWS NSGs for network interfaces capable of handling media traffic. The syntax is similar to the 'management ports', for example: "6000-65535/udp".
 - 'Use main subnet for': Defines the type of traffic handled by the interface connected to the "main" subnet. It's used for configuring AWS NSGs and the default SIP Interface and Media Realm.
 - ◆ 'all traffic (management + VoIP)' – All types of traffic are allowed.
 - ◆ 'management traffic only' – Only management traffic is allowed.
 - 'Advanced config': Defines additional configuration parameters as described in Section 3.8.11.2 Advanced Configuration for Mediant VE.

Figure 3-18: Configuring Mediant VE in AWS Environment

Once you have configured all the above parameters, click **Create** to create the Mediant VE stack instance. The operation progress is displayed at the top of the page.

3.8.5.1 Troubleshooting

The following table lists common problems during Mediant VE stack creation in the AWS environment and their corresponding solutions.

Table 3-3: Troubleshooting Mediant VE Stack Creation in AWS Environment

Problem	Reason	Solution
Mediant VE stack creation freezes at the "Creating stack " step for more than 10 minutes. No EC2 instances are shown in the AWS dashboard.	You haven't subscribed to the Mediant VE offer in AWS Marketplace.	Subscribe to Mediant VE offer in AWS Marketplace, as described in <i>Mediant Virtual Edition SBC for Amazon AWS Installation Manual</i> .
	The IAM role specified during Mediant VE stack creation doesn't exist.	Create an IAM role for Mediant VE, as described in <i>Mediant Virtual Edition SBC for Amazon AWS Installation Manual</i> and specify its name in the Mediant VE Create stack dialog box.

For other problems, go to the **Cloud Formation** service in the AWS dashboard, locate the stack that matches the deployed Mediant CE name, switch to the **Events** tab, and then check for any errors.

3.8.6 Creating Mediant VE in Azure Environment

The following configuration parameters should be configured (in the Create new stack dialog) for Mediant VE stack in the Azure environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant VE**.
- 'Environment': **Azure**.
- 'Region': Defines the region where Mediant VE is to be deployed.
- **Compute:**
 - 'HA Mode': Defines if Mediant VE is deployed in HA mode (includes two virtual machines operating in Active/Standby mode) or as a single virtual machine.
 - 'VM Type': Defines the virtual machine size for Mediant VE deployment.
 - 'Deployment topology': Defines the topology for Mediant CE deployment:
 - ◆ 'availability set' - Mediant CE components are deployed in a single Proximity Placement Group to minimize network latency, with two Availability Sets for Signaling Components and Media Components, respectively. Each Availability Set is configured with two fault and update domains. This deployment topology provides 99.95% SLA at the infrastructure level.
 - ◆ 'availability zones' – Mediant CE components are spread across two Availability Zones. Each group of components are deployed in dedicated Proximity Placement Group to minimize network latency. This deployment topology provides 99.99% SLA at the infrastructure level, but has a slightly higher chance of suffering from temporary lack of resources in specific Azure datacenter.
 - 'Zones': comma-separated list of two availability zones, for example: **1,2**; applicable only when 'Deployment topology' is set to 'availability zones'.
- **Networking:**
 - 'Virtual Network': Defines the virtual network where Mediant VE is to be deployed.
 - 'HA Subnet': (HA deployment only) Defines the subnet for internal communication between Mediant VE instances. If you leave it as -- **none** --, internal communication is done via the secondary IP address on the network interface that is connected to the "Main" subnet (eth0:2).
 - 'Main Subnet': Defines the subnet for carrying management traffic (e.g., connecting to the Mediant VE Web or SSH interface). The subnet can also be used for carrying signaling and media traffic.
 - 1st and 2nd Additional Subnet': Defines additional subnets for carrying signaling and media traffic. If not needed, leave them as -- **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant VE network interfaces) for which Public IPs are assigned.
 - 'Use private IP address for management': Applicable when the public IP address is assigned to the 'Main' interface. It creates an additional private IP address on the 'Main' interface (behind the Internal Load Balancer) and configures it as the management interface.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant VE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant VE Web or SSH interface.



Azure imposes the following limitations on the username and password:

■ **Username:**

- A minimum of 4 and a maximum of 20 characters.
- May not be one of the commonly used usernames, as listed in <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-username-requirements-when-creating-a-vm> (comparison is case insensitive).
- May not end with dot (.).

■ **Password:**

- A minimum of 12 and a maximum of 123 characters..
- May not be one of the commonly used passwords, as listed in <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-password-requirements-when-creating-a-vm> .
- Must use three out of four of the following: lowercase characters, uppercase characters, numbers, and symbols.
- May not be the same as the username.

■ **Advanced:**

- 'SBC version': Defines a list of inbound ports and corresponding transport protocols for the management traffic. It's used for configuring Azure network security groups (NSGs) and Public Load Balancer routing rules for the "main" network interface of Signaling Components. 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,80/tcp,443/tcp,161/udp".

- 'Signaling ports': Defines a list of inbound ports and corresponding transport protocols for signaling traffic. It's used for configuring Azure NSGs and Public Load Balancer routing rules for network interfaces capable of handling signaling traffic. The syntax is similar to the 'management ports'. For example: "5060/udp,5060/tcp,5061/tcp".
- 'Use main subnet for': Defines type of traffic handled by the "main" subnet. Affects configuration of network security groups.
- 'Advanced config': Defines additional configuration parameters as described in Section 3.8.11.2 Advanced Configuration for Mediant VE.

Figure 3-19: Configuring Mediant VE in Azure Environment

Once you have configured all the above parameters, click **Create** to create the Mediant VE stack instance. The operation progress is displayed at the top of the page.



If Stack Manager is assigned with custom IAM roles at Subscription, Network and Resource Group levels, as described in Section 2.8.2.2.2, Option 2: Custom IAM Roles at Subscription, Network and Resource Group Levels, an empty Resource Group must be manually created prior to stack deployment and Stack Manager must be assigned with “Contributor” role in it. The name of this Resource Group must be specified during stack creation by the Advanced Config parameter **resource_group**.

3.8.6.1 Troubleshooting

The following table lists common problems during Mediant VE stack creation in the Azure environment and their corresponding solutions.

Table 3-4: Troubleshooting Mediant VE Stack Creation in Azure Environment

Problem	Reason	Solution
Mediant VE stack creation fails with the error message “Legal terms have not been accepted for this item on this subscription”.	You haven’t subscribed to the Mediant VE offer in Azure Marketplace.	Subscribe to Mediant VE offer in Azure Marketplace by deploying a demo instance of it. Refer to <i>Mediant Virtual Edition SBC for Azure Installation Manual</i> for detailed description.

For other problems, go to the **Resource Group** in the Azure portal that matches the deployed Mediant VE name, switch to the **Deployments** tab, and then check for any errors.

3.8.7 Creating Mediant VE in Google Cloud Environment

The following configuration parameters should be configured (in the Create new stack dialog) for Mediant CE stack in Google Cloud environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Mediant VE**.
- 'Environment': **Google**.
- 'Region': Defines the region where Mediant VE is to be deployed.
- 'Zones': Defines a comma-separated list of two Availability Zones within the specified Region. Mediant VE components will be evenly spread across these two zones.
- 'Image': Defines the name of the Mediant VE/CE image. Refer to *Mediant Virtual Edition for Google Cloud Installation Manual* for detailed instructions on how to upload it to your account.
- **Compute:**
 - 'HA Mode': Defines whether Mediant VE is deployed in HA mode (that includes two VM instances operating in Active/Standby mode) or as a single VM instance.
 - 'VM Type': Defines machine type used for Mediant VE deployment.
- **Networking:**
 - 'HA Subnet': (for HA deployment only) Defines the subnet used for internal communication between Mediant VE components. If you leave it as -- **none** --, internal communication is done through the network interface connected to the "Main" subnet (eth0).
 - 'Main Subnet': Defines the subnet for carrying management traffic (e.g. connecting to the Mediant VE Web or SSH interface) and signaling traffic. The subnet can also be used for carrying media traffic.
 - 1st and 2nd Additional Subnet': Defines additional subnets used for carrying media traffic. If not needed leave them as -- **none** --.
 - 'Public IPs': Defines subnets (and corresponding Mediant VE network interfaces) for which External IPs are assigned.
 - 'Use private IP address for management': Applicable when the public IP address is assigned to the 'Main' interface. It creates an additional private IP address on the 'Main' interface (behind the Internal Load Balancer) and configures it as the management interface.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant VE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant VE Web or SSH interface.
- **Advanced:**
 - 'Management ports': Defines a list of inbound ports and corresponding transport protocols for management traffic. It's used for configuring Google Cloud firewall rules for the "main" network interface. 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,80/tcp,443/tcp,161/udp"

- 'Signaling ports': Defines a list of inbound ports and corresponding transport protocols for signaling traffic. It's used for configuring Google Cloud firewall rules for network interfaces capable of handling signaling traffic. The syntax is similar to the 'management ports'. For example: "5060/udp,5060/tcp,5061/tcp".
- 'Media ports': Defines a list of inbound ports and corresponding transport protocols for media traffic. It's used for configuring Google Cloud firewall rules for network interfaces capable of handling media traffic. The syntax is similar to the 'management ports'. For example: "6000-65535/udp".
- 'Advanced config': Defines additional configuration parameters as described in Section 3.8.11.2 Advanced Configuration for Mediant VE.

Figure 3-20: Configuring Mediant VE in Google Cloud Environment

The screenshot shows a 'Create new stack' form with the following fields and values:

- Name:** ve1
- Stack type:** Mediant VE
- Environment:** Google
- Region:** us-central1
- Image:** sbc-7-20-256-110
- Compute:**
 - HA Mode:** enable
 - VM Type:** n1-standard-2
- Networking:** (Section header, no values visible)

At the bottom of the form are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

Once you have configured all the above parameters, click **Create** to create the Mediant VE stack instance. The operation progress is displayed at the top of the page.

3.8.8 Creating VoiceAI Connect in Amazon Web Services (AWS) Environment



The VoiceAI Connect application is disabled by default. Contact AudioCodes support if you want to enable it.

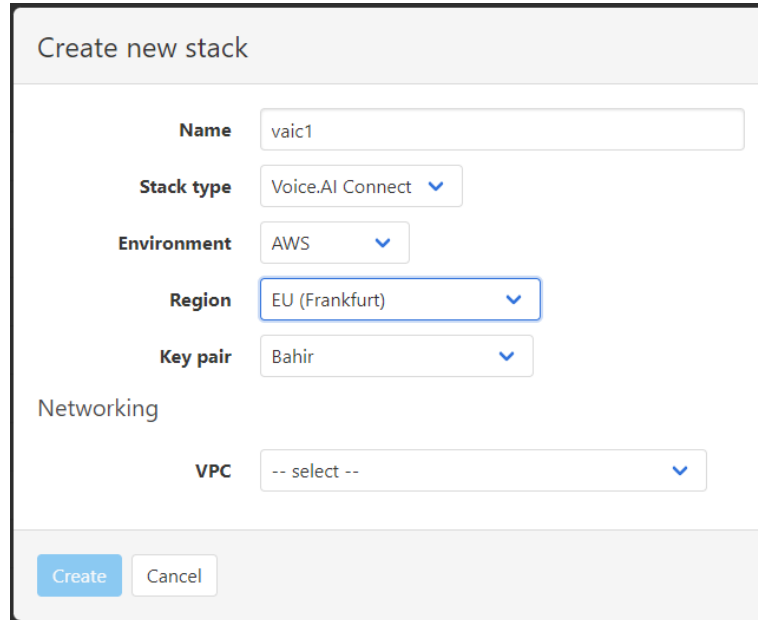
The following configuration parameters should be configured (in the Create new stack dialog) for VoiceAI Connect stack in Amazon Web Services (AWS) environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Voce.AI Connect.**
- 'Environment': **AWS.**
- 'Region': Defines the region where VoiceAI Connect stack is to be deployed.
- 'Key Pair': Defines the key pair for logging in to the stack components' virtual machines through SSH. Alternatively, you can log in using the password specified below.
- **Networking:**
 - 'VPC': Defines the Virtual Private Cloud where VoiceAI Connect stack is to be deployed.
 - 'Main Subnet': Defines the main subnet within the VPC to which all VoiceAI Connect stack components are connected via primary network interface (eth0).
 - 'Public Subnet': (Optional) Defines the public subnet within the VPC.
 - ◆ If configured:
 - "Worker" SBC instances will have additional network interface (eth1) connected to this subnet and assigned with Elastic IP address
 - Front-end SBC will be configured to work in "direct media" mode – i.e., it will handle only signaling (SIP) traffic, while media (RTP) traffic will be passed directly to the "worker" SBC instances via "public" (eth1) interface
 - ◆ If not configured:
 - Media (RTP) traffic will be latched on the front-end SBC and relayed to the "worker" SBC instances via "main" (eth0) interface



All specified subnets must reside in the same Availability Zone.

- **Session Managers**
 - 'Max Number': Defines the total number of "worker" pairs (session manager + SBC) that will be created. It also defines the higher boundary for scale-out operation.
 - 'Min Number': Defines the number of "worker" pairs (session manager + SBC) that will be initially active after stack creation. It also defines the lower boundary for scale-in operation.
- **Front-End SBC**
 - 'Frontend SBC': Defines the name of Mediant VE or CE stack that will be used as a "load balancer" in front of the VoiceAI Connect stack. If configured, "worker" SBC instances will be configured to send / receive calls to / from the corresponding front-end SBC's addresses.
 - 'Create initial config': Defines whether Stack Manager should create configuration on the front-end SBC that forwards the incoming traffic to the "worker" SBC instances, handles outbound and transferred calls. Note that regardless of this parameter value Stack Manager will always create and maintain basic connectivity configuration on the front-end SBC that includes IP Group, Proxy Set and Proxy IPs.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant VE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant VE Web or SSH interface.
- **Advanced:**
 - 'Automatic Update URL': (Optional) Defines the automatic update URL that will be configured on "worker" SBC instances and front-end SBC. If configured:
 - ◆ On the front-end SBC, the `IncrementalIniFileURL` parameter is provisioned with the `${url}/global/fe-incremental.ini` value.
 - ◆ On "worker" SBCs, the `IncrementalIniFileURL` parameter is provisioned with the `${url}/global/sbc-incremental.ini` value.
 - 'Use Bot Dialplan': Configure the bot dialplan on "worker" SBCs. If enabled, the `bots` dialplan is created and the `DialPlanCSVFileUrl` configuration parameter is provisioned with the `${url}/global/bot-dialplan.csv` value.
 - 'Voice.AI Connect Version': Defines the software version that will be installed on the session manager and configuration manager / data center instances.
 - 'Voice.AI Connect Host OS': Defines the operating system that will be used on the session manager and configuration manager / data center instances.
 - 'Syslog Server': Defines IP address of syslog server
 - 'Management via Public IPs': If enabled, all stack components are provisioned with public IP addresses on management interfaces.
 - 'Advanced Config': Defines additional configuration parameters as described in Section 3.8.11.3 Advanced Configuration for VoiceAI Connect.

Figure 3-21: Configuring Voce.AI Connect in AWS Environment

The screenshot shows a 'Create new stack' form with the following fields:

- Name:** vaic1
- Stack type:** Voice.AI Connect
- Environment:** AWS
- Region:** EU (Frankfurt)
- Key pair:** Bahir
- Networking:**
 - VPC:** -- select --

At the bottom of the form are two buttons: 'Create' and 'Cancel'.

Once you have configured all the above parameters, click **Create** to create the VoiceAI Connect stack instance. The operation progress is displayed at the top of the page.

3.8.9 Creating VoiceAI Connect in Azure Environment



VoiceAI Connect application is disabled by default. Contact AudioCodes support if you want to enable it.

The following configuration parameters should be configured (in the Create new stack dialog) for VoiceAI Connect stack in Amazon Web Services (AWS) environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the dash symbol.
- 'Stack type': **Voce.AI Connect**.
- 'Environment': **Azure**.
- 'Region': Defines the region where VoiceAI Connect stack is to be deployed.
- **Networking:**
 - 'Virtual Network': Defines the virtual network where Mediant CE is to be deployed.
 - 'Main Subnet': Defines the main subnet to which all VoiceAI Connect stack components are connected via primary network interface (eth0).
 - 'Public Subnet': (Optional) Defines the public subnet.
 - ◆ If configured:
 - "Worker" SBC instances will have additional network interface (eth1) connected to this subnet and assigned with Public IP address
 - Front-end SBC will be configured to work in "direct media" mode – i.e., it will handle only signaling (SIP) traffic, while media (RTP) traffic will be passed directly to the "worker" SBC instances via "public" (eth1) interface
 - ◆ If not configured:
 - Media (RTP) traffic will be latched on the front-end SBC and relayed to the "worker" SBC instances via "main" (eth0) interface
- **Session Managers**
 - 'Max Number': Defines the total number of "worker" pairs (session manager + SBC) that will be created. It also defines the higher boundary for scale-out operation.
 - 'Min Number': Defines the number of "worker" pairs (session manager + SBC) that will be initially active after stack creation. It also defines the lower boundary for scale-in operation.
- **Front-End SBC**
 - 'Frontend SBC': Defines the name of Mediant VE or CE stack that will be used as a "load balancer" in front of the VoiceAI Connect stack. If configured, "worker" SBC instances will be configured to send / receive calls to / from the corresponding front-end SBC's addresses.
 - 'Create initial config': Defines whether Stack Manager should create configuration on the front-end SBC that forwards the incoming traffic to the "worker" SBC instances, handles outbound and transferred calls. Note that regardless of this parameter value Stack Manager will always create and maintain basic connectivity configuration on the front-end SBC that includes IP Group, Proxy Set and Proxy IPs.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant VE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant VE Web or SSH interface.

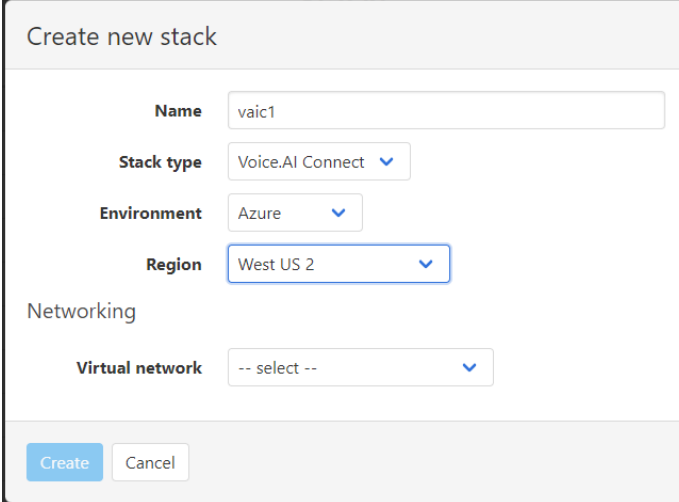


Azure imposes the following limitations on the username and password:

- **Username:**
 - A minimum of 4 and a maximum of 20 characters.
 - May not be one of the commonly used usernames, as listed in <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-username-requirements-when-creating-a-vm> (comparison is case insensitive).
 - May not end with dot (.).
- **Password:**
 - A minimum of 12 and a maximum of 123 characters.
 - May not be one of the commonly used passwords, as listed in <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-password-requirements-when-creating-a-vm> .
 - Must use three out of four of the following: lowercase characters, uppercase characters, numbers, and symbols.
 - May not be the same as the username.

- **Advanced:**

- 'Automatic Update URL': (Optional) Defines the automatic update URL that will be configured on "worker" SBC instances and the front-end SBC. If configured:
 - On the front-end SBC, the `IncrementalIniFileURL` parameter is provisioned with the `${url}/global/fe-incremental.ini` value.
 - On "worker" SBCs, the `IncrementalIniFileURL` parameter is provisioned with the `${url}/global/sbc-incremental.ini` value.
- 'Use Bot Dialplan': Configure the bot dialplan on "worker" SBCs. If enabled, the `bots dialplan` is created and the `DialPlanCSVFileUrl` configuration parameter is provisioned by the `${url}/global/bot-dialplan.csv` value.
- 'Voice.AI Connect Version': Defines the software version that will be installed on the session manager and configuration manager / data center instances.
- 'Voice.AI Connect Host OS': Defines the operating system that will be used on the session manager and configuration manager / data center instances.
- 'Syslog Server': Defines IP address of syslog server
- 'Management via Public IPs': If enabled, all stack components are provisioned with public IP addresses on management interfaces.
- 'Advanced Config': Defines additional configuration parameters as described in Section 3.8.11.3 Advanced Configuration for VoiceAI Connect.

Figure 3-22: Configuring Voce.AI Connect in Azure Environment

The screenshot shows a 'Create new stack' form with the following fields and values:

- Name:** vaic1
- Stack type:** Voice.AI Connect
- Environment:** Azure
- Region:** West US 2
- Networking:**
 - Virtual network:** -- select --

At the bottom of the form, there are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

Once you have configured all the above parameters, click **Create** to create the VoiceAI Connect stack instance. The operation progress is displayed at the top of the page.

3.8.10 Creating VoiceAI Connect in Google Cloud Environment



The VoiceAI Connect application is disabled by default. Contact AudioCodes support if you want to enable it.

The following configuration parameters should be configured (in the Create new stack dialog) for the VoiceAI Connect stack in Google Cloud environment:

- 'Name': Defines the stack name, which can contain lowercase or uppercase letters, digits, and the hyphen symbol.
- 'Stack type': **Voce.AI Connect.**
- 'Environment': **Azure.**
- 'Region': Defines the region where the VoiceAI Connect stack is to be deployed.
- **Networking:**
 - 'Main Subnet': Defines the main network / subnet to which all VoiceAI Connect stack components are connected via the primary network interface (eth0).
 - 'Public Subnet': (Optional) Defines the public network / subnet.
 - ◆ If configured:
 - "Worker" SBC instances will have an additional network interface (eth1) connected to this subnet and assigned with the Public IP address.
 - Front-end SBC will be configured to work in "direct media" mode. In other words, it will handle only signaling (SIP) traffic, while media (RTP) traffic will be passed directly to the "worker" SBC instances through the "public" (eth1) interface.
 - ◆ If not configured: Media (RTP) traffic is latched onto the front-end SBC and relayed to the "worker" SBC instances through the "main" (eth0) interface.
- **Session Managers:**
 - 'Max Number': Defines the total number of "worker" pairs (Session Manager with SBC) that will be created. It also defines the higher boundary for scale-out operation.
 - 'Min Number': Defines the number of "worker" pairs (Session Manager with SBC) that will be initially active after stack creation. It also defines the lower boundary for scale-in operation.
- **Front-End SBC:**
 - 'Frontend SBC': Defines the name of the Mediant VE or CE stack that will be used as a "load balancer" in front of the VoiceAI Connect stack. If configured, "worker" SBC instances will be configured to send / receive calls to / from the corresponding front-end SBC's addresses.
 - 'Create initial config': Defines whether Stack Manager should create configuration on the front-end SBC that forwards the incoming traffic to the "worker" SBC instances, handles outbound and transferred calls. Note that regardless of this parameter value, Stack Manager always creates and maintains basic connectivity configuration on the front-end SBC that includes IP Group, Proxy Set and Proxy IPs.
- **Admin User:**
 - 'Username': Defines the username for logging in to the Mediant VE Web or SSH interface.
 - 'Password': Defines the password for logging in to the Mediant VE Web or SSH interface.
- **Advanced:**

- ‘Automatic Update URL’: (Optional) Defines the automatic update URL that will be configured on “worker” SBC instances and the front-end SBC. If configured:
 - ◆ On front-end SBC, the `IncrementalIniFileURL` parameter is provisioned with the `${url}/global/fe-incremental.ini` value.
 - ◆ On “worker” SBCs, the `IncrementalIniFileURL` parameter is provisioned with the `${url}/global/sbc-incremental.ini` value.
- ‘Use Bot Dialplan’: Configures bot dialplan on “worker” SBCs. If enabled, a `bots dialplan` is created and the `DialPlanCSVFileUrl` configuration parameter is provisioned by the `${url}/global/bot-dialplan.csv` value.
- ‘Voice.AI Connect Version’: Defines the software version that will be installed on the Session Manager and configuration manager / data center instances.
- ‘Voice.AI Connect Host OS’: Defines the operating system that will be used on the Session Manager and configuration manager / data center instances.
- ‘Syslog Server’: Defines the IP address of the syslog server.
- ‘Management via Public IPs’: If enabled, all stack components are provisioned with public IP addresses on management interfaces.
- ‘Advanced Config’: Defines additional configuration parameters, as described in Section 3.8.11.3 Advanced Configuration for VoiceAI Connect.

Figure 3-23: Configuring Voce.AI Connect in Google Cloud Environment

The screenshot shows a 'Create new stack' dialog box with the following configuration:

- Name:** vaic1
- Stack type:** Voice.AI Connect
- Environment:** Google
- Region:** us-central1
- Zones:** a,b
- Networking:**
 - Main subnet:** default
 - Public subnet:** -- none --

At the bottom, there are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

Once you have configured all the above parameters, click **Create** to create the VoiceAI Connect stack instance. The operation progress is displayed at the top of the page.

3.8.11 Advanced Configuration

The Create new stack dialog includes the Advanced Config group that can be used to specify advanced configuration parameters during stack creation.

Specify parameters using the following format:

```
<parameter name> = <value>
```

You can specify multiple parameters on multiple lines.

Figure 3-24: Advanced Configuration Parameters

The screenshot shows a 'Create new stack' dialog box with the following fields:

- Min Number:** 2 (dropdown)
- Max Number:** 5 (dropdown)
- Admin User:**
 - Username:** sbcadmin
 - Password:**
- Advanced:**
 - Advanced Config:**

```
sc_public_ips = eth1.eth2
mc_public_ips = eth1.eth2
```

Buttons: Create, Cancel

3.8.11.1 Advanced Configuration for Mediant CE

The following table describes advanced parameters available for Mediant CE.

Table 3-5: Advanced Parameters Description

Parameter	Applicable Environment	Description	Apply Mode
accelerated_networking	Azure	Enables accelerated networking on D_v2, Dds_v3 and Dds_v4 instances. Note: Dds_v5 instances always have accelerated networking enabled and therefore, this parameter is not applicable. Supported values: <ul style="list-style-type: none"> ■ disable (Default) ■ enable Example: <pre>accelerated_networking = enable</pre>	Update
additional3_subnet_id, additional4_subnet_id, additional5_subnet_id, additional6_subnet_id	AWS, Azure	Defines subnet IDs for Additional 3 to Additional 6 subnets (connected to eth4 to eth8, respectively). For Azure, specify subnet name, e.g.,: <pre>additional3_subnet_id = voip3</pre>	Instant

Parameter	Applicable Environment	Description	Apply Mode
		<p>For AWS, specify subnet ID, e.g.,:</p> <pre>additional3_subnet_id = subnet-12345</pre> <p>Note: Subnet IDs that are currently “in use” can't be modified. If you want to change the subnet ID of an existing network interface, first reduce the number of network interfaces, update the corresponding subnet ID, and then restore the number of interfaces.</p>	
additional1_subnet_cidr, additional2_subnet_cidr, additional3_subnet_cidr, additional4_subnet_cidr, additional5_subnet_cidr, additional6_subnet_cidr	Azure	<p>Defines the CIDR for “additional 1”, “additional 2”, etc. subnets. This can be used to overcome Stack Manager’s lack of permissions to read current subnet configuration.</p> <p>Syntax: same as <code>main_subnet_cidr</code>.</p> <p>See also <code>cluster_subnet_cidr</code> parameter.</p>	Rebuild
additional_route_tables	AWS	<p>Applicable to multi-zone AWS deployments.</p> <p>Defines additional route tables that should be updated with virtual IP addresses.</p> <p>Syntax: comma-separated list of <interface name>:<route table ID>; multiple route table IDs can be specified using pipe () delimiter.</p> <p>Example:</p> <pre>additional_route_tables = eth1 :rtb-123,eth2:rtb-567 rtb-890</pre>	Update
app_insights	Azure	<p>Defines the type of data that is reported to Azure Application Insights.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ disable: (Default) Don't report any data for this stack. ■ enable: Report alarms and metrics (PMs). ■ alarms: Report alarms only. ■ metrics: Report metrics (PMs) only. <p>Example:</p> <pre>app_insights = enable</pre> <p>See Section 7.3, Integration with Azure Application Insights for more information.</p>	Instant
auto_shelve_time	AWS, Azure, Google	<p>Defines the time of day when the stack is automatically “shelved” (i.e., all VMs are stopped and unnecessary resources, for example, Media Components are deleted). If defined, it overrides the global Auto Shelve Time configuration parameter.</p> <p>Supported syntax:</p> <ul style="list-style-type: none"> ■ 08:00: Time of day (24h). ■ 1/08:00: Weekday (0 is Sunday, 1 is Monday, 2 is Tuesday, and so on) and time. 	Instant

Parameter	Applicable Environment	Description	Apply Mode
		<ul style="list-style-type: none"> ■ 0,1,2/08:00: Multiple weekdays and time. ■ 0-5/08:00: Range of weekdays and time. ■ 0,1/08:00 2-4/09:00: Multiple statements. Example: <pre>auto_shelve_time = 08:00</pre>	
auto_start_time	All	Defines the time of day when the stack is automatically started. If defined, it overrides the global Auto Start Time configuration parameter. Syntax is identical to <code>auto_shelve_time</code> parameter. Example: <pre>auto_start_time = 08:00</pre>	Instant
auto_stop_time	All	Defines the time of day when the stack is automatically stopped. If defined, it overrides the global Auto Stop Time configuration parameter. Syntax is identical to <code>auto_shelve_time</code> parameter. Example: <pre>auto_stop_time = 22:00</pre>	Instant
availability_zones	Azure	Defines availability zones where Mediant CE components are deployed. Syntax: <ul style="list-style-type: none"> ■ Two availability zone names separated by a comma. ■ “none” – components are deployed in to an availability set. Example: <pre>availability_zones = 1,2</pre>	Update
cluster_subnet_cidr	Azure	Defines the CIDR for the “cluster” subnet. This can be used to overcome Stack Manager’s lack of permissions to read current subnet configuration. Syntax: same as <code>main_subnet_cidr</code> . See also the <code>additionalX_subnet_cidr</code> parameter.	Rebuild
cluster_nsg_id	AWS, Azure	Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager. Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG. Syntax: <ul style="list-style-type: none"> ■ AWS: Security Group ID, e.g.,: 	Update

Parameter	Applicable Environment	Description	Apply Mode
		<pre>cluster_nsg_id = sg-11223344</pre> <ul style="list-style-type: none"> ■ Azure: Resource Group name / NSG name, e.g.,: <pre>cluster_nsg_id = rg1/cluster-nsg</pre> <p>See also the <code>main_nsg_id</code>, <code>media_nsg_id</code>, <code>oam_nsg_id</code> and <code>signaling_nsg_id</code> parameters.</p>	
common_tags	AWS	<p>Defines tags that are assigned to created network security groups.</p> <p>Syntax: comma-separated list of name=value pairs.</p> <p>Example:</p> <pre>common_tags = type=sbc</pre> <p>See also the <code>sc_tags</code> and <code>mc_tags</code> parameters.</p>	Rebuild
diag_account	Azure	<p>Defines the name of the existing Storage Account.</p> <p>If not empty, the specified Storage Account stores the VM's diagnostics data instead of creating a new one.</p> <p>Syntax: Resource Group name / account name</p> <p>Example:</p> <pre>diag_account = rg1/account1</pre>	Can't be modified after stack creation
disk_encryption_set	Azure	<p>Defines the Disk Encryption Set to implement server-side encryption with customer-managed key (SSE-CMK) for managed disks.</p> <p>Syntax: Resource Group name / Disk Encryption Set name</p> <p>Example:</p> <pre>disk_encryption_set = rg1/des1</pre>	Rebuild
eip_depends_on_instance	AWS	<p>Defines the dependency relation in Cloud Formation scripts used by Stack Manager to create / update stack resources.</p> <p>Prior to Version 3.1.0, Stack Manager defaulted to disable. In the middle of 2023, AWS implemented a change in Cloud Formation behavior that started causing "instance is not in a valid state" error during stack creation / update. To overcome this problem, this parameter was introduced and the default was changed to enable for newly created stacks.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ disable: EC2 instances depend on Elastic IPs. ■ enable: (Default) Elastic IPs depend on EC2 instances. 	Instant

Parameter	Applicable Environment	Description	Apply Mode
		Example: <code>eip_depends_on_instance = enable</code>	
ha_nlb	AWS	Enables the use of AWS Network Load Balancer for 1+1 HA implementation of Signaling Components. Supported values: <ul style="list-style-type: none"> ■ internal – use internal NLB instead of Virtual IPs ■ public – use public NLB instead of Elastic IPs ■ all – use internal / public NLB instead of Virtual / Elastic IPs Example: <code>ha_nlb = internal</code>	Can't be modified after stack creation
ini_incremental	All	Defines additional configuration parameters (in INI file format) for Signaling Components during stack creation / rebuild. Syntax: a single line with \n as line delimiter, e.g.,: <code>ini_incremental = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</code> Specified configuration is applied via REST API and therefore, has no size limit. Therefore, this parameter is preferred over the <code>sc_ini_params</code> parameter that is applied via cloud-init mechanism and is therefore, limited by instance user-data size.	Rebuild
imds	AWS	Defines the version of the AWS meta-data instance for the deployed EC2 instances. Supported values: <ul style="list-style-type: none"> ■ any: Allow both IMDSv1 and IMDSv2. ■ v2: (Default) Enforce IMDSv2. Example: <code>imds = v2</code>	Update
ipv6_ip_sc-X_eth* , ipv6_ip_mc-X_eth*	AWS	Defines pre-defined IPv6 addresses. See also the <code>private_ip_*</code> and <code>public_ip_*</code> parameters.	Update
ipv6_virtual_ips	AWS	Defines whether “virtual IPs” are used for IPv6 addresses in multi-zone AWS deployments. “Virtual IPs” apply only to internal traffic (within the subnet or via Transit Gateway) and therefore, have very limited application. Correspondingly, this parameter defaults to “disable” and it's recommended to use AWS Load Balancer or DNS-based methods for traffic distribution.	Update

Parameter	Applicable Environment	Description	Apply Mode
		<p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: Use “virtual IP” addresses for IPv6 addresses. ■ disable: (default) Don't use “virtual IP” addresses for IPv6 addresses. <p>Example:</p> <pre>ipv6_virtual_ips = enable</pre> <p>When enabled, the virtual IP address can be specified via the <code>virtual_ip_sc_eth*</code> parameter, for example:</p> <pre>virtual_ip_sc_eth2 = fd00::a0dc:1</pre>	
kms_key_id	AWS	<p>Identifier of the AWS KMS key for Amazon EBS disk encryption. You can specify the key via one of the following:</p> <ul style="list-style-type: none"> ■ Key ID ■ Key alias ■ Key ARN ■ Alias ARN <p>Example:</p> <pre>kms-key-id = arn:aws:kms:us-east-1:012345678910:1234abcd-12ab6ef-1234567890ab</pre>	Rebuild
main_subnet_cidr	Azure	<p>Defines the CIDR for the “main” subnet. This can be used to overcome Stack Manager’s lack of permissions to read current subnet configuration.</p> <p>Syntax: Subnet IP / Prefix Length.</p> <p>Example:</p> <pre>mian_subnet_cidr = 10.2.3.0/24</pre>	Rebuild
main_nsg_id	Azure	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to the <code>cluster_nsg_id</code> parameter.</p> <p>When modifying the security group, make sure that it includes rules that enable Stack Manager to access deployed instances via the HTTPS protocol (TCP/443).</p> <p>See also the <code>media_nsg_id</code>, <code>oam_nsg_id</code> and <code>signaling_nsg_id</code> parameters.</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
manage_via_https	All	<p>Defines the protocol used by Stack Manager when connecting to the deployed stack's management interface.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Use HTTPS. ■ disable: Use HTTP. <p>Example:</p> <pre>manage_via_https = disable</pre>	Instant
media_nsg_id	AWS, Azure	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to the <code>cluster_nsg_id</code> parameter.</p> <p>See also the <code>main_nsg_id</code>, <code>oam_nsg_id</code> and <code>signaling_nsg_id</code> parameters.</p>	Update
mc_additional_ips	All	<p>Defines the network interface names of Media Components for which additional private IP addresses are allocated and optionally, the number of corresponding IP addresses.</p> <p>Refer to the <code>sc_additional_ips</code> parameter for more information.</p> <p>After stack creation, use the MC Additional IPs parameter in the Modify dialog to change the parameter value.</p>	Update
mc_image_id	AWS, Azure	<p>Defines the local image for Media Components (instead of the Marketplace image).</p> <p>Refer to the <code>sc_image_id</code> parameter for more information.</p> <p>After stack creation, use the MC Image ID parameter in the Modify dialog to change the parameter value.</p> <p>See also the <code>sbc_image_id</code> parameter.</p>	Update
mc_image_url	AWS, Azure, Google	<p>Defines the URL that contains a plain-text file with the name of the local image for Media Components (instead of the Marketplace image).</p> <p>Refer to the <code>sc_image_url</code> parameter for more information.</p> <p>See also the <code>sbc_image_url</code> parameter.</p>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
mc_ini_params	All	Defines additional configuration parameters (in INI file format) for Media Components during stack creation / rebuild. Refer to the <code>sc_ini_params</code> parameter for more information.	Rebuild
mc_ipv6_ips	AWS, Azure	Defines the network interface names of Media Components for which IPv6 addresses are allocated. Refer to the <code>sc_ipv6_ips</code> parameter for more information.	Update
mc_max_pps_limit	All	Defines the Media Component's maximum forwarding capacity (in packets per second). Supported values: <ul style="list-style-type: none"> ■ auto: (Default) Stack Manager automatically configures Media Component forwarding capacity based on the cloud environment and instance type used ■ <number>: Manually defines the Media Component forwarding capacity. The specified number imposes a limit on the number of sessions supported by the Media Component according to the following formula: $num_of_sessions = max_pps_limit * 9$ Example: <pre>mc_max_pps_limit = 280</pre>	Rebuild
mc_public_ips	All	Defines the network interfaces of Media Components for which public IP addresses are allocated and optionally, the number of corresponding IP addresses. Refer to the <code>sc_public_ips</code> parameter for more information.	Update
mc_tags	AWS, Azure, Google	Defines tags that are assigned to the following Media Components' resources: <ul style="list-style-type: none"> ■ AWS: Tags are assigned to EC2 instance, volume, network interfaces and Elastic IPs. ■ Azure and Google Cloud: Tags are assigned to VM instances. Refer to the <code>sc_tags</code> parameter for more information. See also the <code>tags</code> and <code>common_tags</code> parameters.	Azure: Update Other: Rebuild
mc_user_data	All	Defines additional cloud-init configuration parameters for Media Components. Refer to the <code>sc_user_data</code> parameter for more information.	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
<code>nat_ip_sc_ethX</code> , <code>nat_ip_mc-X_ethY</code>	AWS, Azure, Google	<p>Defines custom entries in the NAT Translation table of Signaling Components and Media Components.</p> <p>The parameter is useful when Public IP addresses are provided by external firewall / NAT gateway, and not attached directly to resources deployed by Stack Manager.</p> <p>Syntax: comma-separated list of IP addresses</p> <p>Example:</p> <pre>nat_ip_sc_eth1 = 10.1.1.5 nat_ip_mc-1_eth1 = 10.1.1.10 nat_ip_mc-2_eth1 = 10.1.1.11</pre> <p>Note:</p> <ul style="list-style-type: none"> ■ If a single IP address is specified, it's configured: <ul style="list-style-type: none"> • Azure, Google Cloud and multi-zone AWS deployments: For primary address, e.g., "eth1". • Single-zone AWS deployments: For first "usable" address, e.g., "eth1:1" on Signaling Components and "eth1" on Media Components. ■ If multiple IP addresses are specified, they are configured for additional / secondary addresses. You can leave irrelevant list elements empty, e.g., the following: <pre>nat_ip_mc-1_eth2 = ,10.1.1.10,10.1.1.11</pre> configures NAT translation for mc-1 on "eth2:1" and "eth2:2". 	Update
<code>nsg_id_mc_ethX</code>	AWS, Azure	<p>Defines the name of the existing Network Security Group (NSG) for a specific Media Components' network interface.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to the <code>nsg_id_sc_ethX</code> parameter.</p>	Update
<code>nsg_id_sc_ethX</code>	AWS, Azure	<p>Defines the name of the existing Network Security Group (NSG) for a specific Signaling Components' network interface.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
		<p>When modifying the security group that contains “management” rules, make sure that it includes rules that enable Stack Manager to access deployed instances via the HTTPS protocol (TCP/443).</p> <p>Syntax:</p> <ul style="list-style-type: none"> ■ AWS: Security Group ID. Multiple groups can be specified as a comma-separated string, e.g.,: <pre>nsg_id_sc_eth1 = sg-123,sg-345</pre> ■ Azure: Resource Group name / NSG name, e.g.,: <pre>nsg_id_mc_eth2 = rg1/cluster-nsg</pre> 	
oam_nsg_id	AWS	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to the <code>cluster_nsg_id</code> parameter.</p> <p>When modifying the security group, make sure that it includes rules that enable Stack Manager to access deployed instances via the HTTPS protocol (TCP/443).</p> <p>See also the <code>main_nsg_id</code>, <code>media_nsg_id</code> and <code>signaling_nsg_id</code> parameters.</p>	Update
oam_ip	AWS, Azure, Google	<p>Defines an IP address on Signaling Components for management traffic (Web, SSH, SNMP).</p> <p>Syntax:</p> <ul style="list-style-type: none"> ■ “default”: Use primary IP address on the “main” network interface for management traffic. In Azure and Google Cloud, Signaling Components reside behind Load Balancer and management traffic should be sent to the frontend addresses on the corresponding Load Balancer. ■ “internal”: Implies the following configuration: <pre>SC Public IPs: main SC Additional IPs: main oam_ip = internal</pre> <ul style="list-style-type: none"> • Azure and Google Cloud: Creates two IP addresses on the “main” network interface. The primary one is placed behind Public Load Balancer and is used 	Update (Azure) or Rebuild (AWS, Google)

Parameter	Applicable Environment	Description	Apply Mode
		<p>for VoIP traffic. The secondary one is placed behind Internal Load Balancer and is used for management traffic.</p> <ul style="list-style-type: none"> AWS: Creates an additional private IP address on the “main” network interface and uses it for management traffic. “additional1” / “additional2”: (Google Cloud only) Use the primary IP address on the network interface connected to the “Additional 1” / “Additional 2” subnet correspondingly for management traffic. <p>During stack creation (via Web interface), this parameter is configured via the Use private IP address for management parameter in the Create dialog.</p>	
proximity_placement_group_vm_sizes	Azure	<p>Defines the optional “intent” parameter of proximity placement groups created as part of stack deployment. Refer to https://learn.microsoft.com/en-us/azure/virtual-machines/co-location for details.</p> <p>Syntax: comma-separated list of VM sizes.</p> <p>Examples:</p> <pre>proximity_placement_group_vm_sizes = Standard_D2ds_v5,Standard_D4ds_v5</pre> <p>Typically used together with the <code>proximity_placement_group_vm_zone</code> parameter (see below).</p>	Can't be modified after stack creation
proximity_placement_group_vm_zone	Azure	<p>Defines the optional “zone” parameter of proximity placement groups created as part of stack deployment. Refer to https://learn.microsoft.com/en-us/azure/virtual-machines/co-location for details.</p> <p>Enables deployment of Mediant CE into an availability set located in the specific availability zone.</p> <p>Examples:</p> <pre>proximity_placement_group_vm_zone = 1</pre> <p>Due to limitations of Azure API, if you specify this parameter you must also specify the <code>proximity_placement_group_vm_sizes</code> parameter.</p>	Can't be modified after stack creation
private_ip_sc-X_eth* , private_ip_mc-X_eth*	AWS, Azure,	Defines pre-defined private IP addresses. See Section 3.25.3 for more information.	Update

Parameter	Applicable Environment	Description	Apply Mode
	Google	See also the <code>ipv6_ip_*</code> and <code>public_ip_*</code> parameters.	
<code>public_ip_sc_eth*</code> , <code>public_ip_mc-X_eth*</code>	AWS, Azure, Google	Defines pre-defined public addresses. See Section 3.25.3 for more information. See also the <code>ipv6_ip_*</code> and <code>private_ip_*</code> parameters.	Update
<code>public_ip_prefix_sc</code> , <code>public_ip_prefix_mc</code> , <code>public_ip_prefix</code>	Azure	Defines a Public IP Prefix from which public IP addresses for Signaling Components or Media Components are allocated. Syntax: comma-separated list of elements: <code><rg>/<name>/<count></code> Where: <ul style="list-style-type: none"> ■ <code><rg></code> is the resource group name. ■ <code><name></code> is the name of the Public IP Prefix. ■ <code><count></code> is the number of IP addresses to be allocated from the specific prefix. Use Components <code>public_ip_prefix</code> parameter to specify the same prefix for both Signaling Components and Media Components. Examples: <pre>public_ip_prefix = rg1/prefix1/16 public_ip_prefix = rg1/prefix1/4,rg2/prefix2/8</pre>	Update
<code>remote_interfaces</code>	All	When set to “refresh”, it triggers a refresh of the Remote Media Interfaces table upon the next “update” action. Example: <pre>remote_interfaces = refresh</pre>	Update
<code>resource_group</code>	Azure	Defines the name of the existing Resource Group. If not empty, stack resources are deployed into this Resource Group instead of creating a new one. The Resource Group must be empty prior to stack creation. Example: <pre>resource_group = SbcGroup1</pre>	Can't be modified after stack creation
<code>sbc_image_id</code>	AWS, Azure	Defines the local image for Signaling Components and Media Components (instead of the Marketplace image). Contrary to the <code>sc_image_id</code> and <code>mc_image_id</code> parameters, this parameter is applied via the Rebuild (and not Update) operation. For Azure, specify the resource group name followed by the image name, e.g.,:	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
		<pre>sc_image_id = rg1/image1</pre> <p>For AWS, specify the AMI ID, e.g.,:</p> <pre>sc_image_id = ami-9a50cff5</pre>	
sbc_image_url	AWS, Azure, Google	<p>Defines the URL that contains a plain-text file with the name of the local image for Signaling Components and Media Components (instead of the Marketplace image).</p> <p>Example:</p> <pre>sc_image_url = https://company.com/123456</pre> <p>See also the <code>mc_image_url</code> and <code>sc_image_url</code> parameter.</p>	Rebuild
sc_additional_ips	All	<p>Defines the network interface names of Signaling Components for which additional private IP addresses are allocated and optionally, the number of corresponding IP addresses.</p> <p>Additional IP addresses are allocated <i>on top of</i> any private IP addresses created by Stack Manager by default and/or due to the public IP addresses assigned to the specific network interface.</p> <p>Syntax: comma-separated list of subnet names: "main", "additional1", "additional2", etc.; e.g.:</p> <pre>sc_additional_ips = additional1,additional2</pre> <p>You can also use "all" to specify all subnets.</p> <p>If more than one additional private IP address is required on the specific network interface, this can be specified as "<name>:<num>", where <num> is the total number of additional private IP addresses to be created; e.g.,:</p> <pre>sc_additional_ips = additional1:2</pre> <p>Alternatively, you can specify interface names "ethX" instead of subnet names, e.g.,:</p> <pre>sc_additional_ips = eth1,eth2</pre> <p>Notes:</p> <ul style="list-style-type: none"> ■ On Azure, for Signaling Components, internal IP addresses are allocated on Internal Load Balancer, and corresponding network interfaces are placed behind it. ■ On Google Cloud, for Signaling Components, internal IP addresses are allocated on Network Load Balancer, and the "main" interface is placed behind it. 	Update

Parameter	Applicable Environment	Description	Apply Mode
		<p>After stack creation, use the SC Additional IPs parameter in the Modify dialog to change the parameter value.</p> <p>See also the <code>mc_additional_ips</code> parameter.</p>	
<code>sc_ha_mode</code>	All	<p>Defines the number of Signaling Components.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Two Signaling Components are created and operate in 1+1 HA mode. ■ disable: One Signaling Component is created. <p>Example:</p> <pre>sc_ha_mode = disable</pre>	<p>Azure: Update</p> <p>Other: Rebuild</p>
<code>sc_image_id</code>	AWS, Azure	<p>Defines the local image for Signaling Components (instead of the Marketplace image).</p> <p>For Azure, specify the resource group name followed by the image name, e.g.,:</p> <pre>sc_image_id = rg1/image1</pre> <p>For AWS, specify the AMI ID, e.g.,:</p> <pre>sc_image_id = ami-9a50cff5</pre> <p>After stack creation, use the SC Image ID parameter in the Modify dialog to change the parameter value.</p> <p>See also the <code>sbc_image_id</code> and <code>mc_image_id</code> parameters.</p>	Update
<code>sc_image_url</code>	AWS, Azure, Google	<p>Defines the URL that contains a plain-text file with the name of the local image for Signaling Components (instead of the Marketplace image).</p> <p>Example:</p> <pre>sc_image_url = https://company.com/123456</pre> <p>See also the <code>mc_image_url</code> and <code>sbc_image_url</code> parameter.</p>	Rebuild
<code>sc_ini_params</code>	All	<p>Defines additional configuration parameters (in INI file format) for Signaling Components during stack creation / rebuild.</p> <p>Syntax: a single line with <code>\n</code> as line delimiter, e.g.,:</p> <pre>ini_incremental = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</pre> <p>Specified configuration is applied via cloud-init mechanism and therefore, is limited by instance user-data size (for example, in AWS it's limited by 16 KB). For long configurations, use the <code>ini_incremental</code> parameter instead.</p>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
		See also the <code>mc_ini_params</code> parameter.	
<code>sc_ipv6_ips</code>	AWS, Azure	<p>Defines the network interface names of Signaling Components for which IPv6 addresses are allocated.</p> <p>For AWS, IPv6 addresses are a <i>new</i> type of addresses that are globally routable. They are assigned <i>in addition</i> to IPv4 addresses specified via the <code>sc_public_ips</code> and <code>sc_additional_ips</code> parameters. If the interface is connected to a subnet that has both IPv4 and IPv6 address ranges, the primary address will be IPv4, and IPv6 will be assigned as a secondary address with the 'ethX:100' name. If the interface is connected to IPv6-only subnet (i.e., subnet that lacks IPv4 address range), only IPv6 addresses are allocated and assigned with regular 'ethX' names. Pre-defined IPv6 addresses can be specified via the <code>ipv6_ip_*</code> parameters.</p> <p>For Azure, IPv6 is a <i>property</i> of normal internal and public addresses. Therefore, this parameter does not add new addresses, but changes the type of those that are specified via the <code>sc_public_ips</code> or <code>sc_additional_ips</code> parameters. Primary addresses can't be IPv6 and therefore, this parameter by default configures IPv6 on the "1st secondary" (second) IP address. For example, the following creates two public addresses – IPv4 and IPv6 – on Signaling Components' interfaces connected to "additional 1" subnet:</p> <pre>SC Public IPs: additional1:2 sc_ipv6_ips = additional1</pre> <p>You can specify secondary address index (1="1st secondary", 2="2nd secondary", and so on) to configure IPv6 on a different secondary address. For example, the following configures IPv6 on the "2nd secondary" (third) IP address of Signaling Components' interfaces connected to "additional 1" subnet:</p> <pre>SC Public IPs: additional1 SC Additional IPs: additional1:2 sc_ipv6_ips = additional1:2</pre> <p>Pre-defined IPv6 addresses can be specified via the <code>private_ip_*</code> and <code>public_ip_*</code> parameters.</p> <p>Syntax: comma-separated list of subnet names: "main", "additional1", "additional2", etc., for example:</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
		<pre>sc_ipv6_ips = additional1,additional2</pre> <p>Alternatively, you can specify interface names “ethX” instead of subnet names, e.g.,:</p> <pre>sc_ipv6_ips = eth1,eth2</pre> <p>Notes:</p> <ul style="list-style-type: none"> ■ Management is always done via IPv4 addresses. ■ For multi-zone deployment in AWS, “virtual IPs” are disabled by default for IPv6 addresses because they work only within the subnet and therefore, have very limited application. Use the <code>ipv6_virtual_ips</code> parameter to change this behavior. 	
sc_public_ips	All	<p>Defines the network interfaces of Signaling Components for which public IP addresses are allocated and optionally, the number of corresponding IP addresses.</p> <p>During stack creation (via Web interface), Stack Manager lets you specify which subnets (and corresponding network interfaces) are assigned with public (Elastic) IP addresses using the Public IPs parameter in the Networking section.</p> <p>When the <code>sc_public_ips</code> and/or <code>mc_public_ips</code> advanced configuration parameters are specified, they override any value configured by the Public IPs parameter. Typically, you use these parameters when:</p> <ul style="list-style-type: none"> ■ You need to create multiple IP addresses on the same network interface. ■ You need to configure IP addresses differently for Signaling Components and Media Components. <p>Syntax: comma-separated list of subnet names: “main”, “additional1”, “additional2”, etc.; e.g.,:</p> <pre>sc_public_ips = additional1,additional2</pre> <p>You can also use “all” to specify all subnets.</p> <p>If more than one public IP address is required on the specific network interface, this can be specified as “<name>:<num>”, where <num> is the total number of public IP addresses to be created; e.g.,:</p> <pre>sc_public_ips = additional1:2</pre> <p>Alternatively, you can specify interface names “ethX” instead of subnet names, e.g.,:</p> <pre>sc_public_ips = eth1,eth2</pre>	Update

Parameter	Applicable Environment	Description	Apply Mode
		<p>Notes:</p> <ul style="list-style-type: none"> On Azure, for Signaling Components, public IP addresses are allocated on Public Load Balancer, and corresponding network interfaces are placed behind it. On Google Cloud, for Signaling Components, public IP addresses are supported only on the “main” interface. They are allocated on Network Load Balancer, and the “main” interface is placed behind it. Stack Manager implicitly creates all private IP addresses required for public IP address assignment. <p>After stack creation, use the SC Public IPs parameter in the Modify dialog to change the parameter value.</p> <p>See also the <code>mc_public_ips</code> parameter.</p>	
<code>sc_tags</code>	AWS, Azure, Google	<p>Defines tags assigned to the following Signaling Components’ resources:</p> <ul style="list-style-type: none"> AWS: Tags are assigned to EC2 instance, volume, network interfaces and Elastic IPs. Azure and Google Cloud: Tags are assigned to VM instances. <p>Syntax:</p> <ul style="list-style-type: none"> AWS or Azure: comma-separated list of name=value pairs, e.g.,: <pre>sc_tags = type=sbc,role=sc</pre> Google: comma-separated list of tags, e.g.,: <pre>sc_tags = sbc,sc</pre> <p>See also the <code>tags</code>, <code>mc_tags</code>, and <code>common_tags</code> parameters.</p>	Azure: Update Other: Rebuild
<code>sc_user_data</code>	All	<p>Defines additional cloud-init configuration parameters for Signaling Components.</p> <p>Syntax: a single line with <code>\n</code> as line delimiter.</p> <p>Example:</p> <pre>sc_user_data = #customer-id\n123456\n#license-key\nokRTr5top...</pre>	Rebuild
<code>sc1_ha_name</code> , <code>sc2_ha_name</code>	All	<p>Defines the name of the first / second Signaling Component in the SBC’s Web interface’s Monitor page.</p> <p>Example:</p> <pre>sc1_ha_name = sc-1 sc2_ha_name = sc-2</pre>	Rebuild
<code>shelve_delete_ips</code>	AWS,	<p>Defines whether public IP addresses are deleted during “shelve” operation. It overrides the global</p>	Instant

Parameter	Applicable Environment	Description	Apply Mode
	Azure	<p>“Delete Public IPs During Shelve” configuration parameter.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: Delete public IPs during “shelve” operation. ■ disable: Don't delete public IPs during “shelve” operation. ■ empty: (Default) Use global configuration parameter. <p>Example:</p> <pre>shelve_delete_ips = enable</pre>	
signaling_nsg_id	AWS, Azure	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant CE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to <code>cluster_nsg_id</code> parameter.</p> <p>See also the <code>main_nsg_id</code>, <code>media_nsg_id</code> and <code>oam_nsg_id</code> parameters.</p>	Update
spot_instances	Azure	<p>Enables the use of Azure Spot instances for testing environments. Note that Spot instances might be abruptly stopped and therefore, should never be used in production environment.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: Use Spot instances. ■ disable: (Default) Use regular instances. <p>Example:</p> <pre>spot_instances = enable</pre>	Can't be modified after stack creation
storage_account_type	Azure	<p>Defines the storage account type for managed disks.</p> <p>Valid values include:</p> <ul style="list-style-type: none"> ■ Standard_LRS ■ Premium_LRS ■ StandardSSD_LRS <p>Example:</p> <pre>storage_account_type = Premium_LRS</pre>	Rebuild
tags	Azure	<p>Defines tags assigned to all created stack resources.</p> <p>Syntax: comma-separated list of name=value pairs.</p> <p>Example:</p>	<p>Azure: Update</p> <p>Other: Rebuild</p>

Parameter	Applicable Environment	Description	Apply Mode
		<pre>tags = type=sbc,role=sc</pre> <p>You can also use the <code>sc_tags</code> and <code>mc_tags</code> parameters to define <i>additional</i> tags that will be assigned to respective VMs only.</p>	
update_needed	All	<p>When set to “reset”, it turns off the <code>update_needed</code> flag without applying any changes upon the next “update” action.</p> <p>Example:</p> <pre>update_needed = reset</pre>	Update
use_availability_set	Azure	<p>Defines whether an availability set is created for ‘availability set’ deployment topology.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Create availability set. ■ disable: Don't create availability set. Virtual machines are deployed without any redundancy constraints. <p>Example:</p> <pre>use_availability_set = disable</pre>	Can't be modified after stack creation
use_placement_group	AWS	<p>Defines if Mediant CE components are deployed in the placement group.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Use placement group. ■ disable: Don't use placement group. <p>Example:</p> <pre>use_placement_group = disable</pre>	Can't be modified after stack creation
use_proximity_placement_group	Azure	<p>Defines if Mediant CE components are deployed in the proximity placement group.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Use proximity placement group. ■ disable: Don't use proximity placement group. <p>Example:</p> <pre>use_proximity_placement_group = disable</pre>	Can't be modified after stack creation
virtual_ip_sc_eth*	AWS	<p>Applicable to multi-zone AWS deployments.</p> <p>Defines virtual IP addresses for communication in the VPC or via the Transit Gateway.</p> <p>Example:</p> <pre>virtual_ip_sc_eth2 = 10.5.1.10</pre>	Update
virtual_ips	AWS	<p>Defines whether “virtual IPs” are used for IPv4 addresses in multi-zone AWS deployments.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable (default) 	Update

Parameter	Applicable Environment	Description	Apply Mode
		<ul style="list-style-type: none">■ disable Example: <code>virtual_ips = disable</code>	
volume_type	AWS	Defines the volume type for EBS disks. Valid values include: <ul style="list-style-type: none">■ gp2■ gp3■ io1■ io2■ sc1■ sc2■ standard Example: <code>volume_type = gp3</code>	Rebuild

3.8.11.2 Advanced Configuration for Mediant VE

The following table describes advanced parameters available for Mediant VE.

Table 3-6: Advanced Parameters Description

Parameter	Applicable Environment	Description	Apply Mode
accelerated_networking	Azure	<p>Enables accelerated networking on D_v2, Dds_v3 and Dds_v4 instances.</p> <p>Note that Dds_v5 instances always have accelerated networking enabled and therefore, this parameter is not applicable.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ disable (Default) ■ enable <p>Example:</p> <pre>accelerated_networking = enable</pre>	Update
additional_ips	All	<p>Defines the network interface names for which additional private IP addresses are allocated and optionally, the number of corresponding IP addresses.</p> <p>Additional IP addresses are allocated <i>on top</i> 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.</p> <p>Syntax: comma-separated list of subnet names: "main", "additional1", "additional2"; e.g.,:</p> <pre>additional_ips = additional1,additional2</pre> <p>You can also use "all" to specify all subnets.</p> <p>If more than one additional private IP address is required on the specific network interface, this can be specified as "<name>:<num>", where <num> is the total number of additional private IP addresses to create; e.g.,:</p> <pre>additional_ips = additional1:2</pre> <p>Alternatively, you can specify interface names "ethX" instead of subnet names, e.g.,:</p> <pre>additional_ips = eth1,eth2</pre> <p>Notes:</p> <ul style="list-style-type: none"> ■ For HA deployment in Azure, the <code>additional_ips</code> parameter specifies "logical" IP addresses. For each such "logical" address, two IP addresses are allocated on the VM - one is placed behind Internal Load Balancer and is used for signaling (SIP) traffic and another one is used for media (RTP) traffic. ■ For HA deployment in Google Cloud, internal IP addresses are allocated on Network Load Balancer and the "main" interface is placed behind it. 	Update

Parameter	Applicable Environment	Description	Apply Mode
		<ul style="list-style-type: none"> After stack creation, use the Additional IPs parameter in the Modify dialog to change the parameter value. 	
additional_route_tables	AWS	<p>Applicable to multi-zone AWS deployments. Defines additional route tables that should be updated with virtual IP addresses.</p> <p>Syntax: comma-separated list of <interface name>:<route table ID>. Multiple route table IDs can be specified using pipe () delimiter.</p> <p>Example:</p> <pre>additional_route_tables = eth1:rtb-123,eth2:rtb-567 rtb-890</pre>	Update
additional3_subnet_id , additional4_subnet_id , additional5_subnet_id , additional6_subnet_id	AWS, Azure	<p>Defines subnet IDs for Additional 3 to Additional 6 subnets.</p> <p>For Azure, specify the subnet name, e.g.,:</p> <pre>additional3_subnet_id = voip3</pre> <p>For AWS, specify the subnet ID, c:</p> <pre>additional3_subnet_id = subnet-12345</pre> <p>Note: Subnet IDs that are currently “in use” can't be modified. If you want to change the subnet ID of an existing network interface, first reduce the number of network interfaces, update the corresponding subnet ID, and then restore the number of interfaces.</p>	Instant
app_insights	Azure	<p>Defines the type of data that is reported to Azure Application Insights.</p> <p>Supported values:</p> <ul style="list-style-type: none"> disable: (Default) Don't report any data for this stack. enable: Report alarms and metrics (PMs). alarms: Report alarms only. metrics: Report metrics (PMs) only. <p>Example:</p> <pre>app_insights = enable</pre> <p>See Section 7.3, Integration with Azure Application Insights for more information.</p>	Instant
auto_shelve_time	All	<p>Defines the time of day when the stack is automatically “shelved” (i.e., all VMs are stopped and unnecessary resources, for example, load balancers are deleted). If defined, it overrides the global Auto Shelve Time configuration parameter.</p> <p>Supported syntax:</p> <ul style="list-style-type: none"> 08:00: Time of day (24h). 1/08:00: Weekday (0 is Sunday, 1 is Monday, 2 is Tuesday, and so on) and time. 0,1,2/08:00: Multiple weekdays and time. 	Instant

Parameter	Applicable Environment	Description	Apply Mode
		<ul style="list-style-type: none"> ■ 0-5/08:00: Range of weekdays and time. ■ 0,1/08:00 2-4/09:00: Multiple statements. Example: <pre>auto_shelve_time = 08:00</pre>	
auto_start_time	All	Defines the time of day when the stack is automatically started. If defined, it overrides the global Auto Start Time configuration parameter. Syntax is identical to the <code>auto_shelve_time</code> parameter. Example: <pre>auto_start_time = 08:00</pre>	Instant
auto_stop_time	All	Defines the time of day when the stack is automatically stopped. If defined, it overrides the global Auto Stop Time configuration parameter. Syntax is identical to the <code>auto_shelve_time</code> parameter. Example: <pre>auto_stop_time = 22:00</pre>	Instant
availability_zones	Azure	Defines availability zones where Mediant VE components will be deployed. Syntax: <ul style="list-style-type: none"> ■ Two availability zone names separated by a comma. ■ "none" – components are deployed into an availability set. Example: <pre>availability_zones = 1,2</pre>	Update
ha_nlb	AWS	Enables the use of AWS Network Load Balancer for 1+1 HA implementation of Signaling Components. Supported values: <ul style="list-style-type: none"> ■ internal – use internal NLB instead of Virtual IPs ■ public – use public NLB instead of Elastic IPs ■ all – use internal / public NLB instead of Virtual / Elastic IPs ■ oam – use internal instead of Virtual IPs for management traffic only Example: <pre>ha_nlb = internal</pre>	Can't be modified after stack creation
image_id	AWS, Azure	Defines the local image (instead of the Marketplace image). For Azure, specify the resource group name followed by the image name, e.g.,: <pre>image_id = rg1/image1</pre> For AWS, specify the AMI ID, e.g.,: <pre>image_id = ami-9a50cff5</pre>	Update

Parameter	Applicable Environment	Description	Apply Mode
		After stack creation, use the Image ID parameter in the Modify dialog to change the parameter value.	
image_url	All	Defines the URL that contains a plain-text file with the name of the local image (instead of the Marketplace image). Example: <pre>image_url = https:// company.com/123456</pre>	Rebuild
ini_incremental	All	Defines additional configuration parameters (in INI file format) that's applied during stack creation / rebuild. Syntax: a single line with \n as line delimiter, e.g.,: <pre>ini_incremental = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</pre> Specified configuration is applied via REST API and therefore, has no size limit. Therefore, this parameter is preferred over the <code>ini_params</code> parameter that is applied via the cloud-init mechanism and therefore, is limited by instance user-data size.	Rebuild
imds	AWS	Defines the version of the AWS meta-data instance for the deployed EC2 instances. Supported values: <ul style="list-style-type: none"> any: Allow both IMDSv1 and IMDSv2. v2: (Default) Enforce IMDSv2. Example: <pre>imds = v2</pre>	Update
ini_params	All	Defines additional configuration parameters (in INI file format) that's applied during stack creation / rebuild. Syntax: a single line with \n as line delimiter, e.g.,: <pre>ini_incremental = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</pre> Specified configuration is applied via the cloud-init mechanism and is therefore limited by instance user-data size (for example, in AWS it's limited by 16 KB). For long configurations, use the <code>ini_incremental</code> parameter instead.	Rebuild
ipv6_ip_sbc-X_eth*	AWS	Defines pre-defined IPv6 addresses. See also the <code>private_ip_*</code> and <code>public_ip_*</code> parameters.	Update
ipv6_ips	AWS, Azure	Defines the network interface names for which IPv6 addresses are allocated. In AWS , IPv6 addresses are a <i>new</i> type of addresses that are globally routable. They are assigned <i>in addition</i> to IPv4 addresses specified via the <code>public_ips</code> and <code>additional_ips</code> parameters.	Update

Parameter	Applicable Environment	Description	Apply Mode
		<p>If the interface is connected to a subnet that has both IPv4 and IPv6 address ranges, the primary address will be IPv4, and IPv6 will be assigned as the secondary address with the 'ethX:100' name. If the interface is connected to an IPv6-only subnet (i.e., subnet lacks IPv4 address range), only IPv6 addresses are allocated and assigned with regular 'ethX' names. Pre-defined IPv6 addresses can be specified via <code>ipv6_ip_*</code> parameters.</p> <p>In Azure, IPv6 is a <i>property</i> of normal internal and public addresses. Therefore, this parameter does not add new addresses, but changes the type of those that are specified via the <code>public_ips</code> or <code>additional_ips</code> parameters. Primary addresses cannot be IPv6 and therefore, this parameter by default configures IPv6 on the "1st secondary" (second) IP address. For example, the following creates two public addresses – IPv4 and IPv6 – on an interface connected to the "additional 1" subnet:</p> <pre>Public IPs: additional1:2 ipv6_ips = additional1</pre> <p>You can specify the secondary address index (1 = "1st secondary", 2 = "2nd secondary", and so on) to configure IPv6 on a different secondary address. For example, the following configures IPv6 on the "2nd secondary" (third) IP address of an interface connected to the "additional 1" subnet:</p> <pre>Public IPs: additional1 Additional IPs: additional1:2 ipv6_ips = additional1:2</pre> <p>Pre-defined IPv6 addresses can be specified via the <code>private_ip_*</code> and <code>public_ip_*</code> parameters.</p> <p>Syntax: comma-separated list of subnet names: "main", "additional1", "additional2", etc.. Example:</p> <pre>ipv6_ips = additional1,additional2</pre> <p>Alternatively, you can specify interface names "ethX" instead of subnet names, e.g.,:</p> <pre>ipv6_ips = eth1,eth2</pre> <p>Notes:</p> <ul style="list-style-type: none"> ■ Management is always done via IPv4 addresses. ■ For HA deployment on Azure, an IPv6 address must be the last address on the specific interface and a "paired" address is not created for it. You should use the same IPv6 address for both signaling and media streams and configure port-based NAT translation rules. 	

Parameter	Applicable Environment	Description	Apply Mode
		<ul style="list-style-type: none"> For multi-zone HA deployment on AWS, “virtual IPs” are disabled by default for IPv6 addresses, because they work only within the subnet and therefore, have very limited application. Use the <code>ipv6_virtual_ips</code> parameter to change this behavior. 	
<code>ipv6_virtual_ips</code>	AWS	<p>Defines whether “virtual IPs” are used for IPv6 addresses in multi-zone AWS deployments.</p> <p>“Virtual IPs” apply only to internal traffic (within the subnet or via Transit Gateway) and therefore, have very limited application. Correspondingly, this parameter defaults to “disable” and it's recommended to use AWS Load Balancer or DNS-based methods for traffic distribution.</p> <p>Supported values:</p> <ul style="list-style-type: none"> enable: Use “virtual IP” addresses for IPv6 addresses. disable: (default) Don't use “virtual IP” addresses for IPv6 addresses. <p>Example:</p> <pre>ipv6_virtual_ips = enable</pre> <p>When enabled, the virtual IP address can be specified via the <code>virtual_ip_eth*</code> parameter, for example:</p> <pre>virtual_ip_eth2 = fd00::a0dc:1</pre>	Update
<code>kms_key_id</code>	AWS	<p>Identifier of the AWS KMS key for Amazon EBS disk encryption. You can specify the key via one of the following:</p> <ul style="list-style-type: none"> Key ID Key alias Key ARN Alias ARN <p>Example:</p> <pre>kms-key-id = arn:aws:kms:us-east-1:012345678910:1234abcd-12ab6ef-1234567890ab</pre>	Rebuild
<code>main_nsg_id</code>	Azure	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant VE for Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to <code>ha_nsg_id</code> parameter.</p> <p>When modifying the security group, make sure that it includes rules that enable Stack Manager to access deployed instances via the HTTPS protocol (TCP/443). See also the <code>voip_nsg_id</code> parameter.</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
manage_via_https	All	<p>Defines the protocol used by Stack Manager when connecting to the deployed stack's management interface.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Use HTTPS protocol. ■ disable: Use HTTP protocol. <p>Example:</p> <pre>manage_via_https = disable</pre>	Instant
media_nsg_id	AWS	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant VE for AWS Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to the <code>ha_nsg_id</code> parameter.</p> <p>See also the <code>oam_nsg_id</code> and <code>signaling_nsg_id</code> parameters.</p>	Update
nsg_id_ethX	AWS, Azure	<p>Defines the name of the existing Network Security Group (NSG) for a specific network interface.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant VE for AWS / Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>When modifying the security group that contains "management" rules, make sure that it includes rules that enable Stack Manager to access deployed instances via the HTTPS protocol (TCP/443).</p> <p>Syntax:</p> <ul style="list-style-type: none"> ■ AWS: Security Group ID. Multiple groups can be specified as a comma-separated string, e.g.,: <pre>nsg_id_eth1 = sg-123,sg-345</pre> ■ Azure: Resource Group name / NSG name, e.g.,: <pre>nsg_id_eth2 = rg1/cluster-nsg</pre> 	Update
oam_ip	All	<p>Defines an IP address for management traffic (Web, SSH, and SNMP). Applicable for HA deployments only.</p> <p>Syntax:</p> <ul style="list-style-type: none"> ■ "default": Use the primary IP address on the "main" network interface for management traffic. For HA deployments on Azure and Google Cloud, VMs reside behind Load Balancer and management traffic should be sent to the frontend addresses on the corresponding Load Balancer. ■ "internal": Implies the following configuration: <pre>Public IPs: main Additional IPs: main oam_ip = internal</pre> 	Update (Azure) or Rebuild (AWS, Google)

Parameter	Applicable Environment	Description	Apply Mode
		<ul style="list-style-type: none"> Azure and Google Cloud: Creates two IP addresses on the “main” network interface. The primary one is placed behind Public Load Balancer and is used for VoIP traffic; the secondary one is placed behind Internal Load Balancer and is used for management traffic. AWS: Creates an additional private IP address on the “main” network interface and uses it for management traffic. ■ “additional1” / “additional2”: (Google Cloud only) Use the primary IP address on the network interface connected to “Additional 1” / “Additional 2” subnet respectively, for management traffic. <p>During stack creation (via Web interface), this parameter is configured via the Use private IP address for management parameter in the Create dialog.</p>	
proximity_placement_group_vm_sizes	Azure	<p>Defines the optional “intent” parameter of proximity placement groups created as part of the stack deployment. Refer to https://learn.microsoft.com/en-us/azure/virtual-machines/co-location for details.</p> <p>Syntax: comma-separated list of VM sizes.</p> <p>Example:</p> <pre>proximity_placement_group_vm_sizes = Standard_D2ds_v5,Standard_D4ds_v5</pre> <p>Typically used together with the <code>proximity_placement_group_vm_zone</code> parameter (see below).</p>	Can't be modified after stack creation
proximity_placement_group_vm_zone	Azure	<p>Defines the optional “zone” parameter of proximity placement groups created as part of the stack deployment. Refer to https://learn.microsoft.com/en-us/azure/virtual-machines/co-location for details.</p> <p>Enables deployment of Mediant CE into an availability set located in the specific availability zone.</p> <p>Example:</p> <pre>proximity_placement_group_vm_zone = 1</pre> <p>Due to limitations of Azure API, if you specify this parameter you must also specify the <code>proximity_placement_group_vm_sizes</code> parameter.</p>	Can't be modified after stack creation
private_ip_sbc-X_eth*	All	<p>Defines pre-defined private IP addresses. See Section 3.25.3 for more information.</p> <p>See also the <code>ipv6_ip_*</code> and <code>public_ip_*</code> parameters.</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
public_ips	All	<p>Defines the Signaling Component's network interface names for which public IP addresses are allocated and optionally, the number of corresponding IP addresses.</p> <p>During stack creation (via Web interface), Stack Manager lets you specify which subnets (and corresponding network interfaces) are assigned with public (Elastic) IP addresses using the Public IPs parameter in the Networking section.</p> <p>When the <code>public_ips</code> advanced configuration parameters is specified, it overrides any value configured by the Public IPs parameter. Typically, you will use this parameter when you need to create multiple IP addresses on the same network interface.</p> <p>Syntax: comma-separated list of subnet names: "main", "additional1", "additional2", etc.. Example:</p> <pre>public_ips = additional1,additional2</pre> <p>You can also use "all" to specify all subnets.</p> <p>If more than one public IP address is required on the specific network interface, this can be specified as "<name>:<num>", where <num> is the total number of public IP addresses to be created, e.g.,:</p> <pre>public_ips = additional1:2</pre> <p>Alternatively, you can specify the interface names "ethX" instead of subnet names, e.g.,:</p> <pre>public_ips = eth1,eth2</pre> <p>Notes:</p> <ul style="list-style-type: none"> ■ For HA deployment on Azure, the <code>public_ips</code> parameter specifies "logical" IP addresses. For each such "logical" address, two IP addresses are allocated on the VM - one is placed behind Public Load Balancer and is used for signaling (SIP) traffic and another one has public IP address attached directly to the VM and is used for media (RTP) traffic. ■ For HA deployment on Google Cloud, public IP addresses are supported only on the "main" interface. They are allocated on Network Load Balancer and the "main" interface is placed behind it. ■ Stack Manager implicitly creates all private IP addresses required for public IP address assignment. ■ After stack creation, use the Public IPs parameter in the Modify dialog to change the parameter value. 	Update
public_ip_eth* , public_ip_sbc-X_eth*	All	Defines pre-defined public IP addresses. See Section 3.25.3 for more information.	Update

Parameter	Applicable Environment	Description	Apply Mode
		See also the <code>ipv6_ip_*</code> and <code>private_ip_*</code> parameters.	
<code>public_ip_prefix</code>	Azure	<p>Defines a Public IP Prefix from which public IP addresses are allocated.</p> <p>Syntax: comma-separated list of elements: <code><rg>/<name>/<count></code></p> <p>Where:</p> <ul style="list-style-type: none"> ■ <code><rg></code> is the resource group name. ■ <code><name></code> is the name of the Public IP Prefix. ■ <code><count></code> is the number of IP addresses allocated from the specific prefix. <p>Examples:</p> <pre>public_ip_prefix = rg1/prefix1/16 public_ip_prefix = rg1/prefix1/4, rg2/prefix2/8</pre>	Update
<code>resource_group</code>	Azure	<p>Defines the name of the existing Resource Group. If not empty, stack resources are deployed in this Resource Group instead of creating a new one. The Resource Group must be empty prior to stack creation.</p> <p>Example:</p> <pre>resource_group = SbcGroup1</pre>	Can't be modified after stack creation
<code>sbc_image_id</code>	AWS, Azure	<p>Defines the local image (instead of the Marketplace image).</p> <p>In contrast to the <code>image_id</code> parameter, this parameter is applied via the Rebuild (and not Update) operation.</p> <p>For Azure, specify the resource group name followed by the image name, e.g.,:</p> <pre>image_id = rg1/image1</pre> <p>For AWS, specify the AMI ID, e.g.,:</p> <pre>image_id = ami-9a50cff5</pre>	Rebuild
<code>sbc_tags</code>	Azure	<p>Defines tags that are assigned to VM instances. These tags are added “on top” of tags specified via the <code>tags</code> parameter, which are assigned to all created resources.</p> <p>Syntax: comma-separated list of name=value pairs, e.g.,:</p> <pre>sbc_tags = duration=overnight</pre>	Update
<code>signaling_nsg_id</code>	AWS	<p>Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager.</p> <p>Refer to the <i>Security Groups</i> chapter in the <i>Mediant VE for AWS Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.</p> <p>Syntax is similar to <code>ha_nsg_id</code> parameter.</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
		See also the <code>media_nsg_id</code> and <code>oam_nsg_id</code> parameters.	
spot_instances	Azure	<p>Enables the use of Azure Spot instances for testing environments. Note that Spot instances might be abruptly stopped and therefore, should never be used in production environment.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: Use Spot instances. ■ disable: (Default) Use regular instances. <p>Example:</p> <pre>spot_instances = enable</pre>	Can't be modified after stack creation
storage_account_type	Azure	<p>Defines the storage account type for managed disks.</p> <p>Valid values include:</p> <ul style="list-style-type: none"> ■ Standard_LRS ■ Premium_LRS ■ StandardSSD_LRS <p>Example:</p> <pre>storage_account_type = Premium_LRS</pre>	Rebuild
tags	All	<p>Defines tags that are assigned to the following stack resources:</p> <ul style="list-style-type: none"> ■ AWS: Tags are assigned to EC2 instance, volume, network interfaces and Elastic IPs. ■ Azure: Tags are assigned to all created resources. ■ Google Cloud: Tags are applied to VM instances. <p>Syntax:</p> <ul style="list-style-type: none"> ■ AWS or Azure: comma-separated list of name=value pairs, e.g.,: <pre>tags = type=sbc,product=ve</pre> ■ Google: comma-separated list of tags <pre>tags = sbc,ve</pre> <p>See also <code>sbc_tags</code> parameter.</p>	<p>Azure: Update</p> <p>Other: Rebuild</p>
update_needed	All	<p>When set to "reset", it turns off the <code>update_needed</code> flag without applying any changes upon the next "update" action.</p> <p>Example:</p> <pre>update_needed = reset</pre>	Update
user_data	All	<p>Defines additional cloud-init configuration parameters.</p> <p>Syntax: a single line with <code>\n</code> as line delimiter.</p> <p>Example:</p> <pre>sc_user_data = #customer-id\n123456\n#license-key\nokRTr5top...</pre>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
use_availability_set	Azure	Applicable for HA deployments. Defines whether availability set is created for 'availability set' deployment topology. Supported values: <ul style="list-style-type: none"> ■ enable: (Default) Create availability set. ■ disable: Don't create availability set. Virtual machines are deployed without any redundancy constraints. Example: <pre>use_availability_set = disable</pre>	Can't be modified after stack creation
use_placement_group	AWS	Applicable for HA deployments. Defines if two Mediant VE instances are deployed in the placement group. Supported values: <ul style="list-style-type: none"> ■ enable: (Default) Use placement group. ■ disable: Don't use placement group. Example: <pre>use_placement_group = disable</pre>	Can't be modified after stack creation
use_proximity_placement_group	Azure	Applicable for HA deployments. Defines if two Mediant VE instances are deployed in the proximity placement group. Supported values: <ul style="list-style-type: none"> ■ enable: (Default) Use proximity placement group. ■ disable: Don't use proximity placement group. Example: <pre>use_proximity_placement_group = disable</pre>	Can't be modified after stack creation
virtual_ip_eth*	AWS	Applicable to multi-zone AWS deployments. Defines virtual IP addresses for communication in the VPC or via the Transit Gateway. Example: <pre>virtual_ip_eth2 = 10.5.1.10</pre>	Update
virtual_ips	AWS	Defines whether "virtual IPs" are used for IPv4 addresses in multi-zone AWS deployments. Supported values: <ul style="list-style-type: none"> ■ enable (default) ■ disable Example: <pre>virtual_ips = disable</pre>	Rebuild
voip_nsg_id	Azure	Defines the name of the existing Network Security Group (NSG) to be used instead of default security groups created by Stack Manager. Refer to the <i>Security Groups</i> chapter in the <i>Mediant VE for Azure Installation Manual</i> for a detailed list of rules that should be included in the specific NSG.	Update

Parameter	Applicable Environment	Description	Apply Mode
		Syntax is similar to the <code>ha_nsg_id</code> parameter. See also the <code>main_nsg_id</code> parameter.	
volume_type	AWS	Defines the volume type for EBS disks. Valid values include: <ul style="list-style-type: none">■ gp2■ gp3■ io1■ io2■ sc1■ sc2■ standard Example: <code>volume_type = gp3</code>	Rebuild

3.8.11.3 Advanced Configuration for VoiceAI Connect

The following table describes advanced parameters available for VoiceAI Connect.

Table 3-7: Advanced Parameters Description

Parameter	Applicable Environment	Description	Apply Mode
auto_start_time	All	<p>Defines the time of day when stack automatically starts.</p> <p>Supported syntax:</p> <ul style="list-style-type: none"> ■ 08:00: Time of day (24h). ■ 1/08:00: Weekday (0=Sunday, 1=Monday, ..., 6=Saturday) and time. ■ 0,1,2/08:00: Multiple weekdays and time. ■ 0-5/08:00: Range of weekdays and time. ■ 0,1/08:00 2-4/09:00: Multiple statements. <p>Example:</p> <pre>auto_start_time = 08:00</pre>	Instant
auto_stop_time	All	<p>Defines the time of day when stack automatically stops.</p> <p>Syntax is identical to the auto_start_time parameter.</p> <p>Example:</p> <pre>auto_stop_time = 22:00</pre>	Instant
automatic_update_url	All	<p>Defines the automatic update URL that is configured on the “worker” SBC instances and front-end SBC.</p> <p>If configured:</p> <ul style="list-style-type: none"> ■ On the front-end SBC, the <code>IncrementalIniFileURL</code> parameter is provisioned with the value: <code>\${url}/global/fe-incremental.ini</code> ■ On the “worker” SBCs, the <code>IncrementalIniFileURL</code> parameter is provisioned with the value: <code>\${url}/global/sbc-incremental.ini</code> <p>Example:</p> <pre>automatic_update_url = https://my.company.com/files</pre> <p>During stack creation (via Web interface), this parameter is configured via the Automatic Update URL parameter in the Create dialog.</p>	Update

Parameter	Applicable Environment	Description	Apply Mode
bot_dialplan	All	<p>Defines the bot Dial Plan on the “worker” SBCs. If enabled, the bots Dial Plan is created and the <code>DialPlanCSVFileUrl</code> configuration parameter is provisioned with the value: <code>\${url}/global/bot-dialplan.csv</code></p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ disable (Default) ■ enable <p>Example:</p> <pre>bot_dialplan = enable</pre> <p>During stack creation (via Web interface), this parameter is configured via the Use Bot Dialplan parameter in the Create dialog.</p>	Update
center_base_url center_token	All	<p>Defines the base URL and API credentials of the pre-created configuration manager (center) instance.</p> <p>If you use these parameters, you are also expected to use the <code>sm_env_vars</code> advanced config parameter to specify the <code>MONGODB_HOST</code> parameter for session managers.</p> <p>Example:</p> <pre>center_base_url = https://fqdn center_token = 123456</pre>	Rebuild (center)
center_disk_size, sm_disk_size	AWS, Azure	<p>Defines the disk size (in GB) for configuration manager / data center and session manager instances.</p> <p>Example:</p> <pre>center_disk_size = 64</pre>	Rebuild
center_env_vars, sm_env_vars	All	<p>Defines additional environment variables for configuration manager / data center and session manager instances.</p> <p>Syntax: a single line with <code>\n</code> as line delimiter, e.g.,:</p> <pre>sm_env_vars = VAR1=val1 center_env_vars = VAR2=val2\nVAR3=val3</pre>	Update

Parameter	Applicable Environment	Description	Apply Mode
center_iam_role, sm_iam_role	AWS	Defines the IAM role assigned to the configuration manager / data center or session manager virtual machine instances. Example: <pre>center_iam_role = CENTER-ROLE</pre>	Rebuild
center_image_id, sbc_image_id, sm_image_id, vaic_image_id	All	Defines the local image for corresponding stack components. Use the <code>vaic_image_id</code> parameter to specify the local image for both configuration manager / data center and session manager components. Syntax: <ul style="list-style-type: none"> ■ AWS: AMI ID, e.g.,: <pre>vaic_image_id = ami-8b41cff2</pre> ■ Azure: Resource Group name / image name, e.g.,: <pre>vaic_image_id = rg1/image1</pre> 	Rebuild
center_image_url, sbc_image_url, sm_image_url, vaic_image_url	All	Defines a URL that contains a text file with the value of the corresponding *_image_id parameter. Syntax: <ul style="list-style-type: none"> ■ <code>(http https)://<path></code> ■ <code>(http https)://<username>:<password>@<path></code> ■ <code>file://<filepath></code> Example: <pre>vaic_image_url = https://sbc123.blob.core.windows.net/pub/vaic_image.txt</pre>	Rebuild
center_instance_type, sbc_instance_type, sm_instance_type	All	Defines the instance type / VM size for the corresponding components ("configuration manager / data center" / SBC / "session manager"). Example: <pre>sm_instance_type = Standard_D4ds_v4</pre>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
center_nsg_id, sbc_nsg_id, sbc_public_nsg_id, sm_nsg_id	AWS	Defines the name of the existing Network Security Group (NSG) for the corresponding component's network interface, instead of creating a new one. Syntax: <ul style="list-style-type: none"> ■ AWS: Security Group ID, e.g.,: <pre>center_nsg_id = sg-123</pre> ■ Azure: Resource Group name / NSG name, e.g.,: <pre>sm_nsg_id = rg1/sm-nsg</pre> 	Rebuild
center_public_ips, sbc_public_ips, sm_public_ips	All	Defines network interface names on the corresponding components ("configuration manager / data center" / SBC / "session manager") that should be assigned with public IP addresses. Syntax: comma-separated list of interface names: <ul style="list-style-type: none"> ■ main: Assign a public IP address to the "main" interface (eth0). ■ public: Assign a public IP address to the "public" interface (eth1) – applicable to SBC components only. Examples: <pre>sm_public_ips = main sbc_public_ips = main,public</pre>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
center_startup_script_url, sm_startup_script_url, vaic_startup_script_url	All	<p>Defines a URL that contains a custom script that is run on the “center” or “session manager” virtual machine instances during their creation. This, for example, can be used to install additional software on “center” or “session manager” virtual machines.</p> <p>Sample script:</p> <pre>#!/bin/bash echo "text" > /opt/test.txt</pre> <p>Use the vaic_startup_script_url parameter if you want to run the same script on both “center” and “session manager” virtual machines.</p> <p>Syntax:</p> <ul style="list-style-type: none"> ■ (http https)://<path> ■ (http https)://<username>:<password>@<path> ■ file://<filepath> <p>Example:</p> <pre>vaic_startup_script_url = https://sbc123.blob.core. windows.net/pub/startup.s h</pre>	Rebuild
center_tags, sm_tags, sbc_tags	All	<p>Defines tags assigned to the following signaling components’ resources:</p> <ul style="list-style-type: none"> ■ AWS: Tags are assigned to EC2 instance, volume, network interfaces and Elastic IPs. ■ Azure and Google Cloud: Tags are assigned to VM instances. <p>Syntax:</p> <ul style="list-style-type: none"> ■ AWS or Azure: comma-separated list of name=value pairs, e.g.,: <pre>sm_tags = type=vaic,role=sm</pre> ■ Google: comma-separated list of tags <pre>sm_tags = vaic,sm</pre> <p>See also the <code>tags</code> and <code>common_tags</code> parameters.</p>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
common_tags	AWS	<p>Defines tags that are assigned to created network security groups.</p> <p>Syntax: comma-separated list of name=value pairs.</p> <p>Example:</p> <pre>common_tags = type=vaic</pre> <p>See also the <code>center_tags</code>, <code>sbc_tags</code> and <code>sm_tags</code> parameters.</p>	Rebuild
deployment_mode	All	<p>Defines the deployment mode for VoiceAI Connect components.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ docker (Default) ■ podman <p>“Podman” mode is supported only on RHEL / CentOS / Rocky Linux hosts.</p>	Rebuild
frontend_proxysset_name	All	<p>Defines the name of the Proxy Set on the front-end SBC, which is updated with the VoiceAI Connect worker SBC IP addresses upon initial configuration, scale-out/scale-in and “reconfigure” operation. If not specified, the 'vaic' Proxy Set is used.</p> <p>Example:</p> <pre>frontend_proxysset_name = my-ps-1</pre>	Reconfigure (fe)
frontend_transport_type	All	<p>Defines the transport type for communication between the front-end SBC and VoiceAI Connect worker SBCs.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ udp (Default) ■ tcp ■ tls <p>Example:</p> <pre>frontend_transport_type = tls</pre>	Reconfigure (sbc)
imds	AWS	<p>Defines the version of the AWS meta-data instance for the deployed EC2 instances.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ any: Allow both IMDSv1 and IMDSv2. ■ v2: (Default) Enforce IMDSv2. <p>Example:</p> <pre>imds = v2</pre>	Update

Parameter	Applicable Environment	Description	Apply Mode
kms_key_id	AWS	<p>Identifier of the AWS KMS key for Amazon EBS disk encryption. You can specify the key via one of the following:</p> <ul style="list-style-type: none"> ■ Key ID ■ Key alias ■ Key ARN ■ Alias ARN <p>Example:</p> <pre>kms-key-id = arn:aws:kms:us-east-1: 012345678910:1234abcd- 12ab6ef-1234567890ab</pre>	Rebuild
manage_via_https	All	<p>Defines the protocol used by Stack Manager when connecting to the deployed stack's management interface.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Use HTTPS protocol. ■ disable: Use HTTP protocol. <p>Example:</p> <pre>manage_via_https = disable</pre>	Instant
oam_ip	All	<p>Defines the IP addresses used by Stack Manager to manage the VoiceAI Connect stack.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ internal: (Default) Use the internal IP addresses. ■ public: Use the public IP addresses. <p>Example:</p> <pre>oam_ip = public</pre> <p>During stack creation (via Web interface), this parameter is configured via the Manage via Public IPs parameter in the Create dialog.</p>	Rebuild
private_ip_center_eth* , private_ip_sbc-X_eth* , private_ip_sm-X_eth*	All	<p>Defines predefined private IP addresses. See Section 3.25.3 for more information. See also the <code>public_ip_*</code> parameters.</p>	Rebuild
public_ip_center_eth* , public_ip_sbc-X_eth* , public_ip_sm-X_eth*	All	<p>Defines predefined public IP addresses. See Section 3.25.3 for more information. See also the <code>private_ip_*</code> parameters.</p>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
resource_group	Azure	<p>Defines the name of the existing Resource Group.</p> <p>If not empty, stack resources are deployed in this Resource Group instead of creating a new one. The Resource Group must be empty prior to stack creation.</p> <p>Example:</p> <pre>resource_group = MyGroup1</pre>	Can't be modified after stack creation
sbc_ini_config	All	<p>Defines additional configuration parameters (in INI file format) for SBC instances during stack creation / rebuild.</p> <p>Syntax: a single line with \n as line delimiter, e.g.,:</p> <pre>ini_incremental = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</pre> <p>Specified configuration is applied via REST API and therefore, has no size limit. Therefore, this parameter is preferred over the <code>sbc_ini_params</code> parameter that is applied via the cloud-init mechanism and is therefore, limited by instance user-data size.</p>	Update
sbc_ini_params	All	<p>Defines additional configuration parameters (in INI file format) for SBC instances during stack creation / rebuild.</p> <p>Syntax: a single line with \n as line delimiter, e.g.,:</p> <p>Example:</p> <pre>sc_ini_params = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</pre> <p>Specified configuration is applied via the cloud-init mechanism and is therefore limited by instance user-data size (for example, in AWS it's limited by 16 KB). For long configurations, use the <code>sbc_ini_config</code> parameter instead.</p>	Rebuild

Parameter	Applicable Environment	Description	Apply Mode
sbc_user_data	All	<p>Defines additional cloud-init configuration parameters for SBC instances.</p> <p>Syntax: a single line with \n as line delimiter.</p> <p>Example:</p> <pre>sc_user_data = #customer-id\n123456\n#license-key\nokRTr5top...</pre>	Rebuild
ssh_key_user, ssh_key_file	AWS	<p>By default, Stack Manager uses credentials provisioned during initial stack creation to establish connection with “configuration manager / data center” and “session manager” instances, for example during <i>Upgrade</i> operation.</p> <p>If you want Stack Manager to authenticate with an SSH public key instead, configure the following parameters:</p> <ul style="list-style-type: none"> ■ ssh_key_user: Username. ■ ssh_key_file: Full path to the private key file in PEM format stored on Stack Manager VM and readable by the <code>stack_mgr</code> user. <p>Example:</p> <pre>ssh_key_user = admin ssh_key_file = /var/stack_mgr/ssh_key_1.pem</pre>	Rebuild
spot_instances	Azure	<p>Enables Azure Spot instances for testing environments. Note that Spot instances might be abruptly stopped and therefore, should never be used in production environment.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: Use Spot instances. ■ disable: (Default) Use regular instances. <p>Example:</p> <pre>spot_instances = enable</pre>	Can't be modified after stack creation

Parameter	Applicable Environment	Description	Apply Mode
storage_account_type	Azure	<p>Defines the storage account type for managed disks.</p> <p>Valid values include:</p> <ul style="list-style-type: none"> ■ Standard_LRS ■ Premium_LRS ■ StandardSSD_LRS <p>Example:</p> <pre>storage_account_type = Premium_LRS</pre>	Rebuild
tags	Azure	<p>Defines tags that are assigned to all created stack resources.</p> <p>Syntax: comma-separated list of name=value pairs</p> <ul style="list-style-type: none"> ■ Example: <pre>tags = type=vaic, region=us</pre> <p>You can also use the <code>center_tags</code>, <code>sbc_tags</code>, and <code>sm_tags</code> parameters to define <i>additional</i> tags that are assigned to corresponding VMs only.</p>	<p>Azure: Update</p> <p>Other: Rebuild</p>
volume_type	AWS	<p>Defines the volume type for EBS disks.</p> <p>Valid values include:</p> <ul style="list-style-type: none"> ■ gp2 ■ gp3 ■ io1 ■ io2 ■ sc1 ■ sc2 ■ standard <p>Example:</p> <pre>volume_type = gp3</pre>	
use_placement_group	AWS	<p>Defines if components are deployed in the placement group.</p> <p>Supported values:</p> <ul style="list-style-type: none"> ■ enable: (Default) Use placement group. ■ disable: Don't use placement group. <p>Example:</p> <pre>use_placement_group = disable</pre>	Can't be modified after stack creation

Parameter	Applicable Environment	Description	Apply Mode
use_proximity_placement_group	Azure	<p>Defines if components are deployed in the proximity placement group.</p> <p>Supported values:</p> <ul style="list-style-type: none">■ enable: (Default) Use proximity placement group.■ disable: Don't use proximity placement group. <p>Example:</p> <pre>use_proximity_placement_group = disable</pre>	Can't be modified after stack creation

3.9 Checking Stack State and Configuration

To check the state and configuration of the existing stack, open the Stacks page and then click the specific stack. The Stack Information page is displayed, which allows you to check the current stack state, inspect and modify its configuration, and perform actions, for example, scale-out or scale-in.

Figure 3-25: Stack Information Page

The screenshot displays the Stack Information page for a stack named 'ce1'. The interface includes a navigation bar with 'stack_mgr' and links for 'Stacks', 'Configuration', 'Logs', and 'About', along with a 'Logout' button. Below the navigation bar is a toolbar with action buttons: Start, Stop, Heal, Scale Out, Scale In, Scale To, Modify, Update, More, and Delete. The main content area is titled 'ce1' and is divided into four panels:

- General:**
 - Name: ce1
 - Type: Mediant CE
 - Environment: AWS
 - State: running
 - IP Address: 3.124.221.82
 - Manage via HTTPS: enable
 - Created On: Nov 21, 2019 11:41:12
 - Started On: Nov 21, 2019 11:41:12
 - Minimum number of media components: 2
 - Maximum number of media components: 5
 - Update needed: false
 - Modified parameters: (empty)
- Automatic Scaling:**
 - Automatic scaling: disable
 - Media utilization scale in threshold: > 250% free
 - Media utilization scale out threshold: < 100% free
 - DSP utilization scale in threshold: disabled
 - DSP utilization scale out threshold: disabled
 - Automatic scaling cool down time: 900 sec
 - Automatic scaling scale-in step: 1
 - Automatic scaling scale-out step: 1
- Media Components:**
 - Instance type: r4.large
 - Image ID: (empty)
 - Profile: forwarding
 - Max rate limit: auto
 - Table of Media Components:

ID	IP Address	Status	% Media	% DSP	Type
mc-1	172.31.228.240	connected	0	-	r4.large
mc-2	172.31.233.182	connected	0	-	r4.large
 - Number of media components: 2
 - Connected media components: 2
 - Free media resources: 200%
- Signaling Components:**
 - Instance Type: r4.2xlarge
 - Image ID: (empty)
 - Disk Size: 10 GB
 - Table of Signaling Components:

ID	IP Address	Status	Type
sc-1	172.31.237.113	active	r4.2xlarge
sc-2		standby	r4.2xlarge

Stack Manager communicates with both cloud API and the deployed stack's management API when checking the stack state. If both APIs are working correctly, individual component statuses are populated with "active" / "standby" / "connected" / "disconnected" / "up" / "down" values.

If stack manager fails to communicate with the stack's management API, it displays component states "running" / "stopped" / "deallocated" reported by the cloud API and raises "Cannot connect to SBC via REST API" alarm". Refer to 3.9.3 Checking Connectivity on how to troubleshoot and fix such a problem.

3.9.1 Viewing IP Addresses of Stack Components

You can view IP addresses of all stack components. On the toolbar, click **More > Show IP Addresses**. The output also includes the **Advanced Config** section that can be used to re-create the stack while preserving its current network addresses.

Figure 3-26: Show IP Addresses Page

IP Addresses

Component	Interface	Private IP Address	Public IP Address
public-lb	eth1		20.115.200.133
	eth0	10.9.0.5	
sbc-1	eth1	10.9.1.4	
	eth1:1	10.9.1.5	20.99.165.170
	eth0	10.9.0.4	
sbc-2	eth1	10.9.1.6	
	eth1:1	10.9.1.7	20.115.200.151

Advanced config

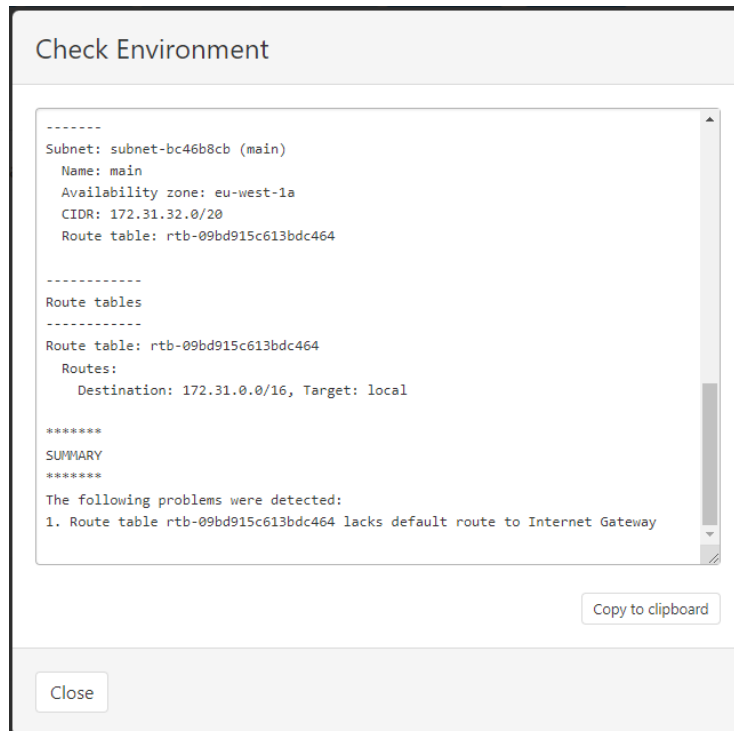
```

public_ip_sbc_eth1 = Eitan-HA-VE/Eitan-HA-VE-sbc-eth1-ip
public_ip_sbc-1_eth1 = Eitan-HA-VE/Eitan-HA-VE-sbc-1-eth1-ip
private_ip_sbc-1_eth0 = 10.9.0.5
private_ip_sbc-1_eth1 = 10.9.1.4,10.9.1.5
public_ip_sbc-2_eth1 = Eitan-HA-VE/Eitan-HA-VE-sbc-2-eth1-ip
private_ip_sbc-2_eth0 = 10.9.0.4
          
```

3.9.2 Checking Deployment Environment

For Mediant VE and CE stacks deployed in an AWS environment, you can check the deployment environment, by clicking **More > Check Environment**. This provides a detailed summary of the deployment environment and tries to detect common (mis) configuration issues (e.g., lack of EC2 Endpoint or invalid configuration of associated security group).

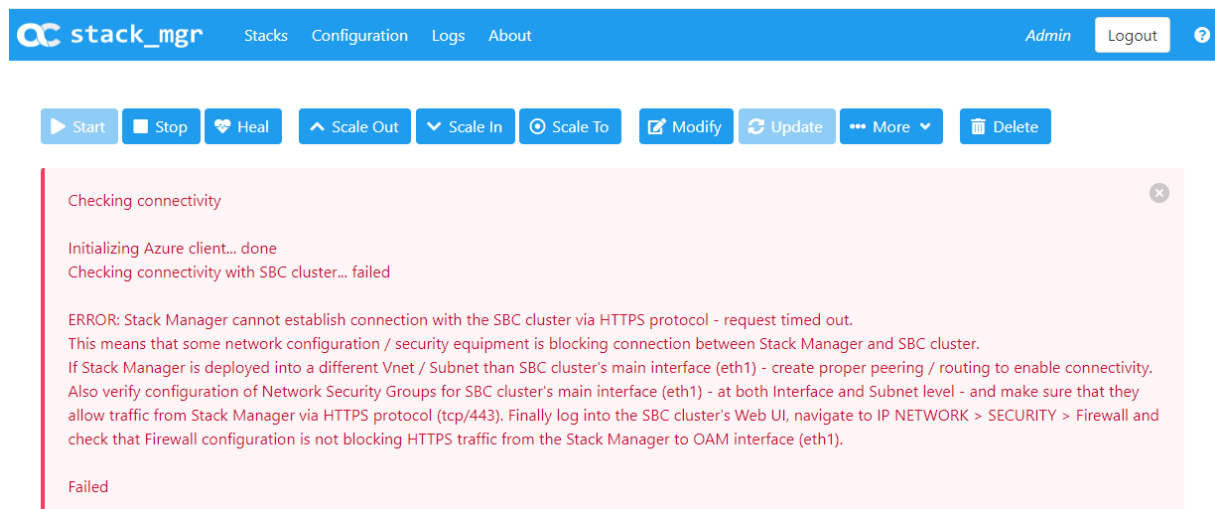
Figure 3-27: Check Environment Page



3.9.3 Checking Connectivity

You can check connectivity between Stack Manager and the deployed stack, by clicking **More > Check Connectivity**. Stack Manager runs basic connectivity tests and suggests common problem resolutions if connectivity tests fail.

Figure 3-28: Check Connectivity Output



3.9.4 Updating Connectivity

If connectivity with the stack fails because of an incorrect IP address or credentials, click **More > Update Connectivity** to update these parameters and restore connectivity.

Figure 3-29: Update Connectivity Page

Update Connectivity

The following parameters are used by Stack Manager to connect to the SBC cluster via REST API. Update them if connectivity between Stack Manager and SBC cluster 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

Username

Password



For Mediant VE and CE stacks, Stack Manager creates a dedicated *StackMgr* user with a randomized password during stack deployment and uses it to communicate with the deployed stack (via REST API). It's recommended to keep using this dedicated user and only update its password, if needed.

3.10 Active Alarms

Stack Manager periodically checks the state of all created stacks and raises alarms if it discovers any problem. Active alarms are displayed in Stacks summary screen and in the detailed Stack Information page.

Figure 3-30: Active Alarms in Stacks Summary Screen

stack_mgr

[Stacks](#)
[Configuration](#)
[Logs](#)
[About](#)
Logout

+ Create new stack

Stacks

Name	Type	Environment	State	Alarms	IP Address
alex-ce-test-1	Mediant CE	AWS	running	mc-2-down	18.158.47.8
alex-ce-test-2	Mediant CE	Azure	stopped		40.64.82.179

Figure 3-31: Active Alarms in Stack Information Page

The screenshot shows the Stack Manager web interface. At the top, there is a navigation bar with the 'stack_mgr' logo and links for 'Stacks', 'Configuration', 'Logs', and 'About'. A 'Logout' button is in the top right corner. Below the navigation bar is a toolbar with buttons for 'Start', 'Stop', 'Heal', 'Scale Out', 'Scale In', 'Scale To', 'Modify', 'Update', 'More', and 'Delete'. The main content area is titled 'alex-ce-test-1' and is divided into three panels:

- General:** A table with the following information:

Name	alex-ce-test-1
Type	Mediant CE
Environment	AWS
State	running
IP Address	18.158.47.8
OS Version	6
- Active Alarms:** A table with the following information:

Description	Raised on
Media component 'mc-2' is 'disconnected'	Jul 02, 2020 09:47:56
- Automatic Scaling:** A table with the following information:

Automatic scaling	disable
Media utilization scale in threshold	> 250% free

The following alarms are supported:

- **rest-api:** The alarm is raised when Stack Manager can't read the status of Mediant VE/CE via REST API
- **mc-status:** The alarm is raised when Stack Manager can't read the status of Mediant CE's Media Components via REST API.
- **mc-X-down:** The alarm is raised when Media Component mc-X is not in service (alarm description provides detailed Media Component state).
- **mc-X-missing:** The alarm is raised when Media Component mc-X is missing from Mediant CE's configuration and Stack Manager can't fix it.
- **sc-X-down:** The alarm is raised when Signaling Component sc-X is down.
- **sc-ha-alarm:** The alarm is raised when Signaling Components are not in HA synchronized state

To avoid false alarms, most of the alarms are raised only after the problem persists for 5 minutes.

3.11 Performing Operations on Stack

You can perform operations on the running stack (e.g., Scale Out), by clicking the corresponding button on the toolbar of the Stack Information page.

All operations, except for Delete and Heal, are serialized and can be performed one at a time. For example, if you started the *Scale Out* operation, you have to wait until it completes prior to starting the *Scale In* operation.

The stack state is updated accordingly when an operation is being performed.

3.12 Scaling Mediant CE Stack



This section is applicable only to Mediant CE stacks.

The number of active Media Components in the Mediant CE stack can vary to match the required service capacity. This is called scaling and ensures that the stack utilizes the optimal amount of resources at any point of time and elastically scales on demand. An operation that increases the amount of active Media Components is called *Scale Out*; an operation that decreases the amount of active Media Components is called *Scale In*.

To ensure fast and reliable scaling, Stack Manager pre-creates all needed Media Components in advance (up to the maximum number) and stops/starts them accordingly during scale in/out operations.

Scaling decision can be triggered either manually—by running the *Scale In*, *Scale Out* or *Scale To* commands—or automatically based on the current cluster utilization.

The size of the cluster is configured by the following two configuration parameters:

- Minimum Number of Media Components
- Maximum Number of Media Components

3.12.1 Scale Out Operation

The *Scale Out* operation increases the number of Media Components in the Mediant CE stack, by starting additional pre-created "idle" Media Components (for example, corresponding to the AWS EC2 instance state changes from *stopped* to *running*).

You must specify the number of Media Components to add to the service. Alternatively, you can specify names of Media Components that will be added to the service (e.g., "mc-3,mc-4").

Figure 3-32: Scale Out Operation

Scale Out

How many Media Components do you want to activate?

1

Specify names of components to be activated (optional)

e.g. mc-3,mc-4

Scale Out Cancel

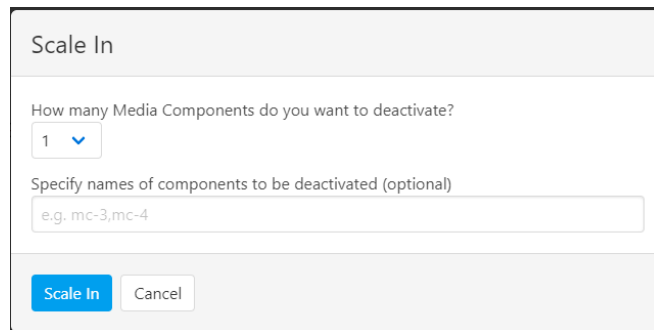
The *Scale Out* operation is not allowed when *Automatic Scaling* is enabled. Use the *Scale To* operation instead.

3.12.2 Scale In Operation

The *Scale In* operation decreases the number of Media Components in the Mediant CE stack, by stopping a certain number of "active" Media Components (for example, corresponding to the AWS EC2 instance state changes from *running* to *stopped*).

You must specify the number of Media Components to be removed from the service. Alternatively, you can specify names of Media Components that will be removed from the service (e.g., "mc-3,mc-4").

Figure 3-33: Scale In Operation



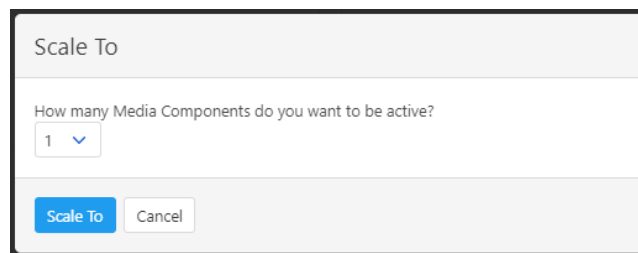
The screenshot shows a dialog box titled "Scale In". It contains a label "How many Media Components do you want to deactivate?" followed by a dropdown menu with the value "1" selected. Below this is a text input field with the placeholder text "Specify names of components to be deactivated (optional)" and the example text "e.g. mc-3,mc-4". At the bottom of the dialog are two buttons: "Scale In" (highlighted in blue) and "Cancel".

The *Scale In* operation is not allowed when *Automatic Scaling* is enabled. Use the *Scale To* operation instead.

3.12.3 Scale To Operation

The *Scale To* operation sets the number of Media Components in the Mediant CE stack to the specified value. It essentially performs a *Scale In* or *Scale Out* operation, depending on the current stack state.

Figure 3-34: Scale To Operation



The screenshot shows a dialog box titled "Scale To". It contains a label "How many Media Components do you want to be active?" followed by a dropdown menu with the value "1" selected. At the bottom of the dialog are two buttons: "Scale To" (highlighted in blue) and "Cancel".

In contrast to *Scale In* and *Scale Out* operations, the *Scale To* operation is allowed when *Automatic Scaling* is enabled. Regardless of whether it adds or removes Media Components, for the purposes of calculating a cool down period, the *Scale To* operation is considered to be equivalent to the *Scale Out* operation. This means that the cluster size can be increased immediately after completing the *Scale To* command, if needed.

3.13 Automatic Scaling



This section is applicable only to Mediant CE stacks.

Automatic Scaling adjusts the Mediant CE cluster size to the current service needs, by measuring current cluster utilization and changing its size accordingly. It's implemented by a background job performed by the Stack Manager.

For every stack that is in "running" state and has Automatic Scaling enabled, Stack Manager calculates the total amount of "free" media and DSP resources, using accumulative percentage points, where 100% corresponds to the capacity of a single Media Component. For example, for a cluster that is in the following state:

id	IP address	status	%media	%dsp
mc-1	172.31.78.116	connected	30	0
mc-2	172.31.75.42	connected	40	0
mc-3	172.31.65.5	connected	25	0

Free media resources are calculated as follows:

$$free_media = (100-30) + (100-20) + (100-25) = 205 \%$$



The calculated number is the number of excessive Media Components capacity in the Mediant CE cluster. For example, 100% corresponds to the state where the total amount of excessive capacity equals the capacity of a single Media Component. In this state, the failure of a single Media Component has no effect on traffic capacity, thus providing N+1 redundancy for the media cluster.

The calculated number is then compared against *Scale In* and *Scale Out Thresholds*, which are defined in the stack configuration. If the number is below the *Scale Out Threshold*, the *Scale Out* operation is triggered. If the number is above the *Scale In Threshold*, the *Scale In* operation is triggered.

It's possible to disable media or DSP thresholds, by setting them to 0 (zero).

If both media and DSP thresholds are used, the decision is made as follows:

- *Scale Out* is performed when *either* media or DSP utilization is below the threshold
- *Scale In* is performed when *both* media and DSP utilization are above the threshold

Maximum / Minimum Number of Media Components parameters define the maximum / minimum cluster size, and automatic scaling mechanism takes them into account when making its decisions.

Automatic scaling logs are collected in the *auto-job* log, which can be viewed through Web or CLI management interfaces:

```
$ stack_mgr log --name auto_job --lines 10
300% of media resources in stack 'stack1' are unused
MEDIA_UTIL_SCALE_IN_THRESHOLD is 250
Trigger automatic scale in

Choosing SBC media components to be removed..... done
Preparing SBC media component 'mc-3' for removal.... done

Initializing AWS client... done
Updating SBC cluster configuration.... done
Removing SBC media components..... done
```

3.13.1 Cool Down Period

To prevent stack size 'bouncing', the *Automatic Scaling Cool Down Time* parameter defines the minimum time (in seconds) between consecutive *Scale Out* and *Scale In* decisions.

3.13.2 Auto Scale Step

The number of Media Components to be added or removed by the automatic scaling mechanism can be configured using the *Automatic Scaling Scale-In / Scale-Out Step* parameters.

Both parameters are set to 1 by default, thus enabling Automatic Scaling to add or remove one Media Component at a time. If you change the *Automatic Scaling Scale-Out Step* parameter to a greater value (e.g., 2), your stack size will grow quickly to adjust to traffic demands, but will shrink slowly when traffic is reduced.

3.13.3 Changing Cluster Size at Specific Time of Day

In certain scenarios, service capacity is typically expected to change at certain times of day. For example, if the Contact Center starts to operate at 9:00 AM, it would be reasonable to expect that SBC traffic will surge at that time.

It's possible to change Mediant CE scaling while having *Automatic Scaling* enabled, using one of the following methods:

- Changing the *Minimum Number of Media Components* parameter, which defines the minimum cluster size
- Defining the target cluster size by the *Scale To operation*

If you choose to define the target cluster size by the *Scale To operation*, keep in mind that the cool-down period is calculated as if the *Scale Out* operation was performed. Therefore, cluster size will grow immediately if required and will not be reduced for the cool-down period even if traffic hasn't started yet.

The corresponding operations can be programmed to run at a specified time of day using CLI and the cron scheduler. Make sure that commands are run by the *stack_mgr* user, and replace the **stack_mgr** command with the expression `"/var/stack_mgr/bin/stack_mgr"`. For example:

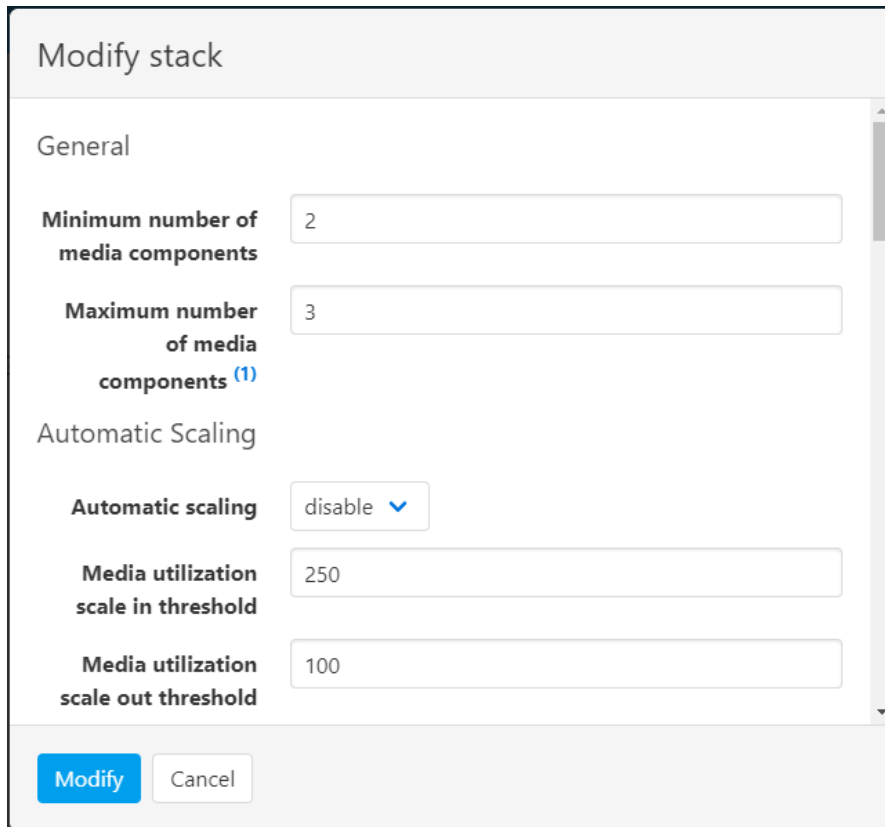
```
$ cat /var/stack_mgr/scale_to.sh
#!/bin/bash
STACK_MGR="/var/stack_mgr/bin/stack_mgr"
$STACK_MGR scale $1 -n $2 >> /var/log/stack_mgr/cron.log
```

```
$ cat /etc/cron.d/stack_mgr
* 9 * * * stack_mgr /var/stack_mgr/scale_to.sh stack1 10
```

3.14 Modifying Stack Configuration

To modify configuration of the existing Mediant VE/CE stack, open the Stack information page, and then click the **Modify** button on the toolbar to open the Modify stack dialog box. Change stack configuration parameters as desired, and then click **Modify** to apply your changes.

Figure 3-35: Modifying Stack Configuration



The screenshot shows a dialog box titled "Modify stack". It is divided into two main sections: "General" and "Automatic Scaling".

- General:**
 - Minimum number of media components:** A text input field containing the value "2".
 - Maximum number of media components (1):** A text input field containing the value "3".
- Automatic Scaling:**
 - Automatic scaling:** A dropdown menu currently set to "disable".
 - Media utilization scale in threshold:** A text input field containing the value "250".
 - Media utilization scale out threshold:** A text input field containing the value "100".

At the bottom of the dialog box, there are two buttons: "Modify" (highlighted in blue) and "Cancel".

Most of the parameters are applied immediately and have no adverse effect on service. However, change of some parameters might require an additional *Update* operation and be service affecting. Such parameters are explicitly marked in the **Modify** screen and the detailed description is provided at the screen footnote.

Figure 3-36: Modify Screen Footnote

Modify stack

OS Version 6 ▾

Advanced Config (3)

Comments

(1) change of parameter requires "Update" command
(2) change of parameter requires "Update" command and causes service interruption
(3) change of Advanced Config requires "Rebuild" command and causes service interruption

Modify Cancel

Figure 3-37: Modifying Parameter that Requires Update

stack_mgr Stacks Configuration Logs About Logout

Start Stop Heal Scale Out Scale In Scale To Modify Update More Delete

Modifying stack
Modifying stack configuration... done

Stack configuration was modified.
Use 'update' command to apply the changes.
Done

3.14.1 Update Operation

The *Update* operation updates the stack to the new configuration. It's required when modified configuration requires applying some changes to the underlying virtual infrastructure resources, for example, when you resize the cluster.

The need to do an *Update* operation is indicated in the *Modify* operation output and on the Stack information page:

Figure 3-38: Stack in "Update Needed" State

The screenshot shows the Stack Manager interface for a stack named 'ce1'. The interface includes a navigation bar with 'stack_mgr', 'Stacks', 'Configuration', 'Logs', 'About', and a 'Logout' button. Below the navigation bar is a toolbar with buttons for 'Start', 'Stop', 'Heal', 'Scale Out', 'Scale In', 'Scale To', 'Modify', 'Update', 'More', and 'Delete'. The main content area is divided into three panels: 'General', 'Automatic Scaling', and 'Media Components'. The 'General' panel shows details for the stack 'ce1', including its name, type (Mediant CE), environment (AWS), state (running), IP address (3.124.221.82), and management settings. The 'Automatic Scaling' panel shows various scaling thresholds and settings. The 'Media Components' panel shows instance type (r4.large), image ID, and profile (forwarding). The 'Update needed' field in the 'General' panel is circled in red, indicating that an update is required. The 'Modified parameters' field shows 'max_mc_num'.

Click the **Update** button on the toolbar to start the *Update* operation.

Figure 3-39: Update Screen

The screenshot shows the 'Update' dialog box. The title is 'Update'. The text inside asks 'How do you want to perform the update?' and provides a hint: '(Hitless update minimizes service interruption, but may be significantly slower)'. There is a dropdown menu with 'Regular update' selected. At the bottom, there are two buttons: 'Update' and 'Cancel'.

You are prompted to choose the update mode:

- **Regular update:** Minimizes update time, but might cause service interruption.
- **Hitless update:** Minimizes service interruption, by performing updates “one component at a time”. It might take significantly more time than the “regular update” mode.

Note that some configuration updates and deployment topologies are not compatible with the “hitless update” mode. For example, if you resize the Mediant VE stack deployed in standalone (non-HA) deployment topology and try to apply the change in “hitless update” mode, an error is displayed, explaining that your configuration change requires service interruption and therefore, should be applied via the “regular update” mode.

Figure 3-40: Updating Stack Configuration

The screenshot shows the Stack Manager web interface. At the top, there is a navigation bar with 'stack_mgr' logo and links for 'Stacks', 'Configuration', 'Logs', and 'About'. A 'Logout' button is in the top right. Below the navigation bar is a toolbar with buttons for 'Start', 'Stop', 'Heal', 'Scale Out', 'Scale In', 'Scale To', 'Modify', 'More', and 'Delete'. The main content area shows a progress bar for 'Updating stack' with the following steps: 'Initializing AWS client... done', 'Checking that configuration is allowed... done', 'Updating signaling components... done', and 'Updating media components.....'. Below the progress bar, the configuration details for a stack named 'ce1' are displayed in two panels: 'General' and 'Automatic Scaling'.

General		Automatic Scaling	
Name	ce1	Automatic scaling	disable
Type	Mediant CE	Media utilization scale in threshold	> 250% free
Environment	AWS	Media utilization scale out threshold	< 100% free
State	updating	DSP utilization scale in threshold	disabled
IP Address	3.124.221.82	DSP utilization scale out threshold	disabled

3.14.2 Modifiable Parameters for Mediant CE

The following table lists all stack configuration parameters that can be modified.

Table 3-8: Modifiable Stack Configuration Parameters

Group Name	Parameter	Applicable Environment	Requires Update	Service Affecting
General	Minimum number of media components	All	No	No
	Maximum number of media components	All	Yes	No
Automatic Scaling	Automatic scaling	All	No	No
	Media utilization scale in threshold	All	No	No
	Media utilization scale out threshold	All	No	No
	DSP utilization scale in threshold	All	No	No
	DSP utilization scale out threshold	All	No	No
	Automatic scaling cool down time	All	No	No
	Automatic scaling scale-in step	All	No	No
Automatic scaling scale-out step	All	No	No	
Automatic Healing	Automatic healing	AWS, Azure, Google	No	No

Group Name	Parameter	Applicable Environment	Requires Update	Service Affecting
	Automatic healing interval	AWS, Azure, Google	No	No
Signaling Components	Number of network interfaces	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Interfaces with public IP	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Interfaces with additional IP	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Management ports	AWS, Azure, Google	Yes	No
	Signaling ports	AWS, Azure, Google	Yes	No
	Media ports	AWS, Azure, Google	Yes	No
	Use main subnet for	AWS, Azure, Google	Yes	No
	Instance type	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Image ID	AWS, Azure, Google	Yes	Yes ⁽¹⁾
Media Components	Number of network interfaces	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Interfaces with public IP	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Interfaces with additional IP	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Media ports	AWS, Azure, Google	Yes	No
	Profile	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Instance type	AWS, Azure, Google	Yes	Yes ⁽¹⁾
	Image ID	AWS, Azure, Google	Yes	Yes ⁽¹⁾
Network Subnets	Additional 1 subnet	AWS, Azure, Google	No ⁽²⁾	No
	Additional 2 subnet	AWS, Azure, Google	No ⁽²⁾	No
Advanced	OS version	AWS, Azure	Yes	Yes ⁽³⁾
	Advanced config	All	Yes ⁽⁴⁾	Yes ⁽¹⁾
	Comments	All	No	No

⁽¹⁾ In AWS and Azure environments, service interruption can be minimized by choosing the “Hitless update” mode in the **Update** operation.

⁽²⁾ Modification of additional subnets is allowed only when they are not in use.

⁽³⁾ Modification of the ‘OS version’ parameter requires an Update operation, during which all VMs are rebuilt. During this operation, the serial number of Signaling Components changes and therefore, their local license will be invalidated. You need to obtain, activate and apply the new license to the Signaling Components to restore service.

⁽⁴⁾ See Section 3.8.11.1 Advanced Configuration for Mediant CE for more information.

3.14.3 Modifiable Parameters for Mediant VE

The following table lists all stack configuration parameters that can be modified.

Table 3-9: Modifiable Stack Configuration Parameters

Group Name	Parameter	Applicable Environment	Requires Update	Service Affecting
Compute	Instance type	All	Yes	Yes ⁽¹⁾
Networking	Number of network interfaces	All	Yes	Yes ⁽¹⁾
	Interfaces with public IP	All	Yes	Yes ⁽¹⁾
	Interfaces with additional IP	All	Yes	Yes ⁽¹⁾
	Management ports	All	Yes	No
	Signaling ports	All	Yes	No
	Media ports	All	Yes	No
	Use main subnet for	All	Yes	No
	Additional 1 subnet	All	No ⁽²⁾	No
	Additional 2 subnet	All	No ⁽²⁾	No
Automatic Healing	Automatic healing	All	No	No
	Automatic healing interval	All	No	No
Advanced	OS version	AWS, Azure	Yes	Yes ⁽³⁾
	Advanced config	All	Yes ⁽⁴⁾	Yes ⁽¹⁾
	Comments	All	No	No

⁽¹⁾ In AWS and Azure environments, service interruption can be minimized by choosing the “Hitless update” mode in the **Update** operation.

⁽²⁾ Modification of additional subnets is allowed only when they are not in use.

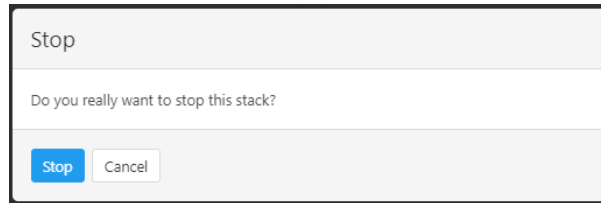
⁽³⁾ Modification of the ‘OS version’ parameter requires an Update operation, during which all VMs are rebuilt. During this operation, the serial number of Signaling Components changes and therefore, their local license will be invalidated. You need to obtain, activate and apply the new license to the Signaling Components to restore service.

⁽⁴⁾ Refer to section 3.8.11.2 Advanced Configuration for Mediant VE for more information.

3.15 Stopping and Starting Stack

If you want to temporarily stop all Mediant CE components (e.g., in a lab environment) use the *Stop* operation. Use the *Start* operation afterwards to return all components back to service.

Figure 3-41: Stopping Stack

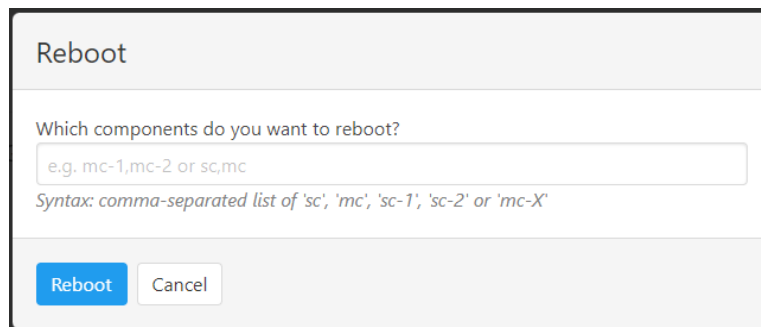


The screenshot shows a dialog box titled "Stop". Below the title bar, the text "Do you really want to stop this stack?" is displayed. At the bottom of the dialog, there are two buttons: a blue "Stop" button and a white "Cancel" button.

3.16 Rebooting Stack Components

Use the *Reboot* operation to reboot specific stack components.

Figure 3-42: Rebooting Stack



The screenshot shows a dialog box titled "Reboot". Below the title bar, the text "Which components do you want to reboot?" is displayed. Below this text is a text input field containing the example "e.g. mc-1,mc-2 or sc,mc". Below the input field, the text "Syntax: comma-separated list of 'sc', 'mc', 'sc-1', 'sc-2' or 'mc-X'" is displayed. At the bottom of the dialog, there are two buttons: a blue "Reboot" button and a white "Cancel" button.

3.17 Healing Stack

The *Heal* operation verifies the state of all stack components and fixes any errors if detected. For example, it can remove Media Components that are not properly registered in the Signaling Components or remove orphaned entries from the "Media Components" configuration table.

The command is typically used after Stack Manager is interrupted in the middle of some operation, for example, during stack creation or *Scale Out*. It can also be useful when the output of some operation (e.g., *Scale In*) indicates an intermittent failure.

In most cases, Stack Manager heals itself automatically (see the following section). However, in some cases, manual healing is needed to ensure that the stack state matches its configuration.

Figure 3-43: Healing Stack

The screenshot shows the Stack Manager web interface. At the top, there is a navigation bar with 'stack_mgr' and links for 'Stacks', 'Configuration', 'Logs', and 'About'. A 'Logout' button is in the top right. Below the navigation bar is a toolbar with buttons for 'Start', 'Stop', 'Heal', 'Scale Out', 'Scale In', 'Scale To', 'Modify', 'Update', 'More', and 'Delete'. The main content area displays the output of the 'Heal' operation for stack 'ce1'.

```

Healing stack
Stack is in 'running' state

Initializing AWS client... done
Checking all components that should be 'up'... done
Checking all media components that should be 'down'... done
Stopping components 'mc-5, mc-4, mc-3'..... done
Checking that all media components have matching SBC configuration... done
Verifying that all media components are unlocked... done

Finished healing

Done
  
```

Below the log output, the stack 'ce1' is selected, and its configuration is shown in two panels: 'General' and 'Automatic Scaling'.

General		Automatic Scaling	
Name	ce1	Automatic scaling	disable
Type	Mediant CE	Media utilization scale in threshold	> 250% free
Environment	AWS	Media utilization scale out threshold	< 100% free
State	running	DSP utilization scale in threshold	disabled
IP Address	3.124.221.82	DSP utilization scale out threshold	disabled
Manage via HTTPS	enable	Automatic scaling cool down time	900 sec

3.17.1 Automatic Healing

Stack Manager automatically triggers a *Heal* operation when it detects that an operation (e.g., *Scale In* or *Scale Out*) was interrupted.

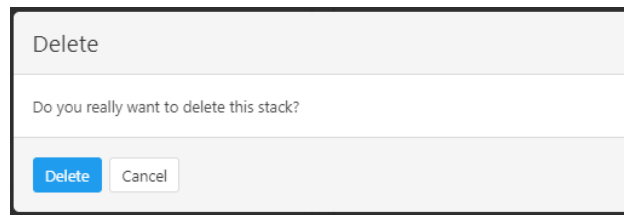
In addition to the above, for stacks that have Automatic Healing enabled, the operational state of all components is periodically monitored and *Stop*, *Start* or *Rebuild* operations are triggered if needed.

The automatic healing logs are collected in the *auto-job* log, which can be viewed through the Web or CLI management interfaces.

3.18 Deleting Stack

The *Delete* operation deletes the stack and releases all resources allocated during its creation.

Figure 3-44: Deleting a Stack



3.19 Rebuilding Stack

The *Rebuild* operation rebuilds specific stack components. The command is typically used when specific stack components stop operating correctly and their operation can't be restored through regular backup/restore procedures.

Component names must be explicitly specified as the *Rebuild* operation parameter, for example:

- sc-1: Rebuilds the first Signaling Component instance
- mc-1,mc-2: Rebuilds the first two Media Component instances
- sc: Rebuilds all Signaling Component instances
- mc: Rebuilds all Media Component instances
- sbc-1: Rebuilds the first Mediant VE instance

The *Rebuild* operation deletes the corresponding virtual machines and creates new ones instead of them. Network interfaces are preserved and therefore, both private and public IP addresses remain unchanged.

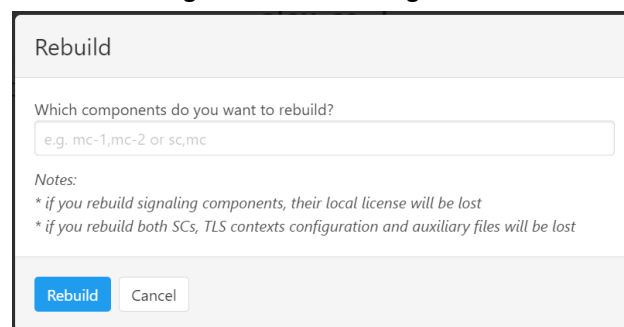
During the *rebuild* operation, the serial number of the rebuilt instances changes and therefore, their local license is lost. Obtain, activate and apply the new license to the rebuilt components to restore their service. Note that Media Components don't have a local license and therefore, this limitation doesn't apply to them.

The *Rebuild* operation uses a default Marketplace image for new instances initialization. As soon as these instances come up and establish connection with other cluster components, they automatically update their software version and align to the current stack configuration.

If you rebuild *both* Signalling Component of Mediant CE stack or all components of Mediant VE stack, the following parts of the SBC configuration will be lost and need to be manually restored from backup:

- TLS Contexts configuration (private key and certificates)
- Auxiliary files (e.g., Pre-recorded Tone files)

Figure 3-45: Rebuilding Stack



3.20 Managing Files

Stack Manager versions 3.5.0 and later implement Files Repository that allows you to add multiple CMP files it.

CMP files added to the files repository can be used during the **Upgrade** operation, as described in Section 3.21.1, Hosting Software Load (CMP) Files on Stack Manager.



CMP files added to Stack Manager's Files Repository are publicly accessible through the following URL: `<stack-mgr-base-url>/files/<filename>`

To add a file to Stack Manager's file repository:

1. Navigate to the **Files** screen.
2. Click **Add file**.
3. Choose the SBC software load (CMP) file, and then click **Add**.
4. Wait until the file upload completes.

Figure 3-46: Files Screen

Name	Type	Added on	Added by	Status	Used by	Action
HostedTP_CENTOS8_SIP_F7.40LN.501.112.cmp	cmp	Jul 25, 2024 07:12:14	audc	ok		Delete
HostedTP_CENTOS8_SIP_F7.40VA.500.044.cmp	cmp	Jul 08, 2024 15:49:52	unknown	ok		Delete

3.21 Upgrading Stack

The *Upgrade* operation upgrades all stack components, using a software load (CMP) file stored on some HTTP/HTTPS server.

It's especially useful for Mediant CE stacks, allowing upgrade via a single (although lengthy) operation, instead of the regular upgrade procedure that consists of multiple steps performed via various management interfaces, namely:

- Upgrade Signaling Components: using the Software Upgrade wizard in Mediant CE's Web interface
- Upgrade "active" (currently running) Media Components: using the Cluster Management page in Mediant CE's Web interface
- Upgrade "idle" (currently stopped) Media Components: using Stack Manager, as described in Section 3.21.2



The *Upgrade* operation does not support transition between software loads based on different OS versions (e.g., from software load based on CentOS 6 to a load based on CentOS 8). This is because such upgrade requires the use of a different image and can't be performed using a CMP file. Use the *Modify* and *Update* operations instead to perform such a transition. Refer to the *Mediant VE / CE Installation Manuals* for detailed instructions.

The *Upgrade* operation requires a software load (CMP) file to be available on some HTTP/HTTPS server and accessible by both Stack Manager and Mediant VE/CE stack components. You would typically use cloud-native storage services (e.g., AWS S3 or Azure Storage) for this purpose. Each Mediant VE/CE component accesses the specified URL directly, using its management interface. Therefore, you need to make sure that your network topology and security rules allow such access.

You can optionally specify which components you want to upgrade:

- `sc`: upgrades Signaling Components
- `mc`: upgrades Media Components
- `sc,mc`: upgrades all components

You can choose whether to upgrade Signaling Components using the hitless upgrade procedure (that upgrades them one by one while preserving service) or not. You can also specify a graceful timeout for Media Components upgrade, during which new calls will not be allocated to the Media Components, but existing calls will be allowed to end prior to starting the upgrade. Note that this value affects the total upgrade time and therefore, it's recommended to set it to a relatively low value.

Figure 3-47: Upgrading Mediant CE Stack via CMP File Hosted on External Server



If you host the CMP file on an external server, you should specify the complete URL in the 'Software (CMP) URL' field, as in the example above, and leave the 'CMP file' drop-down list as **--none--**.

3.21.1 Hosting Software Load (CMP) Files on Stack Manager

You can optionally use Stack Manager to host the software load (CMP) file that will be used to upgrade the Mediant VE/CE components. This can be useful if, for example, Mediant VE/CE components are configured to use private IP addresses on management interfaces and therefore, are unable to access cloud-native storage services through public IP addresses.



The upgrade of Mediant VE/CE software involves the download of the CMP file from a publicly available URL. For this communication, Mediant VE/CE serves as an HTTP client.

If you decide to host the CMP file on the Stack Manager, consider the following:

- CMP files hosted on Stack Manager are publicly accessible.
- Make sure that Stack Manager is reachable from the deployed Mediant VE/CE stack through the HTTP or HTTPS protocol.

To use CMP file hosted on Stack Manager during Stack upgrade:

1. Add the CMP file to Stack Manager's file repository, as described in Section 3.20, Managing Files.
2. Navigate to the **Stacks** screen.
3. Choose the stack that you want to upgrade.
4. Click **More > Upgrade**.
5. The **Software (CMP) File URL** field is automatically populated with Stack Manager's base URL. Update this field as needed. Make sure that the value ends with a forward slash ("/").
6. Choose the **CMP file** that you added in Step 1.
7. Choose the upgrade mode.
8. Click **Upgrade** to start the upgrade.

Figure 3-48: Upgrading Mediant VE stack via CMP File hosted on Stack Manager

Upgrade

Software (CMP) URL
http://10.95.1.7/files/

CMP file
HostedTP_CENTOS8_SIP_F7.40LN.501.112.cmp

Hitless upgrade

Upgrade Cancel

3.21.2 Upgrading Software on Idle Media Components



This section is applicable only to Mediant CE stacks.

When software upgrade of Media Components is performed through the Mediant CE's Web interface (**Setup > IP Network > Cluster Manager Settings > Start Upgrade**), as described in the *Mediant Software User's Manual*, it applies only to "active" Media Components (that are in "started" state).

To complete upgrade for "idle" Media Components (that are in "stopped" state), click the **More > Update Idle MCs** button on the toolbar.

The operation temporarily starts "idle" Media Components, waits until they complete software upgrade, and then shuts them down.

3.22 Shelving and Unshelving Stack

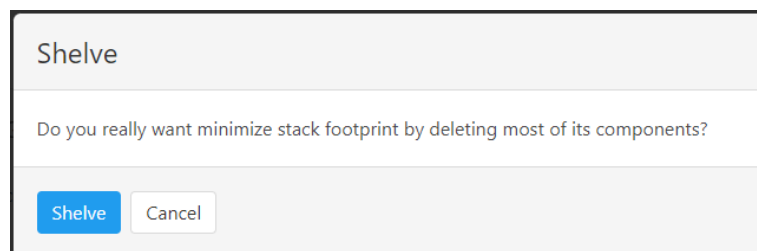
If you are not using Mediant VE or CE stack for a relatively long amount of time (e.g. in lab environment) you can use *Shelve* action to reduce stack footprint and minimize infrastructure (cloud) costs.

Shelve operation is available for Mediant VE and CE stacks only and does the following:

- Deletes Media Components' virtual machines (for CE stacks)
- Deletes Load Balancers in Azure environment
- (Optionally) Deletes Public IPs according to the **Delete Public IPs During Shelve** global configuration parameter

Signaling components, network interfaces, and complete Mediant VE / CE configuration (including INI file, license key, TLS keys and certificates, auxiliary files) are preserved.

Figure 3-49: Shelving Stack

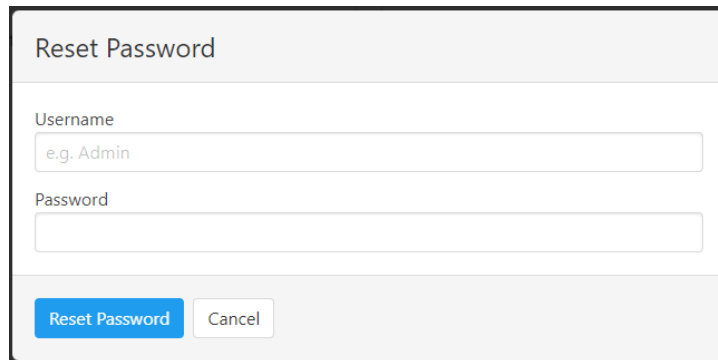


Use *Unshelve* operation to restore deleted components and bring stack to fully operational state.

3.23 Resetting Stack Password

If you forgot login credentials to the deployed Mediant VE / CE stack, you can use *Reset Password* operation to configure new credentials.

Figure 3-50: Resetting Stack Password



Reset Password

Username
e.g. Admin

Password

Reset Password Cancel

Note that for Mediant VE and CE stacks, Stack Manager creates a dedicated *StackMgr* user with a randomized password during stack deployment and uses it to communicate with the deployed stack (via REST API). The *Reset Password* operation doesn't change these credentials; instead, it changes the "admin" username / password used for interactive login to the SBC's Web / CLI management interfaces. If you want to change credentials used by Stack Manager, refer to Section 3.9.4, Updating Connectivity.

3.24 Sending INI File

Use *Send INI File* operation to update configuration of the deployed Mediant CE / CE stack via incremental INI file.

Figure 3-51: Sending INI File



Send INI File

Send incremental INI file to the SBC

Choose File No file chosen

Send Cancel

In an Azure environment, you can also configure the INI Repository through the corresponding global configuration parameter, as described in Section 3.4.4, Microsoft Azure Parameters. If you do so, the Send INI File dialog box also allows you to choose the INI file from the repository.

3.25 Stack Deployment Details

This section describes the methods that Stack Manager uses to deploy stacks in different virtualization environments. Understanding these details allows you to monitor stack behavior using the virtualization environment's management interfaces (e.g., AWS dashboard) and to troubleshoot various abnormal scenarios. It's also needed to alter some stack configuration, as described in Section 3.25.2, Adjusting Security Groups.

3.25.1 Deployment Method

Stack Manager uses native cloud orchestration services to perform stack deployment. This simplifies deployment of multiple stack components and provides tracking for all resources that correspond to the specific stack. Specifically the following services are used:

Virtual Environment	Orchestration Service
Amazon Web Services (AWS)	Cloud Formation
Microsoft Azure	Azure Resource Manager
Google Cloud	Deployment Manager
OpenStack	Heat Orchestration Service

In AWS, Google Cloud and OpenStack, multiple orchestration templates are used, depending on the stack type. For example, for Mediant CE deployment in AWS the following native stacks are created:

- **<stack_name>-network:** Creates security groups and the cluster interface of Signaling Components
- **<stack_name>-sc:** Creates Signaling Component instance(s)
- **<stack_name>-mc-N:** Creates Media Component instance mc-N (where N is 1, 2, etc.)

In Azure, a single Resource Group **<stack_name>** is used and all stack resources are deployed in it.

Figure 3-52: Cloud Formation Stacks for Mediant CE in AWS

The screenshot shows the AWS CloudFormation console interface. At the top, there are navigation tabs for 'Services', 'Resource Groups', and 'Stacks'. Below the tabs, there are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter is set to 'Active' and the search term is 'stack1'. The table below shows 7 stacks with columns for Stack Name, Created Time, Status, and Description.

Stack Name	Created Time	Status	Description
stack1-sc	2018-05-21 14:27:19 UTC+0300	CREATE_COMPLETE	SBC Cluster - Signaling Component
stack1-mc-5	2018-05-21 14:26:01 UTC+0300	UPDATE_COMPLETE	SBC Cluster - Media Component
stack1-mc-4	2018-05-21 14:26:01 UTC+0300	UPDATE_COMPLETE	SBC Cluster - Media Component
stack1-mc-3	2018-05-21 14:26:00 UTC+0300	UPDATE_COMPLETE	SBC Cluster - Media Component
stack1-mc-2	2018-05-21 14:24:42 UTC+0300	UPDATE_COMPLETE	SBC Cluster - Media Component
stack1-mc-1	2018-05-21 14:24:42 UTC+0300	UPDATE_COMPLETE	SBC Cluster - Media Component
stack1-network	2018-05-21 14:24:11 UTC+0300	CREATE_COMPLETE	SBC Cluster - Network Resources

Once all components are created, Stack Manager manages their state (for example, the state of Media Components) by stopping and starting corresponding instances. Instances that correspond to "active" Media Components are "started" and are expected to be in the "running" state. Instances that correspond to "inactive" Media Components are "stopped" and are expected to be in the "stopped" state.

Stack Manager implements the *Update* command by altering the resource state to match the updated configuration. Implementation in AWS, Google Cloud, and OpenStack updates the native orchestration template(s) and issues the *Update* to the specific native stack(s). Implementation in Azure directly modifies the state of corresponding native resources.

3.25.2 Adjusting Security Groups

Stack Manager creates Security Groups required for normal Mediant VE/CE operation during stack creation. A list of allowed inbound ports is specified via “Management ports”, “Signaling ports” and “Media ports” configuration parameters during stack creation.

If you need to adjust this configuration after the stack is created, for example, to allow signaling traffic on additional ports, use the *Modify* operation to change these configuration parameters and then *Update* to apply the changes.

Alternatively, you can create your own Security Groups and specify them using the Advanced Config parameters (e.g., “main_nsg_id” or “nsg_id_sc_ethX”). See Section 3.8.11, Advanced Configuration for more information.

For additional information and for a detailed list of rules in each Security Group, refer to *Mediant Virtual Edition for AWS/Azure/Google Installation Manual* and to *Mediant Cloud Edition Installation Manual*.

3.25.2.1 Modifying Security Groups Created by Stack Manager in Azure Environment

In an Azure environment, you can modify Security Groups created by Stack Manager through the Azure portal, CLI or PowerShell, by removing some rules and adding matching rules with the “keep-” prefix in their names. Such rules are preserved during the *Update* operation and if they specify the same protocol / port as a standard rule, standard rule won't be created.

For example, if you want to limit access to the SSH interface to specific IP addresses, remove the “22-tcp” rule created by Stack Manager and instead, create one or more rules with the “keep-” prefix (e.g., “keep-ssh”) and specify port 22, protocol TCP and a list of IP addresses allowed to access the SSH interface.

You can also use the “keep-” prefix for custom Load Balancer configuration, specifically for frontend IP configurations, backend pools and routing rules. For example, to temporarily expose access to the Mediant VE/CE Web interface via a public IP address, add a rule with the “keep-” prefix, and specify port 80 (or 443) and protocol TCP.

Rules with the “keep-” prefix are preserved during the *Update* operation. However, you must make sure that they don't overlap / contradict standard configuration created by Stack Manager.



Modification of Network Security Groups created by Stack Manager in AWS, Google Cloud, and OpenStack environments is not supported. If you make changes, they may disappear after you change Mediant VE/CE configuration. Use custom security groups instead and specify them using the Advanced Config parameters (e.g., “main_nsg_id” or “nsg_id_sc_ethX”). See Section 3.8.11, Advanced Configuration for more information.

3.25.3 Using Pre-Defined Public IP Addresses

Stack Manager assigns Public (Elastic/External/Floating) IP addresses to deployed components based on the **Public IPs** configuration parameter and **sc_public_ips**, **mc_public_ips** and **public_ips** advanced configuration parameters, as described in Section 3.8.11, Advanced Configuration.

By default, it allocates new Public IP addresses and assigns them to the instances.

If you want to use pre-defined Public IP addresses instead, you need to add the following parameters to stack's Advanced Config section:

```
public_ip_<component name>_<interface name> = <ID>
```

where:

- <component name> is the name of the component to which you want to assign the pre-defined Elastic IP address. Valid component names are:
 - **Mediant CE:**
 - ◆ **Signaling Components:** Specify "sc" (and not "sc-1" / "sc-2"). This defines a "floating" IP address that is logically attached to the active Signaling Component. The exact implementation differs from cloud to cloud. For example, for AWS these are Elastic IP addresses that float across EC2 Instances, while for Azure and Google Cloud, these public IP addresses are assigned to the Load Balancer.
 - ◆ **Media Components:** Specify the full name of the Media Component(e.g., "mc-1", "mc-2", etc.).
 - **Mediant VE:**
 - ◆ Skip the <component_name> part and specify public_ip_<interface_name> instead.
 - ◆ For Mediant VE HA deployment in Azure cloud, the above defines the public IP address for signaling streams. If you want to specify public IP addresses for media streams, use "sbc-1" or "sbc-2" as the component name.
- <interface name> is the name of the network interface to which you want to assign pre-defined Elastic IP addresses; for example "eth0", "eth1", etc.
- <ID> is the environment-specific Public IP address identifier:
 - AWS: Allocation ID of pre-defined Elastic IP address
 - Azure: Resource Group/Name of pre-defined Public IP address
 - Google and OpenStack: Pre-defined external/floating IP address

For example:

Mediant CE in AWS:

```
public_ip_sc_eth1 = eipalloc-461b3468
public_ip_mc-1_eth1 = eipalloc-37818019
public_ip_mc-2_eth1 = eipalloc-f51f1edb
```

Mediant CE in Azure:

```
public_ip_sc_eth1 = CelResourceGroup/ScPublicIP
public_ip_mc-1_eth1 = CelResourceGroup/Mc1PublicIP
```

Mediant VE in AWS:

```
public_ip_eth1 = eipalloc-461b3468
```

Mediant VE HA in Azure:

```
public_ip_eth1 = VeResourceGroup/SignalingPublicIP
```

```
public_ip_sbc-1_eth1 = VeResourceGroup/Media1PublicIP  
public_ip_sbc-2_eth1 = VeResourceGroup/Media2PublicIP
```

Stack Manager uses pre-defined Public IP addresses for all user-defined components/interfaces as per the above configuration and allocates new Public IP addresses for all the rest.

If multiple public IP addresses are configured on the same interface via **sc_public_ips** advanced configuration parameter, multiple pre-allocated addresses can be specified in the <ID> element as a comma-separated list. For example:

```
public_ip_sc_eth1 = eipalloc-461b3468,eipalloc-37818019
```

If you have an existing Mediant VE / CE stack and want to view its currently allocated public IP addresses or determine the advanced config parameter name that can be used to modify them, use the “Show IP Addresses” action, as described in Section 3.9.1. The output includes the “Advanced Config” section with all the relevant parameters and currently used values.

3.25.4 Using Pre-Defined Private IP Addresses

Stack Manager assigns Private IP addresses to deployed components based on configured network interfaces and **sc_additional_ips**, **mc_additional_ips** and **additional_ips** advanced configuration parameters, as described in Section 3.8.11, Advanced Configuration.

By default, IP addresses are dynamically allocated from the corresponding subnets.

If you want to specify static private IP addresses instead, you need to add the following parameters to stack's Advanced Config section:

```
private_ip_<component name>_<interface name> = <private IPs>
```

where:

- <component name> is the name of the component to which you want to assign pre-defined private IP addresses. Valid component names are:
 - **Mediant CE:** "sc-1", "sc-2", "mc-1", "mc-2", etc. For Azure, you can also use the "sc" component name to specify a pre-defined private IP address for the Internal Load Balancer.
 - **Mediant VE:** "sbc-1", "sbc-2". For Azure, you can also skip the component name to specify a pre-defined private IP address for the Internal Load Balancer.
- <interface name> is the name of the network interface to which you want to assign pre-defined private IP addresses for example "eth0", "eth1", etc.
- <private IPs> is a comma-separated list of private IP addresses. The first address is the primary address while additional addresses are secondary addresses.

The **private_ip_...** configuration parameters must specify *all* private IP addresses on the specific network interface of the specific instance. It's impossible to configure some IP addresses of the network interface statically and allocate others dynamically.

Adhere to the following rules when using the **private_ip_...** configuration parameter:

- **Mediant CE in AWS:**
 - For "sc-1":
 - ◆ "eth0" must have two IP addresses. The first IP address is used as the HA interface and the second as the Cluster interface.
 - ◆ For single-zone deployments, other interfaces must have two IP addresses plus as many additional IP addresses, as specified by **sc_public_ips** and **sc_additional_ips** advanced configuration parameters. The first IP address is configured as the primary ENI address and is not used by the SBC application.
 - ◆ For multi-zone deployments, other interfaces must have one IP address plus as many additional IP addresses, as specified by **sc_public_ips** and **sc_additional_ips** advanced configuration parameters.
 - For "sc-2":
 - ◆ "eth0" must have one IP address, which is used as the HA interface.
 - ◆ Other interfaces must have one IP address. These IP addresses are configured as the primary ENI address and are not used by the SBC application for single-zone deployments.
 - For "mc-1", "mc-2" etc.:
 - ◆ "eth0" must have one IP address, which is used as the Cluster interface.
 - ◆ Other interfaces must have one IP address plus as many additional IP addresses, as specified by **mc_public_ips** and **mc_additional_ips** advanced configuration parameters.

■ Mediant CE in Azure:

- For "sc-1" and "sc-2":
 - ◆ "eth0" must have two IP addresses. The first IP address is used as the HA interface and the second as the Cluster interface.
 - ◆ Other interfaces must have one IP address plus as many additional IP addresses, as specified by **sc_public_ips** and **sc_additional_ips** advanced configuration parameters.
- For "sc":
 - ◆ Applicable only to configurations that use an Internal Load Balancer.
 - ◆ Specified IP address is assigned to the Internal Load Balancer interface.
 - ◆ For each applicable interface, one IP address must be specified.
- For "mc-1", "mc-2" etc.:
 - ◆ "eth0" must have one IP address, which is used as the Cluster interface.
 - ◆ Other interfaces must have one IP address plus as many additional IP addresses, as specified by **mc_public_ips** and **mc_additional_ips** advanced configuration parameters.

■ Mediant CE in Google Cloud:

- For "sc-1" and "sc-2":
 - ◆ "eth1" must have two IP addresses. The first IP address is used as the HA interface and the second as the Cluster interface.
 - ◆ Other interfaces must have one IP address plus as many additional IP addresses, as specified by **sc_public_ips** and **sc_additional_ips** advanced configuration parameters.
- For "mc-1", "mc-2" etc.:
 - ◆ "eth1" must have one IP address, which is used as the Cluster interface.
 - ◆ Other interfaces must have one IP address plus as many additional IP addresses, as specified by **mc_public_ips** and **mc_additional_ips** advanced configuration parameters.

■ Mediant CE in OpenStack:

- For "sc-1":
 - ◆ "eth0" must have two IP addresses. The first IP address is used as the HA interface and the second as the Cluster interface.
 - ◆ Other interfaces must have one IP address.
- For "sc-2":
 - ◆ "eth0" must have one IP address, which is used as the HA interface.
 - ◆ Other interfaces must have one IP address. These addresses are not used by the SBC application.
- For "mc-1", "mc-2" etc.:
 - ◆ "eth0" must have one IP address, which is used as the Cluster interface.
 - ◆ Other interfaces must have one IP address.

■ Mediant VE in AWS:

- If HA is enabled:
 - ◆ For "sbc-1":
 - "eth0" must have one IP address, which is used as the HA interface.

- For single-zone deployments, other interfaces must have two IP addresses plus as many additional IP addresses, as specified by **public_ips** and **additional_ips** advanced configuration parameters. The first IP address is configured as the primary ENI address and is not used by the SBC application.
- For multi-zone deployments, other interfaces must have one IP address plus as many additional IP addresses, as specified by **public_ips** and **additional_ips** advanced configuration parameters.
- ◆ For "sbc-2":
 - "eth0" must have one IP address, which is used as the HA interface.
 - Other interfaces must have one IP address. These IP addresses are configured as the primary ENI address and are not used by the SBC application for single-zone deployments.
- If HA is disabled:
 - ◆ For "sbc-1":
 - Each interface must have one IP address plus as many additional IP addresses, as specified by **public_ips** and **additional_ips** advanced configuration parameters.
- **Mediant VE in Azure:**
 - If HA is enabled:
 - ◆ For "sbc-1" and "sbc-2":
 - "eth0" must have one IP address, which is used as the HA interface.
 - Other interfaces must have one IP addresses and as many additional IP addresses, as specified by **public_ips** and **additional_ips** advanced configuration parameters.
 - ◆ For empty component name:
 - Applicable only to configurations that use an Internal Load Balancer.
 - Specified IP address is assigned to the Internal Load Balancer interface.
 - For each applicable interface, one IP address must be specified.
 - If HA is disabled:
 - ◆ For "sbc-1":
 - Each interface must have one IP address and as many additional IP addresses, as specified by **public_ips** and **additional_ips** advanced configuration parameters.
- **Mediant VE in Google Cloud:**
 - If HA is enabled:
 - ◆ For "sbc-1" and "sbc-2":
 - "eth1" must have one IP address, which is used as the HA interface.
 - Other interfaces must have one IP address plus as many additional IP addresses, as specified by **public_ips** and **additional_ips** advanced configuration parameters.
 - If HA is disabled:
 - ◆ For "sbc-1":
 - Each interface must have one IP address plus as many additional IP addresses as specified by **public_ips** and **additional_ips** advanced configuration parameters.

For example:

```
private_ip_sc-1_eth0 = 172.31.128.1,172.31.129.1
private_ip_sc-1_eth1 = 172.31.68.1,172.31.69.1
private_ip_sc-1_eth2 = 172.31.78.1,172.31.79.1

private_ip_sc-2_eth0 = 172.31.128.2
private_ip_sc-2_eth1 = 172.31.68.2
private_ip_sc-2_eth2 = 172.31.78.2

private_ip_mc-1_eth0 = 172.31.128.101
private_ip_mc-1_eth1 = 172.31.68.101
private_ip_mc-1_eth2 = 172.31.78.101

private_ip_mc-2_eth0 = 172.31.128.102
private_ip_mc-2_eth1 = 172.31.68.102
private_ip_mc-2_eth2 = 172.31.78.102
```

If you have existing Mediant VE / CE stack and want to view its currently allocated private IP addresses or determine the exact advanced config parameter name that can modify them, use the “Show IP Addresses” action, as described in Section 3.9.1. The output includes the “Advanced Config” section with all the relevant parameters and currently used values.

3.25.5 Using Pre-Defined Virtual IP Addresses

For multi-zone Mediant VE and CE deployments in an AWS environment, Stack Manager uses Virtual IP addresses to enable communication inside the VPC or via the Transit Gateway. Virtual IP addresses must reside outside the VPC CIDR range and are manually plugged into the routing tables of the corresponding subnets. Refer to [Mediant VE SBC for AWS Installation Manual](#) or [Mediant CE SBC for AWS Installation Manual](#) for more information.

By default, Stack Manager automatically allocates Virtual IP addresses from the 169.254.64.0/24 subnet. If you want to specify your virtual IP addresses, add the following parameters to stack's Advanced Config section:

```
virtual_ip_<component name>_<interface name> = <virtual IP>
```

where:

- <component name> is the name of the component to which you want to assign virtual IP addresses. Valid component names are:
 - **Mediant CE:** "sc"
 - **Mediant VE:** skip <component_name> part and specify `virtual_ip_<interface_name>` instead
- <interface name> is the name of the network interface to which you want to assign pre-defined private IP addresses (e.g., "eth0", "eth1", etc.).
- <virtual IP> is the virtual IP address that you want to use for the specific network interface.

For example:

```
Mediant CE in AWS:  
  virtual_ip_sc_eth2 = 10.1.5.11  
  virtual_ip_sc_eth3 = 10.1.5.12
```

```
Mediant VE in AWS:  
  virtual_ip_eth2 = 10.1.5.15
```

3.25.6 Using Pre-Defined IPv6 Addresses

Stack Manager assigns IPv6 addresses to Mediant VE / CE stacks in AWS and Azure environments, according to the `sc_ipv6_ips`, `mc_ipv6_ips` and `ipv6_ips` Advanced Configuration parameters, as described in Section 3.8.11, Advanced Configuration.



You must use SBC version 7.40A.500.700 or later for proper IPv6 support.

By default, IPv6 addresses are dynamically allocated from the corresponding subnets. If you want to specify static IPv6 addresses instead, do the following:

- **AWS:** Use the following advanced config parameter:

```
ipv6_ip_<component name>_<interface name> = <IPv6 address>
```

where:

- <component name> is the name of the component to which you want to assign pre-defined private IP addresses. Valid component names are:
 - ◆ **Mediant CE:** "sc-1", "sc-2", "mc-1", "mc-2", etc.
 - ◆ **Mediant VE:** "sbc-1", "sbc-2".
 - <interface name> is the name of the network interface to which you want to assign pre-defined private IP addresses, for example, "eth0", "eth1", etc.
 - <IPv6 address> is the IPv6 address to be used.
- **Azure:** Use the `public_ip_*` or `private_ip_*` advanced config parameters, as described in Sections 3.25.3 and 3.25.4.

If you have an existing Mediant VE / CE stack and you want to view its currently allocated IPv6 addresses or determine the exact advanced config parameter name that can modify them, use the "Show IP Addresses" action, as described in Section 3.9.1. The output includes the "Advanced Config" section with all the relevant parameters and currently used values.

4 CLI Interface

4.1 Accessing CLI Interface

Stack Manager's CLI is accessed by switching to the *stack_mgr* user, using the following command:

```
$ stack_mgr_cli
```

If the above command doesn't work, use the following alternative command to do the same:

```
$ sudo su - stack_mgr
```

4.2 Invocation

Most of the Stack Manager CLI is provided using the **stack_mgr** command.

Auto-completion is available for sub-commands and optional parameters.

4.3 Usage Information

Brief usage information is provided by running the **stack_mgr** command without arguments:

```
$ stack_mgr

usage: stack_mgr [-h] [--version]
               {create,delete,list,show,scale-out,scale-in,
               scale,heal,auto-scale,auto-job,modify,
               update,stop,start,upgrade,rebuild,purge,
               configure,log}
               ...
```

More detailed usage information is provided when '-h' or '--help' arguments are specified:

```
$ stack_mgr --help

usage: stack_mgr [-h] [--version]
               {create,delete,list,show,scale-out,scale-in,
               scale,heal,auto-scale,auto-job,modify,
               update,stop,start,upgrade,rebuild,purge,
               configure,log}
               ...

AudioCodes Stack Manager

positional arguments:
  {create,delete,list,show,scale-out,scale-in,scale,heal,auto-
  scale,auto-job,modify,update,stop,start,upgrade,
  rebuild,purge,configure,log}
  create                create stack
  delete                delete stack
  list                  list stacks
  show                  show stack
  scale-out             scale out stack
```

```

scale-in          scale in stack
scale             scale stack
heal             heal stack
auto-scale       auto-scale stack
auto-job         automatic job
modify           modify stack configuration
update           update stack
stop             stop stack
start            start stack
upgrade          upgrade stack
rebuild          rebuild stack components
purge            purge stack
configure        stack manager configuration
log              show logs

```

optional arguments:

```

-h, --help      show this help message and exit
--version       show program's version number and exit

```

4.4 Managing Users

Stack Manager versions 3.5.0 and later allow you to configure multiple users that are allowed access to the Web Interface and REST API.

See Section 3.3, Managing Users for more information.

```

$ stack_mgr user-list --help
usage: stack_mgr user-list [-h]

optional arguments:
  -h, --help  show this help message and exit

$ stack_mgr user-add --help
usage: stack_mgr user-add [-h] [--password PASSWORD]
                        [--password-hash PASSWORD_HASH]
                        [--type {sec-admin,admin,operator,monitor}]
                        [--status {active,change-password,locked}]
                        [--password-expiration PASSWORD_EXPIRATION]
                        name

positional arguments:
  name                user name

optional arguments:
  -h, --help          show this help message and exit
  --password PASSWORD password
  --password-hash PASSWORD_HASH

```

```

                                password hash
--type {sec-admin,admin,operator,monitor}
                                account type
--status {active,change-password,locked}
                                account status
--password-expiration PASSWORD_EXPIRATION
                                password expiration in days (0=never)

$ stack_mgr user-modify --help
usage: stack_mgr user-modify [-h] [--new-name NEW_NAME] [--
password PASSWORD]
                                [--password-hash PASSWORD_HASH]
                                [--type {sec-
admin,admin,operator,monitor}]
                                [--status {active,change-
password,locked}]
                                [--password-expiration
PASSWORD_EXPIRATION]
                                name

positional arguments:
  name                user name

optional arguments:
  -h, --help          show this help message and exit
  --new-name NEW_NAME  new user name
  --password PASSWORD  password
  --password-hash PASSWORD_HASH
                        password-hash
  --type {sec-admin,admin,operator,monitor}
                        account type
  --status {active,change-password,locked}
                        account status
  --password-expiration PASSWORD_EXPIRATION
                        password expiration in days (0=unlimited)

$ stack_mgr user-delete --help
usage: stack_mgr user-delete [-h] name

positional arguments:
  name                user name

optional arguments:
  -h, --help          show this help message and exit

```

4.5 Global Configuration

The **configure** command performs Stack Manager configuration:

```
$ stack_mgr configure --help
usage: stack_mgr configure [-h] [--aws-access-key ACCESS_KEY]
                          [--aws-secret-key SECRET_KEY]
                          [--aws-s3-bucket BUCKET]
                          [--name-prefix PREFIX]
                          [--azure-tenant-id ID]
                          [--azure-client-id ID]
                          [--azure-secret SECRET]
                          ...

optional arguments:
  -h, --help            show this help message and exit
  --aws-access-key AWS_ACCESS_KEY
                        AWS access key
  --aws-secret-key AWS_SECRET_KEY
                        AWS secret key
  --aws-s3-bucket AWS_S3_BUCKET
                        AWS S3 bucket name
  --name-prefix NAME_PREFIX
                        Prefix to be assigned to stacks and instances
  --azure-tenant-id AZURE_TENANT_ID
                        Azure tenant id
  --azure-client-id AZURE_CLIENT_ID
                        Azure client id
  --azure-secret AZURE_SECRET
                        Azure secret
  ...
```

To show current configuration, use the command without any arguments.

To update a specific configuration parameter(s), use the command with arguments.

4.6 Listing Available Stacks

The **list** command lists available stacks.

```
$ stack_mgr list --help
usage: stack_mgr list [-h] [--no-status]

optional arguments:
  -h, --help            show this help message and exit
  --no-status           do not show real-time status

$ stack_mgr list

+-----+-----+-----+-----+-----+
| name  | type          | vim  | state          | ip          |
+-----+-----+-----+-----+-----+
| stack1| sbc-cluster  | azure| running        | 51.143.59.195 |
| stack2| sbc-cluster  | azure| scaling-out    | 51.143.61.128 |
+-----+-----+-----+-----+-----+
```

4.7 Creating a New Stack

Creation of a new stack through CLI consists of the following steps:

1. Creating the stack configuration file, which can be done using one of the following methods:
 - SBC Cluster Configuration Tool (recommended) – see Section 4.7.1, Creating Stack Configuration File via SBC Cluster Configuration Tool (Recommended Method) for more information
 - Manually, by editing provided reference files – see Section 4.7.2, Creating Stack Configuration File Manually (Alternative Method) for more information
2. Creating the stack by the **create** command.

The stack configuration file contains configuration parameters of the created stack. The same configuration file can be used to create multiple stacks.

4.7.1 Creating Stack Configuration File via SBC Cluster Configuration Tool (Recommended Method)

The SBC Cluster Configuration Tool provides a simple interactive user interface (UI) for creating the configuration file.

- To create the stack configuration file for Mediant CE, type **sbc_cluster_config**
- To create the stack configuration file for Mediant VE, type **sbc_config**

You are prompted for the basic Mediant VE/CE configuration parameters, and a new configuration file will be created. You can use this file to create the Mediant VE/CE instance using the **stack_mgr create** command, as described in Section 4.7.3, Creating a New Stack. It's recommended to review the created file prior to the instance creation and modify it if needed.



The following output is provided as an example only and therefore, might not be up to date.

```
$ sbc_cluster_config
-----
SBC Cluster Configuration Tool
-----

This tool creates configuration file that may be used to create
the Mediant CE cluster via "stack_mgr create" command.

Enter configuration file name: stack1.cfg

Virtual environments:
+---+-----+
| # | vim      |
+---+-----+
```

```

| 1 | aws      |
| 2 | azure    |
| 3 | google   |
| 4 | openstack|
+---+-----+

Choose virtual environment: 1

List of AWS regions:
+---+-----+
| # | name          |
+---+-----+
| 1 | ap-south-1    |
| 2 | ap-northeast-2|
| 3 | ap-southeast-1|
| 4 | ap-southeast-2|
| 5 | ap-northeast-1|
| 6 | ca-central-1  |
| 7 | eu-central-1  |
| 8 | eu-west-1     |
| 9 | eu-west-2     |
| 10| eu-west-3     |
| 11| eu-north-1    |
| 12| sa-east-1     |
| 13| us-east-1     |
| 14| us-east-2     |
| 15| us-west-1     |
| 16| us-west-2     |
+---+-----+

Choose region: 7

List of AWS VPCs:
+---+-----+-----+-----+
| # | id              | name          | cidr block    |
+---+-----+-----+-----+
| 1 | vpc-45f3152c   | DefaultVPC   | 172.31.0.0/16 |
| 2 | vpc-39d23352   | TestVPC      | 172.16.138.0/24|
+---+-----+-----+-----+

Choose VPC: 1

Key pair is used to provide secure access to the Mediant CE's CLI
interface
via SSH protocol. It is mandatory for AWS environment even though
SBC in its

```


default configuration supports SSH login using username/password.

```
+-----+-----+
| # | name |
+-----+-----+
| 1 | infra-key |
| 2 | sbc-ssh-key |
| 3 | test-key |
+-----+-----+
```

Choose key pair: **2**

You must create IAM role that allows SBC to manage its IP addresses.

The role must look as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:AssignPrivateIpAddresses",
        "ec2:UnassignPrivateIpAddresses",
        "ec2:AssociateAddress",
        "ec2:DescribeAddresses",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Refer to the Mediant CE Installation Manual for more information.

Enter IAM role: **SBC-HA-3**

Mediant CE components may have 2, 3 or 4 network interfaces that are connected as follows:

```
+-----+-----+-----+
| iface | subnet | traffic |
+-----+-----+-----+
| eth0 | cluster | internal cluster communication |
| eth1 | main | management (HTTP, SSH) + signaling |
| | | SIP) + media (RTP) |
| eth2 | additional1 | signaling (SIP) + media (RTP) |
| eth3 | additional2 | signaling (SIP) + media (RTP) |
+-----+-----+-----+
```

```
Enter number of network interfaces (2, 3 or 4): 3
```

In order to communicate with signaling components from outside the AWS cloud Elastic IP addresses must be assigned to the relevant network interfaces.

Provide comma-separated list of SC network interfaces that will be assigned with Elastic IP addresses. Specify interface by corresponding subnet name.

For example: "main" or "main,additional1"

Notes:

- if you want to access management interface via Internet assign Elastic IP to "main" interface
- if all management and signaling communication happens inside the VPC and therefore you do not need Elastic IPs, press Enter to continue

```
Enter value: main
```

In order to communicate with media components from outside the AWS cloud Elastic IP addresses must be assigned to the relevant network interfaces.

Provide comma-separated list of MC network interfaces that will be assigned with Elastic IP addresses. Specify interface by corresponding subnet name.

For example: "main" or "main,additional1"

Notes:

- if all media communication happens inside the VPC and therefore you do not need Elastic IPs, press Enter to continue

```
Enter value: main
```

Cluster subnet carries internal traffic between SBC cluster components and is used for accessing AWS API. It must support outbound access to EC2 API - either via private EC2 API endpoint or via NAT Gateway configured as default route (refer to Mediant CE Installation Manual for more information). Use dedicated subnet and protect it from unauthorized access.

```
+---+-----+-----+-----+-----+
| # | id          | name    | cidr range      | avail zone      |
+---+-----+-----+-----+-----+
| 1 | subnet-5d2d | voip2   | 172.31.224.0/20 | eu-central-1b   |
```

2	subnet-ec6c	test	172.31.144.0/20	eu-central-1b	
3	subnet-09a2	cluster	172.31.80.0/20	eu-central-1b	
4	subnet-7c73		172.31.16.0/20	eu-central-1a	
5	subnet-4e08		172.31.32.0/20	eu-central-1c	
6	subnet-1538	oam	172.31.64.0/20	eu-central-1b	
8	subnet-fb63	voip1	172.31.0.0/20	eu-central-1b	

Cluster subnet: **3**

Main subnet carries management (HTTP, SSH, etc), signaling (SIP) and media (RTP) traffic.

#	id	name	cidr range	avail zone
1	subnet-5d2d	voip2	172.31.224.0/20	eu-central-1b
2	subnet-ec6c	test	172.31.144.0/20	eu-central-1b
3	subnet-09a2	cluster	172.31.80.0/20	eu-central-1b
4	subnet-1538	oam	172.31.64.0/20	eu-central-1b
5	subnet-fb63	voip1	172.31.0.0/20	eu-central-1b

Main subnet: **4**

Additional subnets (additional1, additional2) carry signaling (SIP) and media (RTP) traffic. It is possible to specify the same Subnet ID for both Main and additional subnets - in this case Mediant CE components will have multiple network interfaces (ENIs) connected to the same subnet.

#	id	name	cidr range	avail zone
1	subnet-5d2d	voip2	172.31.224.0/20	eu-central-1b
2	subnet-ec6c	test	172.31.144.0/20	eu-central-1b
3	subnet-09a2	cluster	172.31.80.0/20	eu-central-1b
4	subnet-1538	oam	172.31.64.0/20	eu-central-1b
5	subnet-fb63	voip1	172.31.0.0/20	eu-central-1b

Additional subnet: **5**

Instance type of Signaling Components (SC) is r5.2xlarge.
Instance type of Media Components (MC) depends on their profile.

```

+---+-----+-----+
| # | mc profile | instance type |
+---+-----+-----+
| 1 | forwarding | m5.large      |
| 2 | transcoding| c5.4xlarge    |
+---+-----+-----+

```

Choose media components profile: **1**

The size of the cluster, and specifically the number of media components, may vary to match the required service capacity. This ensures that the cluster utilizes optimal amount of resources at any point of time and elastically scales on demand.

The scaling decision may be done either manually - by executing 'scale-in' or 'scale-out' commands - or automatically based on the current cluster utilization.

The size of the cluster is controlled by the following two parameters:

- * Minimum Number of Media Components
- * Maximum Number of Media Components

To ensure the fast scaling, Stack Manager pre-creates all needed media components in advance (up to the maximum number) and stops/starts them accordingly during scale in/out operations.

Minimum Number of Media Components (0-21): **3**

Maximum Number of Media Components (3-21): **5**

Credentials for management interface.

Username: **sbcadmin**

Password: *********

Retype password: *********

Creating configuration file stack1.cfg

Done



When selecting the region, VPC, subnets and other listed objects, enter either a corresponding row number (e.g., "1") or an Object ID (e.g., "vpc-45f3152c").

4.7.2 Creating Stack Configuration File Manually (Alternative Method)

As an alternative to running the SBC Cluster Configuration Tool (described in the previous section), you can create the stack configuration file manually by copying it from the `/opt/stack_mgr/cfg` directory and then modifying it using a text editor tool.

You can edit the copied file in one of the following ways:

- On the server itself, by using, for example, a "vi" or "nano" editor:

```
$ cp /opt/stack_mgr/cfg/sbc-cluster-aws.cfg stack1.cfg
$ vi stack1.cfg
```

- By transferring the copied file from the server through SFTP/SCP to a computer, modifying it using a standard text editor (e.g., Notepad), and then transferring it back to the server.

When you create the stack configuration file manually, make sure that the following parameters are updated:

- **Amazon Web Services (AWS):**
 - **aws_region:** Defines the AWS region where the Mediant CE stack will be deployed.
 - **vpc_id:** Defines the VPC where the Mediant CE stack will be deployed.
 - ***_subnet_id:** Defines the subnet IDs for all applicable subnets.
 - **ssh_key_pair:** Defines the SSH key pair for connecting to the Mediant CE CLI.
 - ***_image_id:** Defines the AMI ID of the local copy of the Mediant VE/CE image.
 - **sc_iam_role:** Defines the SBC IAM Role name. Refer to the *Mediant Cloud Edition Installation Manual* for detailed instructions on how to create this role.
- **Microsoft Azure:**
 - **location:** Defines the Azure location where the Mediant CE stack will be deployed.
 - **vnet_id:** Defines the Virtual Network where the Mediant CE stack will be deployed.
 - ***_subnet_id:** Defines the subnet name for all applicable subnets.
- **Google Cloud:**
 - **region:** Defines the Google Cloud region where the Mediant CE stack will be deployed.
 - ***_subnet_id:** Defines the subnet name for all applicable subnets.
 - ***_image_id:** Defines the Image ID of the Mediant VE/CE image.
- **OpenStack:**
 - ***_subnet_id:** Defines the subnet name for all applicable subnets.
 - ***_image_id:** Defines the image name of the Mediant VE/CE image.
 - ***_instance_type:** Defines the flavor of the Mediant CE instances.

4.7.2.1 Sample Configuration File

The following is a sample configuration file for Mediant CE in the AWS cloud:



The file is provided as an example only and therefore, might not be up to date. Use files from the `/opt/stack_mgr/cfg` directory when creating a new stack configuration file.

```
# -----  
# Stack descriptor  
# -----  
  
# stack type  
stack_type = sbc-cluster  
  
# virtual infrastructure manager  
vim = aws  
  
# -----  
# Generic parameters  
# -----  
  
# Initial cluster size  
mc_num = 3  
  
# Minimal cluster size  
min_mc_num = 2  
  
# Maximum cluster size  
max_mc_num = 5  
  
# -----  
# Auto-scaling configuration  
# -----  
  
# Auto-scaling - enable/disable  
auto_scale = disable  
  
# Media utilization scale in threshold - in accumulative free  
# percentage points (when auto-scaling is enabled and total  
# amount of free resources in the cluster raises above this  
# threshold, scale-in is triggered)  
media_util_scale_in_threshold = 250  
  
# Media utilization scale out threshold - in accumulative free  
# percentage points (when auto-scaling is enabled and total  
# amount of free resources in the cluster falls below this  
# threshold, scale-in is triggered)  
media_util_scale_out_threshold = 100
```

```
# DSP utilization scale in threshold - in accumulative free
# percentage points (when auto-scaling is enabled and total
# amount of free resources in the cluster raises above this
# threshold, scale-in is triggered)
dsp_util_scale_in_threshold = 0

# DSP utilization scale out threshold - in accumulative free
# percentage points (when auto-scaling is enabled and total
# amount of free resources in the cluster falls below this
# threshold, scale-in is triggered)
dsp_util_scale_out_threshold = 0

# Auto-scaling cool down time in seconds
# (minimum time between two consecutive 'opposite' auto-scaling
# operations, e.g., scale-out after scale-in)
auto_scale_cooldown_time = 900

# Auto-scaling scale-in step
# (number of media instances to be removed)
auto_scale_in_step = 1

# Auto-scaling scale-out step
# (number of media instances to be added)
auto_scale_out_step = 1

# -----
# Network configuration
# -----

# AWS region name
# (use 'aws ec2 describe-regions' command to find all
# available regions)
aws_region = eu-central-1

# VPC where stack is deployed
vpc_id = vpc-45f3152c

# SBC cluster requires the following subnets:
# - cluster      - used for internal communication between
#                  cluster nodes
# - main         - used for management (HTTP, SSH), signaling
#                  (SIP) and media (RTP) traffic
# - additional1  - (optional) used for signaling (SIP) and
#                  media (RTP) traffic
# - additional2  - (optional) used for signaling (SIP) and
#                  media (RTP) traffic
#
# Notes:
# - during normal cluster operation only active Signaling
```

```
# component (SC) is accessed for management purposes (Web /
# CLI / SNMP / REST)
#
# It is perfectly fine to specify the same value for all below
# subnet_ids except for cluster_subnet_id.

cluster_subnet_id = subnet-be6e8bc3
main_subnet_id = subnet-1536d368
additional1_subnet_id =
additional2_subnet_id =

# Type of traffic supported in main subnet
# - all - management, signaling and media
# - oam - management only
main_subnet_traffic = all

# Key Pair provides secure access to the SBC cluster's
# CLI interface via SSH protocol. It is mandatory for the AWS
# environment even though SBC in its default configuration
# supports SSH login using username/password.
ssh_key_pair = aws_ssh_frankfurt_1

# -----
# Signaling Component (SC) configuration
# -----

# 1+1 HA mode - enable / disable
sc_ha_mode = enable

# Signaling Components (SC) network interfaces are connected
# as follows:
# - eth0: cluster
# - eth1: main
# - eth2: additional1
# - eth3: additional2
#
# At least two network interfaces are required.
# Notes:
# - Primary IP addresses are not used except for "eth0" (cluster
# interface). Secondary IP addresses are used instead and
# 'float' across the two SC instances (in HA configuration).

# Number of network interfaces - valid values: 2, 3, 4
sc_num_of_interfaces = 2

# Comma-separated list of network interfaces will be assigned
# with Public IP addresses (Elastic IPs) and optionally number of
# corresponding public IP addresses.
# Network interfaces are specified by corresponding subnet name
# (main, additional1, additional2) or by interface name (ethX -
```



```
# deprecated)
# For example:
#   "main,additional1" - assign one public IP address to
#                       interfaces connected to Main and
#                       1st Additional subnets
#   "main:2"           - assign two public IP addresses to
#                       interface connected to Main subnet
#   "main:2,additional1" - assign two public IP addresses to
#                       interface connected to Main subnet
#                       and one public IP address to interface
#                       connected to 1st Additional subnet
# Notes:
# - if you need to access SBC management interface via Internet
#   assign Public IP to interface connected to Main subnet
# - if all management and signaling communication happens inside
#   the VPC and therefore you do not need Public IPs, leave
#   this field blank
sc_public_ips = main

# Comma-separated list of network interfaces that will be assigned
# with additional private IP address and optionally, number of
# corresponding additional private IP addresses
#
# Network interfaces are specified by corresponding subnet name
# (main, additional1, additional2) or by interface name (ethX -
# deprecated)
# For example:
#   "main,additional1" - assign one additional private IP
#                       address to interfaces connected to
#                       Main and 1st Additional subnets
#   "main:2"           - assign two additional private IP
#                       addresses to interface connected to
#                       Main subnet
#   "main:2,additional1" - assign two additional private IP
#                       addresses to interface connected to
#                       Main subnet and one additional
#                       private IP address to interface
#                       connected to 1st Additional subnet
sc_additional_ips =

# AWS instance type
# if empty, r4/r5.2xlarge is used
sc_instance_type =

# AWS image id
# If empty, Marketplace image is used (default)
# For custom image, specify AMI ID (e.g. ami-9a50cff5)
sc_image_id =

# AWS IAM role that allows SC components to automatically
# configure network interfaces and perform switchover
```

```
sc_iam_role = SBC-HA-3

# URL of initial SBC cluster configuration file
# For example: "https://s3-eu-central-1.amazonaws.com/ac/sc.ini"
# If you don't have such URL, leave value blank
sc_ini_file_url =

# Configuration file contains Admin user - true / false
# (change this to "false" if your configuration file doesn't
# contain WebUsers table and you want the Stack Manager to
# automatically create default Admin user).
sc_ini_file_contains_admin_user = true

# Comma-separated list of tags (name=value) to be assigned to
# Signaling Components
# For example:
#   sc_tags = type=sbc,role=sc
sc_tags =

# Names for HA configuration
sc1_ha_name = sc-1
sc2_ha_name = sc-2

# Additional Signaling Components configuration parameters
# If you need to add a few additional parameters to SC
# configuration file specify them here. Use \n as line delimiter.
# For example:
#   sc_ini_params = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3
sc_ini_params =

# Comma-separated list of management ports
# (to configure network security group)
# Each list element may be one of the following:
#   <port>/<protocol>[/<cidr>]
#   where:
#       <port> is individual port number (e.g., 22) or port range
#       (e.g., 22-23)
#       <protocol> is tcp or udp
#       <cidr> is optional and can be IP address (e.g., 10.1.2.3)
#       or CIDR (e.g., 10.1.0.0/16)
sc_oam_ports = 22/tcp,80/tcp,443/tcp

# Comma-separated list of signaling ports
# (to configure network security group)
sc_signaling_ports = 5060/udp,5060/tcp,5061/tcp

# Comma-separated list of media ports
# (to configure network security group)
mc_media_ports = 6000-65535/udp

# -----
```

```
# Media Component (MC) configuration
# -----

# Media Components (MC) network interfaces are connected
# as follows:
#   - eth0: cluster
#   - eth1: main
#   - eth2: additional1
#   - eth3: additional2
#
# At least two network interfaces are required.
# Primary IP addresses are available on all interfaces.

# Number of network interfaces - valid values: 2, 3, 4
mc_num_of_interfaces = 2

# Comma-separated list of network interfaces will be assigned
# with Public IP addresses (Elastic IPs) and optionally number of
# corresponding public IP addresses.
# Network interfaces are specified by corresponding subnet name
# (main, additional1, additional2) or by interface name (ethX -
# deprecated)
# For example:
#   "main,additional1" - assign one public IP address to
#                       interfaces connected to Main and
#                       1st Additional subnets
#   "main:2"           - assign two public IP addresses to
#                       interface connected to Main subnet
#   "main:2,additional1" - assign two public IP addresses to
#                       interface connected to Main subnet
#                       and one public IP address to interface
#                       connected to 1st Additional subnet
# Notes:
#   - if all media communication happens inside the VPC and
#     therefore you do not need Public IPs, leave this field
#     blank
mc_public_ips = main

# Comma-separated list of network interfaces that will be assigned
# with additional private IP address and optionally number of
# corresponding additional private IP addresses
#
# Network interfaces are specified by corresponding subnet name
# (main, additional1, additional2) or by interface name (ethX -
# deprecated)
# For example:
#   "main,additional1" - assign one additional private IP
#                       address to interfaces connected to
#                       Main and 1st Additional subnets
#   "main:2"           - assign two additional private IP
#                       addresses to interface connected to
```

```
#                               Main subnet
#   "main:2,additional1" - assign two additional private IP
#                               addresses to interface connected to
#                               Main subnet and one additional
#                               private IP address to interface
#                               connected to 1st Additional subnet
mc_additional_ips =

# AWS instance type
# if empty:
#   - r4/m5.large or r4/m5.xlarge for media forwarding
#   - c4/c5.4xlarge for transcoding
mc_instance_type =

# AWS image id
# If empty, Marketplace image is used (default)
# For custom image, specify AMI ID (e.g. ami-9a50cff5)
mc_image_id =

# Media component profile - forwarding / transcoding
mc_profile = forwarding

# Media component max rate limit (in kpps)
# In addition to numeric values the following special string
# values are supported:
#   - "auto" means that PPS limit is automatically calculated
#     based on instance type
#   - "unlimited" means that no limit is imposed
mc_max_pps_limit = auto

# Comma-separated list of tags (name=value) to be assigned to
# media components
# For example:
#   mc_tags = type=svc,role=mc
mc_tags =

# Additional Media Components configuration parameters
# If you need to add a few additional parameters to MC
# configuration file specify them here. Use \n as line delimiter.
mc_ini_params =

# -----
# Additional configuration
# -----

# Prefix to be added to all created components
# (note that there is also global stack_mgr configuration
# parameter with a similar name, but this one overrides it if
# set to non-empty value)
name_prefix =
```

```
# Manage SBC cluster via HTTPS or HTTP protocol - valid values:
# enable / disable
# (change this to Disable if, for example, your firewall
# intercepts HTTPS connections and blocks them due to self-signed
# certificate being used)
manage_via_https = enable

# OS version - 6/8
os_type = 8

# Volume type for disks - valid values: gp2, gp3, io1, io2, st1,
scl, standard
volume_type = gp3
```

Sample configuration files for additional environments are available in the `/opt/stack_mgr/cfg` directory.

4.7.3 Creating a New Stack

After creating the stack configuration file, use the **create** command to create a new stack.

Specify the stack name and provide the stack configuration file.

```
$ stack_mgr create --help
usage: stack_mgr create [-h] name cfg_file

positional arguments:
  name                Name of the stack; may contain letters,
                    numbers and dash symbol only (spaces
                    are not allowed)
  cfg_file            configuration file

optional arguments:
  -h, --help          Show this help message and exit
```



Prior to creating a Mediant CE stack instance(s), make sure that all pre-requisites specified in the *Mediant Cloud Edition Installation Manual* are met. The document can be downloaded from AudioCodes website at <https://www.audiocodes.com/library/technical-documents>.

The *create* process takes a few minutes and detailed progress information is displayed on the console:

```
$ stack_mgr create stack1 sbc-cluster.cfg
Initializing AWS client... done
Creating SBC network resources..... done
Creating SBC media components..... done
Creating SBC signaling components..... done
Waiting until signaling components are ready..... done
Waiting until media components are ready.... done
```

```
Removing media components 'mc-3, mc-4, mc-5' from SBC
configuration
Removing media components 'mc-3, mc-4, mc-5'... done
Stopping components 'mc-3, mc-4, mc-5'..... done
U/52.58.15.164 to connect to the management interface.
Stack 'stack1' is successfully created
```

After the **create** command completes, you can connect to the Mediant CE's management interface through Web or SSH. The corresponding URL is shown in the summary following the stack creation.

Use the credentials provided in the stack configuration file to log in to the Mediant CE management interface.

4.8 Checking Stack State and Configuration

The **show** command displays detailed information about a specific stack.

You must specify a valid stack name.

```
$ stack_mgr show --help
usage: stack_mgr show [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  --no-status         do not show real-time status
  --idle-mcs          show 'idle' media components
```

```
$ stack_mgr show stack1
```

```
Name           : stack1
Type           : sbc-cluster
VIM            : aws
State          : idle

Created at     : April 09, 2021 08:55:29

Region        : eu-central-1
VPC           : vpc-45f3152c
```

```
-----
Signaling Components
-----
```

```
Instance type : r5.2xlarge
Image ID      :
```

```
+-----+-----+-----+-----+-----+
| id   | IP address   | status | type       | version     |
+-----+-----+-----+-----+-----+
| sc-1 | 172.31.65.177 | active | r5.2xlarge | 7.40A.005.314 |
```

```

| sc-2 |          | standby | r5.2xlarge | 7.40A.005.314 |
+-----+-----+-----+-----+-----+
Network configuration:
+-----+-----+-----+-----+
| interface | subnet      | id          | status |
+-----+-----+-----+-----+
| eth0      | cluster    | subnet-be6e8bc3 | in-use |
| eth1      | oam        | subnet-1536d368 | in-use |
| eth2      | additional1 |              |        |
| eth3      | additional2 |              |        |
+-----+-----+-----+-----+
SC number of network interfaces      : 2
SC interfaces with public IPs        : all
SC interfaces with additional IPs     :

-----
Media Components
-----

Instance type : m5.large
Image ID      :
Profile       : forwarding
Max rate limit : auto

+-----+-----+-----+-----+-----+-----+-----+
+-----+
| id   | IP address      | status   | %media | %dsp | type   |
| version |          |          |        |      |        |
+-----+-----+-----+-----+-----+-----+
+-----+
| mc-1 | 172.31.69.170 | connected | 0      | -    | m5.large |
| 7.40A.005.314 |          |          |        |      |          |
| mc-2 | 172.31.76.92  | connected | 0      | -    | m5.large |
| 7.40A.005.314 |          |          |        |      |          |
+-----+-----+-----+-----+-----+-----+
+-----+
Number of media components            : 2
Connected media components           : 2
Free media resources                  : 200%
Free DSP resources                    : -

Network configuration:
+-----+-----+-----+-----+
| interface | subnet      | id          | status |
+-----+-----+-----+-----+
| eth0      | cluster    | subnet-be6e8bc3 | in-use |
| eth1      | oam        | subnet-1536d368 | in-use |
| eth2      | additional1 |              |        |
| eth3      | additional2 |              |        |

```

```

+-----+-----+-----+-----+
MC number of network interfaces      : 2
MC interfaces with public IPs       : all
MC interfaces with additional IPs    :

Min number of media components      : 2
Max number of media components      : 10

Automatic scaling                   : enable
Media utilization scale in threshold : 250%
Media utilization scale out threshold : 100%
DSP utilization scale in threshold   : 0 (disabled)
DSP utilization scale out threshold  : 0 (disabled)
Automatic scaling cool down time    : 900 sec
Automatic scaling scale-out step    : 1
Automatic scaling scale-in step     : 1

Management IP address               : 52.58.15.164
Use HTTPS for cluster management    : enable

```

Unless the **--no-status** argument is specified, Stack Manager collects the following additional information:

- For Signaling Components:
 - Runtime status (running/stopped), using the cloud-specific API
 - Active instance that currently holds the "public IP", using the cloud-specific API
- For Media Components:
 - Runtime status (running/stopped), using the cloud-specific API
 - Connectivity status (connected/disconnected), using the SBC REST API
 - Media and DSP utilization, using the
 - SBC REST API

If the **--no-status** argument is specified or the Stack Manager fails to communicate with the SBC cluster, it displays an internal state of the component instead.

4.8.1 Checking Idle Media Components

The number and detailed status of Media Components reported by the **show** command corresponds to the "active" (running) Media Components. "Inactive" (stopped) Media Components can be viewed by adding the **--idle-mcs** argument to the **show** command, or by using the virtual environment's (e.g., AWS EC2) dashboard – corresponding instances are in the "stopped" state.

```
$ stack_mgr show stack1 --idle-mcs

...
-----
Media Components
-----

Instance type : m5.large
Image ID      : ami-d771563c
Profile       : forwarding
Max rate limit : auto

+-----+-----+-----+-----+-----+-----+
| id   | IP address   | status   | %media | %dsp | type   |
+-----+-----+-----+-----+-----+-----+
| mc-1 | 172.31.67.240 | connected | 0      | -    | m5.large |
| mc-2 | 172.31.67.15  | connected | 0      | -    | m5.large |
| mc-3 | 172.31.70.66  | down     | -      | -    | m5.large |
| mc-4 | 172.31.75.108 | down     | -      | -    | m5.large |
| mc-5 | 172.31.67.179 | down     | -      | -    | m5.large |
+-----+-----+-----+-----+-----+-----+

Number of media components      : 2
Connected media components     : 2
Free media resources            : 200%
Free DSP resources              : -
...
```

4.8.2 Viewing IP Addresses of Stack Components

Use the **show-ips** command to show IP addresses of stack components:

```
$ stack_mgr show-ips --help
usage: stack_mgr show-ips [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
```

```
~$ stack_mgr show-ips stack1

Fetching IP addresses of stack components... done
```

```

+-----+-----+-----+-----+
| component | interface | private IP address | public IP address |
+-----+-----+-----+-----+
| public-lb | eth1      |                    | 20.115.200.133    |
|           |           |                    |                   |
| sbc-1     | eth1      | 10.9.1.4           |                   |
|           | eth1:1    | 10.9.1.5           | 20.99.165.170    |
|           |           |                    |                   |
| sbc-2     | eth1      | 10.9.1.6           |                   |
|           | eth1:1    | 10.9.1.7           | 20.115.200.151   |
+-----+-----+-----+-----+

-----
Advanced Config
-----

public_ip_sbc_eth1 = Eitan-HA-VE/Eitan-HA-VE-sbc-eth1-ip
public_ip_sbc-1_eth1 = Eitan-HA-VE/Eitan-HA-VE-sbc-1-eth1-ip
private_ip_sbc-1_eth0 = 10.9.0.5
private_ip_sbc-1_eth1 = 10.9.1.4,10.9.1.5
public_ip_sbc-2_eth1 = Eitan-HA-VE/Eitan-HA-VE-sbc-2-eth1-ip
private_ip_sbc-2_eth0 = 10.9.0.4
private_ip_sbc-2_eth1 = 10.9.1.6,10.9.1.7

```

4.8.3 Checking Deployment Environment

Use the **check-env** command to check the deployment environment of Mediant VE and CE stacks deployed in an AWS environment. The operation provides a detailed summary of the deployment environment and tries to detect common (mis) configuration issues (e.g., lack of EC2 Endpoint or improper configuration of security group attached to it).

```

$ stack_mgr check-env --help
usage: stack_mgr check-env [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit

```

```

~$ stack_mgr check-env stack1

-----
Network interfaces
-----
Network interface: eni-016cf2f8ceaba3318 (sbc-eth0)
  IP addresses: 172.31.35.181
  Security groups: sg-0872da39c1b6112d8, sg-038adbdf823381bf8, sg-0a3b1e2d2f039eec9

```

```
<skipped>

*****
SUMMARY
*****

The following problems were detected:
1. Route table rtb-09bd915c613bdc464 lacks default route to
Internet Gateway
```

4.8.4 Checking Connectivity

Use **check-connectivity** command to check the connectivity between Stack Manager and deployed stacks. If connectivity tests fail, you will be provided with common problem resolutions.

```
$ stack_mgr check-connectivity --help
usage: stack_mgr check-connectivity [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
```

```
~$ stack_mgr check-connectivity stack1

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

ERROR: Stack Manager cannot establish connection with the SBC
cluster via HTTPS protocol - request timed out.
This means that some network configuration / security equipment is
blocking connection between Stack Manager and SBC cluster.
If Stack Manager is deployed into a different Vnet / Subnet than
SBC cluster's main interface (eth1) - create proper peering /
routing to enable connectivity. Also verify configuration of
Network Security Groups for SBC cluster's main interface (eth1) -
at both Interface and Subnet level - and make sure that they allow
traffic from Stack Manager via HTTPS protocol (tcp/443). Finally
log into the SBC cluster's Web UI, navigate to IP NETWORK >
SECURITY > Firewall and check that Firewall configuration is not
blocking HTTPS traffic from the Stack Manager to OAM interface
(eth1).
```

4.8.5 Updating Connectivity

If connectivity with the stack fails due to the wrong IP address or credentials, use the following commands to update these parameters and restore the connectivity:

```
$ stack_mgr modify --management-ip <ip-address>
$ stack_mgr modify --username <username>
$ stack_mgr modify --password <password>
```



For Mediant VE and CE stacks, Stack Manager creates a dedicated *StackMgr* user with a randomized password during stack deployment and uses it to communicate with the deployed stack (via REST API). It's recommended to keep using this dedicated user and only update its password if needed

4.9 Scaling Mediant CE Stack

The number of active Media Components in the Mediant CE stack can vary to match the required service capacity. This is called scaling and ensures that the stack utilizes the optimal number of resources at any point of time and elastically scales on demand. Operation that increases the number of active Media Components is called *Scale Out*. Operation that decreases the number of active Media Components is called *Scale In*.

4.9.1 Scale Out Operation

The *Scale Out* operation increases the number of Media Components in the Mediant CE stack by starting additional pre-created "idle" Media Components (for example, corresponding to the AWS EC2 instance state changes from *stopped* to *running*).

You must specify a valid stack name and optionally, specify a number of Media Components to be added to the service. If the number of Media Components is not specified, one Media Component is added.

```
$ stack_mgr scale-out --help
usage: stack_mgr scale-out [-h] [-n num] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -n num, --num       number of media components to be added
  -i ids, --ids ids   comma-separated list of media component id's
                     to be added, e.g. "mc-3,mc-4"
```

The **scale-out** command is not allowed when *Automatic Scaling* is enabled. Use the **scale** command instead.

```
$ stack_mgr scale-out stack1
The following media components will be brought into service: mc-3
Checking that configuration is allowed... done
```

```

Initializing AWS client... done
Starting components 'mc-3'..... done
Successfully started 'mc-3'
Adding media components 'mc-3' to SBC configuration... done
Waiting until media components are ready..... done

```

4.9.2 Scale In Operation

The *Scale In* operation decreases the number of Media Components in the Mediant CE stack by stopping a certain number of "active" Media Components (for example, corresponding to the AWS EC2 instance state changes from *running* to *stopped*).

You must specify the valid stack name and optionally, specify one of the following:

- Number of Media Components to be removed from the service
- Names of specific Media Components to be removed from the service

If none of the above parameters are specified, one Media Component is removed.

If you don't specify Media Component names, Stack Manager automatically removes Media Components with the lowest media utilization:

```

$ stack_mgr scale-in --help
usage: stack_mgr scale-in [-h] [-n num] [-i ids] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -n num, --num num   number of media components to be removed
  -i ids, --ids ids   comma-separated list of media component
                     ids to be removed, e.g. mc-3,mc-4

```

The **scale-in** command is not allowed when *Automatic Scaling* is enabled. Use **scale** command instead.

```

$ stack_mgr scale-in stack1
Choosing media components to be taken out of service..... done
The following media components will be taken out of service: mc-3
Checking that configuration is allowed... done

Initializing AWS client... done
Removing media components 'mc-3' from SBC configuration
Locking media component 'mc-3'.... done
Removing media components 'mc-3'... done
Stopping components 'mc-3'..... done

```

4.9.3 Scale To Operation

Scale To operation sets the number of Media Components in the Mediant CE stack to the specified value. It essentially performs *Scale In* or *Scale Out* operation, depending on the current stack state.

You must specify the valid stack name and a number of active Media Components in the cluster.

```

$ stack_mgr scale --help

```

```
usage: stack_mgr scale [-h] [-n num] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -n num, --num num   number of media components
```

Contrary to *Scale In* and *Scale Out* operations, the *Scale To* operation is allowed when *Automatic Scaling* is enabled. Regardless of whether it adds or removes Media Components, for the purposes of calculating a cool down period, the *Scale To* operation is considered to be equivalent to the *Scale Out* operation. This means that cluster size can be increased immediately after completing the **Scale To** command, if needed.

```
$ stack_mgr scale stack1 -n 3
The following media components will be brought into service: mc-3
Checking that configuration is allowed... done

Initializing AWS client... done
Starting components 'mc-3'..... done
Successfully started 'mc-3'
Adding media components 'mc-3' to SBC configuration... done
Waiting until media components are ready..... done
```

4.10 Modifying Stack Configuration

The **modify** command modifies the configuration of the stack.

```
$ stack_mgr modify --help
usage: stack_mgr modify [-h] [--max-mc-num MAX_MC_NUM]
                        [--min-mc-num MIN_MC_NUM]
                        [--auto-scale {enable,disable}]
                        ...
                        name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  --max-mc-num MAX_MC_NUM
                        maximum number of media components
  --min-mc-num MIN_MC_NUM
                        minimum number of media components
  --auto-scale {enable,disable}
                        auto scaling
  ...
```

The **modify** command is not allowed when some other operation is performed, for example, when the **scale-in** command is in progress.

```
$ stack_mgr modify stack1 --max-mc-num 5
Modifying stack configuration... done
```

The **modify** command has no effect on the stack service and completes without any delay. Some modifications require the **update** command to apply the changes. This is indicated in the **modify** command response:

```
$ stack_mgr modify stack1 --mc-num-of-interfaces 4
Modifying stack configuration... done

Stack configuration was modified.
Use 'update' command to apply the changes.
```

The indication is also provided in the output of the **show** command:

```
$ stack_mgr show stack1

<skipped>

Stack configuration changed           : update is needed
The following parameters were changed : mc_num_of_interfaces
```

For a detailed list of modifiable parameters and their effect on service, see Section 3.14.2.

4.10.1 Update Operation

The **update** command updates stack configuration. It is typically used after the **modify** command when the output of the latter indicates that an update is needed. For example, the **update** command is needed when the number of network interfaces on signaling or Media Components is changed.



The *Update* operation might be service affecting cause. It's therefore recommended to run it during periods of maintenance.

```
$ stack_mgr update --help
positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -f, --force         force update even if it's not needed
```

Usually, the **update** command does nothing unless the 'update is needed' flag was turned on by the **modify** command. This behavior can be overridden by providing the '--force' argument.

```
$ stack_mgr update stack1

Initializing AWS client... done
Checking that configuration is allowed... done
```

```
Updating signaling components... done
Updating media components... done
Updating SBC cluster configuration... done
Wait for new configuration to be applied..... done
```


4.11 Stopping and Starting the Stack

4.11.1 Stopping Stack

The **stop** command stops all stack components (both signaling and media). It's typically used to temporarily shut down stacks in a lab environment.

```
$ stack_mgr stop --help
usage: stack_mgr stop [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -i ids, --ids ids   comma-separated list of component id's,
                     e.g. "mc-1,mc-3"
```

```
$ stack_mgr stop stack1
Initializing AWS client... done
Stopping stack components..... done
```

The **stop** command can also be used to stop specific components by using the **--ids** argument. This option is primarily used for debugging.

4.11.2 Starting Stack

The **start** command starts all stack components. It's typically used after the **stop** command, to restore the stack to its operational state.

```
$ stack_mgr start --help
usage: stack_mgr start [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -i ids, --ids ids   comma-separated list of component ids,
                     e.g. "mc-1,mc-3"
```

```
$ stack_mgr start stack1
Initializing AWS client... done
Starting stack components..... done
```

The **start** command can also be used to start specific components by using the **--ids** argument. This option is primarily used for debugging.

4.12 Rebooting Stack Components

The **reboot** command reboots specific stack components specified by **-ids** argument.

```
$ stack_mgr reboot --help
usage: stack_mgr reboot [-h] [-i ids] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -i ids, --ids ids   comma-separated list of component id's to be
rebooted, e.g. "sc-2,mc-3"
```

```
$ stack_mgr reboot stack1 --ids mc-1
Initializing AWS client... done
Stopping components 'mc-1'..... done
Starting components 'mc-1'..... done
```

4.13 Deleting Stack

The **delete** command deletes the existing stack. You must specify the stack name.

```
$ stack_mgr delete --help
usage: stack_mgr delete [-h] name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
```

The *delete* process takes a few minutes and detailed progress information is displayed on the console:

```
$ stack_mgr delete stack1

Initializing AWS client... done
Deleting signaling components..... done
Deleting media components..... done
Deleting network resources..... done

Stack 'stack1' is successfully deleted
```

4.13.1 Purging Deleted Stack

The deleted stack is displayed by the `list` command (with the state "deleted") for 30 minutes after deletion:

```
$ stack_mgr list

+-----+-----+-----+-----+
| name  | type          | vim  | state  |
+-----+-----+-----+-----+
| stack1 | sbc-cluster  | aws  | deleted |
+-----+-----+-----+-----+
```

If you want to immediately remove the deleted stack from the list, use the `purge` command:

```
$ stack_mgr purge stack1
Stack 'alex1' is purged

$ stack_mgr list
No stacks exist
```

4.14 Healing Stack

The *Heal* operation verifies the state of all stack components and fixes any errors if detected. For example, it can remove Media Components that are not properly registered in the Signaling Components or remove orphaned entries from the "Media Components" configuration table.

```
$ stack_mgr heal stack1

Checking media components status... done
'mc-3' should be removed
Removing media components..... done
```

4.15 Rebuilding Stack

The *Rebuild* operation rebuilds specific stack components. You must specify the stack name and component names to be rebuilt.

```
$ stack_mgr rebuild --help
usage: stack_mgr rebuild [-h] [-i ids] name

positional arguments:
  name                  name of the stack

optional arguments:
  -h, --help            show this help message and exit
  -i ids, --ids ids     comma-separated list of component id's to
                        be rebuilt, e.g. "sc-2,mc-3"
```

The *Rebuild* operation deletes the corresponding virtual machine and creates a new one instead of it. Network interfaces are preserved and therefore, both private and public IP addresses remain unchanged.

If you rebuild Signaling Component instances, you need to generate and apply a new license to them. This is because the instance's serial number changes during the rebuild operation.

```
$ stack_mgr rebuild stack1 --ids mc-2

Initializing AWS client... done
Waiting until signaling components are ready... done
Terminating component 'mc-2'..... done
Rebuilding media component 'mc-2'.....done
Removing media components 'mc-2' from SBC configuration... done
Adding media components 'mc-2' to SBC configuration... done
Checking that all media components have matching SBC
configuration... done
Verifying that all media components are unlocked... done
```

4.16 Managing Files

Stack Manager versions 3.5.0 and later implements Files Repository that allows you to add multiple CMP files. CMP files added to the files repository can be used during the **Upgrade** operation, as described in Section 4.17.1, Hosting Software Load (CMP) Files on Stack Manager.



CMP files added to Stack Manager's Files Repository are publicly accessible through the following URL: <stack-mgr-base-url>/files/<filename>

```
$ stack_mgr file-list --help
usage: stack_mgr file-list [-h]

optional arguments:
  -h, --help  show this help message and exit

$ stack_mgr file-add --help
usage: stack_mgr file-add [-h] source_path name

positional arguments:
  source_path  source path
  name        file name

optional arguments:
  -h, --help  show this help message and exit

$ stack_mgr file-delete --help
usage: stack_mgr file-delete [-h] name

positional arguments:
  name        file name

optional arguments:
  -h, --help  show this help message and exit
```

4.17 Upgrading Stack

The *Upgrade* operation upgrades all stack components. You must specify the stack name and publicly accessible HTTP/HTTPS URL with software load (CMP).

```
$ stack_mgr upgrade --help
usage: stack_mgr upgrade [-h] [-i ids] [--cmp-url URL]
                        [--graceful-timeout TIMEOUT]
                        name

positional arguments:
  name                  name of the stack

optional arguments:
  -h, --help            show this help message and exit
  --cmp-url URL         SBC software load URL
  -i ids, --ids ids    comma-separated list of component id's,
                        e.g. "sc,mc"
  --mode {hitless,non-hitless}
                        upgrade mode
  --graceful-timeout TIMEOUT
                        graceful timeout for media components
                        upgrade (in seconds)
```

The *upgrade* process takes considerable amount of time and detailed progress information is displayed on the console:

```
$ stack_mgr upgrade alex-test-2 --cmp-url
https://sbc2.blob.core.windows.net/pub/test1.cmp

Initializing AWS client... done
Checking that configuration is allowed... done
Checking URL https://sbc2.blob.core.windows.net/pub/test1.cmp...
done
Upgrading signaling components..... done
Version after upgrade: 7.40A.005.509
Upgrading media components..... done
```

4.17.1 Hosting Software Load (CMP) Files on Stack Manager

You can optionally use Stack Manager to host the software load (CMP) file used to upgrade the Mediant VE/CE components.

See Section 3.21.1, Hosting Software Load (CMP) Files on Stack Manager for more information.

3. Copy CMP file to Stack Manager using the SCP/SFTP protocol.
4. Add the copied file to the Files Repository using the **stack_mgr file-add** command.
5. Use the added file by specifying the following value for the **--cmp-url** parameter in the **stack_mgr upgrade** command:

```
<stack-mgr-base-url>/files/<filename>
```

4.17.2 Upgrading Software on Idle Media Components



This section is applicable only to Mediant CE stacks.

When software upgrade of Media Components is performed through the Mediant CE's Web interface (**Setup > IP Network > Cluster Manager Settings > Start Upgrade**), as described in the *Mediant Software User's Manual*, it applies only to "active" Media Components (that are in "started" state).

To complete upgrade for "idle" Media Components (that are in "stopped" state), use the following CLI command:

```
$ stack_mgr update --idle-mcs
```

4.18 Shelving and Unshelving the Stack

4.18.1 Shelving Stack

The **shelve** command reduces Mediant CE stack footprint by deleting Media Components' virtual machines and Load Balancers in an Azure environment. It's typically used to reduce infrastructure (cloud) costs in a lab environment.

```
$ stack_mgr shelve --help
usage: stack_mgr shelve [-h] name

positional arguments:
  name          name of the stack

optional arguments:
  -h, --help  Show this help message and exit
```

```
$ stack_mgr shelve stack1
Initializing Azure client... done
Stopping components 'sc-1, sc-2, mc-1, mc-2'..... done
Checking components status.... done
Deleting components 'lb-public, mc-1, mc-2, mc-3'..... done
Deleting disks of media components..... done
Deleting network interfaces of media components..... done
Deleting public IP addresses of media components..... done
```

4.18.2 Unshelving Stack

The **unshelve** command restores components deleted during the *Shelve* operation and brings stack to fully operational state.

```
$ stack_mgr unshelve --help
usage: stack_mgr unshelve [-h] name

positional arguments:
  name          name of the stack

optional arguments:
  -h, --help  Show this help message and exit
```

```
$ stack_mgr unshelve stack1
Initializing Azure client... done
Checking status of all components... done
Rebuilding load balancers..... done
Starting components 'sc-1, sc-2'..... done
Successfully started 'sc-1, sc-2'
Waiting until signaling components are ready..... done
Rebuilding media components 'mc-1, mc-2, mc-3'..... done
Removing media components 'mc-1, mc-2' from SBC configuration...
done
Adding media components 'mc-1, mc-2' to SBC configuration... done
```



```
Waiting until media components 'mc-1, mc-2' are ready..... done
Adding media components 'mc-3' to SBC configuration... done
Waiting until media components 'mc-3' are ready... done
Removing media components 'mc-3' from SBC configuration... done
Stopping components 'mc-3'....., done
```

4.19 Resetting Stack Password

If you forget the login credentials to the deployed Mediant VE / CE stack, you can use the **reset-password** command to configure new credentials:

```
$ stack_mgr reset-password --help
usage: stack_mgr reset-password [-h] [-u USERNAME] [-p PASSWORD]
name

positional arguments:
  name                name of the stack

optional arguments:
  -h, --help          show this help message and exit
  -u USERNAME, --username USERNAME  user name
  -p PASSWORD, --password PASSWORD  password
```

4.20 Sending INI File

You can update configuration of the deployed Mediant VE / CE stack via the **send-ini** command:

```
$ stack_mgr send-ini --help
usage: stack_mgr send-ini [-h] name ini_file

positional arguments:
  name                name of the stack
  ini_file           ini file

optional arguments:
  -h, --help          show this help message and exit
```

4.21 Multiple Operations

Stack Manager limits every stack to a single operation (create, scale-out, scale-in, or update) at a time. Attempting to run some commands while other commands are in progress, results in the following output:

```
$ stack_mgr scale-out stack1

ERROR: stack 'stack1' is not in 'running' state (current state is
'scaling-in')
```

This limitation does not apply to the **show** and **list** commands, which can be performed in any state.

For different stacks, multiple operations can be performed simultaneously. For example, you can *scale-out* **stack1** while **stack2** is being *deleted*.

5 REST API

5.1 Overview

The REST API is available under the `/api/v1` path.

The following table provides a brief overview of the functionality supported using the REST API. Detailed information for each command is provided in subsequent sections.

Table 5-1: Supported Functionality by REST API

Method	Path	Command
GET	<code>/api/v1/stacks</code>	<code>list stacks</code>
GET	<code>/api/v1/stacks/<stack_name></code>	<code>show stack</code>
POST	<code>/api/v1/stacks/<stack_name></code>	<code>create stack</code>
DELETE	<code>/api/v1/stacks/<stack_name></code>	<code>delete stack</code>
PUT	<code>/api/v1/stacks/<stack_name></code>	<code>modify stack</code>
PURGE	<code>/api/v1/stacks/<stack_name></code>	<code>purge stack</code>
POST	<code>/api/v1/stacks/<stack_name>/heal</code>	<code>heal stack</code>
POST	<code>/api/v1/stacks/<stack_name>/scale-in</code>	<code>scale-in stack</code>
POST	<code>/api/v1/stacks/<stack_name>/scale-out</code>	<code>scale-out stack</code>
POST	<code>/api/v1/stacks/<stack_name>/scale</code>	<code>scale stack</code>
POST	<code>/api/v1/stacks/<stack_name>/update</code>	<code>update stack</code>
GET	<code>/api/v1/config</code>	<code>get global configuration</code>
PUT	<code>/api/v1/config</code>	<code>update global configuration</code>

5.2 Asynchronous Tasks

Most of the POST commands are performed asynchronously. A typical response contains a reference to an asynchronous task URL.

```
POST /api/v1/stack/<name>/scale-out
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

The REST client should poll this URL to get task status and detailed command output.

```
GET /api/v1/tasks/1
```

```
200 OK
Content-Type: application/json
{
  "status": "in_progress",
  "output": "Removing SBC media components... "
}
```

```
GET /api/v1/tasks/1
```

```
200 OK
Content-Type: application/json
{
  "status": "success",
  "output": "Removing SBC media components..... done"
}
```

Valid task status values are:

- **idle** The task didn't start execution yet
- **in_progress** The task is being executed
- **success** The task has successfully completed
- **failed** The task has failed



The 'output' element can contain newline "\n" characters.

5.3 Authorization

REST API availability and authorization scheme is defined by the **REST API Mode** global configuration parameter.

This parameter supports the following values:


- **Enable** – REST API is enabled and uses the authorization scheme similar to the one used by the Web interface:
 - If login through Azure Entra ID is “disabled”, Basic authorization scheme is used. Use the same credentials (username/password) as for accessing the Web interface. See Section 3.1, Accessing the Web Interface for more information.
 - If login through Azure Entra ID is “enabled”, Bearer authorization scheme (with Entra ID token) is used. See the section below for more information.
 - If login through Azure Entra ID is “optional”, both Basic and Bearer authorization schemes can be used
- **Disable** – REST API is disabled.
- **Enable with Basic Auth** – REST API is enabled with Basic authorization scheme.
- **Enable with Azure Auth** – REST API is enabled with Bearer authorization scheme (which must contain Azure token).
- **Enable with any Auth** – REST API is enabled with both Basic and Bearer authorization schemes.

5.3.1 Authorization via Azure Entra ID

If Stack Manager is configured to login via Azure Entra ID, as described in Section 3.6, Login via Azure Entra ID, use the following procedure to perform authentication for REST API endpoints:

- Enable the Web API in the **Azure application** that represents the Stack Manager.
- Create a new **Azure application** that represents the REST client and configure credentials for it.
- Grant the REST client application access to the Web API in the Stack Manager application.
- Use these credentials to acquire the **access token**.
- Pass the **access token** in the Authorization header when accessing the REST API endpoints.

To enable Web API in Azure application that represents Stack Manager:

1. Open the Azure portal at <https://portal.azure.com/>.
2. If you have access to multiple tenants, click the **Directory + subscription** filter  to select the tenant in which you want to register an application.
3. Navigate to the App registrations page.
4. Select the application created in Section 3.6, Login via Azure Entra ID that represents the Stack Manager.
5. Switch to the Expose an API screen.
6. If **Application ID URI** is not configured yet, click **Set**.
7. Enter the following value for the Application ID URI:

```
api://<client-id>
```

Replace `<client-id>` with the Azure application's **Application (client) ID**, that can be found on the Overview page.

To create new Azure application that represents REST client:

1. Navigate to the App registrations page.
2. Click **New registration**.
3. Enter a display **Name** for the new application (e.g., "My REST Client").
4. Leave the **Redirect URI** group empty.
5. Click **Register** to register the new application.
6. In the app's Overview screen, find the **Application (client) ID** and **Directory (tenant) ID** and store them for future use.
7. Switch to the **Certificates & secrets** screen.
8. Click **New client secret**.
9. Enter **Description** for the new secret, choose the expiration time, and then click **Add**.
10. Copy the value of the generated client secret and store it for future use.

To grant REST client application access to Web API on Stack Manager application:

1. Switch to the API Permissions screen.
2. Click **Add a permission** to add a new permission.
3. In the **My APIs** tab, select the Azure application that represents the Stack Manager.
4. Select **Application permissions** and then the proper permission (e.g., **SecAdmin**).
5. Click **Add permissions**.
6. Ask your Azure Directory administrator to grant admin consent to your app permissions.
7. Wait until the Azure Directory administrator completes the task and verify that the status for your application permissions changes to **Granted**.

To acquire access token using REST Client application credentials:

1. Send a request to the Microsoft identity platform's **token** endpoint:

```
// Line breaks are for legibility only
POST
https://login.microsoftonline.com/<tenant_id>/oauth2/v2.0/token
Content-Type: application/x-www-form-urlencoded
grant_type=client_credentials&
client_id=<rest_client_id>&
client_secret=<rest_client_secret>&
scope=api://<stack_mgr_client_id>/default
```

Replace **<rest_client_id>** with your tenant ID; replace **<rest_client_id>** and **<rest_client_secret>** with client ID and secret of REST Client application; replace **<stack_mgr_client_id>** with client ID of Stack Manager application.

2. A successful response will contain the access token:

```
200 OK
Content-Type: application/json
{
  "token_type": "Bearer",
  "expires_in": 3599,
  "ext_expires_in": 3599,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIs..."
}
```

To access Stack Manager's REST API using access token:

1. Include the access token in the Authorization header when accessing Stack Manager's REST API endpoints:

```
GET https://<stack_mgr>/api/v1/status
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIs...
```

2. If you need to send multiple REST API requests, use the session Cookie returned in the response to the first request in the upcoming requests.

5.4 Discovery

Method: GET
Path: /api/v1
Arguments: None
Description: Returns supported API structure

```
GET /api/v1

200 OK
Content-Type: application/json
{
  "items": [
    {
      "description": "list of available stacks",
      "id": "stacks",
      "url": "/api/v1/stacks"
    },
    {
      "description": "global configuration",
      "id": "config",
      "url": "/api/v1/config"
    },
    {
      "description": "application version",
      "id": "version",
      "url": "/api/v1/version"
    }
  ]
}
```

5.5 Managing Users

5.5.1 Listing Users

Method: GET
Path: /api/v1/users
Arguments: none
Description: Lists configured users

```

GET /api/v1/users

200 OK
Content-Type: application/json
{
  "users": [
    {
      "created_at": 1722117527,
      "name": "Admin",
      "password_expiration": 0,
      "password_expiration_text": "never",
      "password_updated_at": 1722117527,
      "status": "active",
      "type": "sec-admin",
      "updated_at": 0
    }
  ]
}

```

5.5.2 Adding User

Method: POST

Path: /api/v1/users

Arguments: none

Content:

<i>name</i>	username
<i>password</i>	password
<i>type</i>	“sec-admin” (default) “admin” “operator” “monitor”
<i>status</i>	“active” (active) “change-password” “locked”
<i>password_expiration</i>	number of days for password expiration or 0 (unlimited); default: 0

Content type: application/json

Description: Adds a new user.

```

POST /api/v1/users
Content-Type: application/json
{
  "name": "Admin",
  "password": "***",
  "type": "sec-admin",
  "status": "active",
  "password_expiration": 0
}

200 OK
Content-Type: application/json
{

```



```
    "description": "user was added"
  }
```

5.5.3 Modifying User

Method:	PUT
Path:	/api/v1/users
Arguments:	none
Content:	
<i>name</i>	username
<i>new_name</i>	new user name; default: unchanged
<i>password</i>	password; default: unchanged
<i>type</i>	"sec-admin" (default) "admin" "operator" "monitor"
<i>status</i>	"active" (default) "change-password" "locked"
<i>password_expiration</i>	number of days for password expiration or 0 (unlimited); default: 0
Content type:	application/json
Description:	Modifies existing user.

```
PUT /api/v1/users
Content-Type: application/json
{
  "name": "Admin",
  "new_name": "secadmin"
}

200 OK
Content-Type: application/json
{
  "description": "user was modified"
}
```

5.5.4 Deleting User

Method:	DELETE
Path:	/api/v1/users
Arguments:	none
Content:	
<i>name</i>	username
Content type:	application/json
Description:	Deletes user.

```
DELETE /api/v1/files
Content-Type: application/json
{
  "name": "Admin"
}

200 OK
Content-Type: application/json
{
  "description": "file was deleted"
}
```

5.6 Global Configuration

Method: GET
Path: /api/v1/config
Arguments: None
Description: Returns Stack Manager's global configuration

```
GET /api/v1/config

200 OK
Content-Type: application/json
{
  "aws_access_key": "ABCDEFGHIJKLMN",
  "aws_prefix": "",
  "aws_secret_key": "123456789012345678901234567890",
  "rest_api_password": "",
  "rest_api_username": "",
  ...
}
```

5.6.1 Updating Global Configuration

Method:	PUT
Path:	/api/v1/config
Arguments:	None
Content Type:	application/json
Content:	Dictionary of parameter value/pairs
Description:	Updates Stack Manager's global configuration

```
PUT /api/v1/config
Content-Type: application/json
{
  "aws_access_key": "ABCDEDEFGHIJKLMN",
  "aws_secret_key": "12345678901234567890"
}

200 OK
Content-Type: application/json
{
  "description": "success"
}
```

5.7 Listing Available Stacks

Method:	GET
Path:	/api/v1/stacks
Arguments:	None
Description:	Returns a list of all available stacks and basic information per stack

```
GET /api/v1/stacks

200 OK
Content-Type: application/json
{
  "stacks": [
    {
      "created_at": "Mar 14, 2021 16:59:15",
      "deleted_at": "",
      "id": "stack1",
      "management_ip": "51.124.138.162",
      "state": "running",
      "type": "sbc-cluster",
      "url": "/api/v1/stacks/alex1",
      "vim": "aws"
    }
  ]
}
```

5.8 Creating New Stack

Method:	POST
Path:	/api/v1/stacks/<stack_name>
Arguments:	none
Content:	configuration parameters as JSON dictionary or file – configuration file as multipart/form-data
Content type:	application/json or multipart/form-data
Description:	Creates new stack. Stack parameters can be provided as JSON dictionary or as a configuration file. Use /api/v1/template API to get a sample JSON dictionary for new stack creation. Alternatively, you can find sample configuration files in /opt/stack_mgr/cfg directory.
Response:	URL of asynchronous task (as described in Section 5.2)

```

POST /api/v1/stacks/stack1
Content-Type: application/json
{
  "stack_type": "sbc-cluster",
  "vim": "aws",
  "mc_num": 3,
  "min_mc_num": 2,
  "max_mc_num": 10,
  "auto_scale": "enable",
  "media_util_scale_in_threshold": 250,
  "media_util_scale_out_threshold": 100,
  "dsp_util_scale_in_threshold": 0,
  "dsp_util_scale_out_threshold": 0,
  "auto_scale_cooldown_time": 900,
  "auto_scale_in_step": 1,
  "auto_scale_out_step": 1,
  "aws_region": "eu-central-1",
  "vpc_id": "vpc-45f3152c",
  "cluster_subnet_id": "subnet-be6e8bc3",
  "oam_subnet_id": "subnet-1536d368",
  "additional1_subnet_id": "",
  "additional2_subnet_id": "",
  "ssh_key_pair": "aws_ssh_frankfurt_1",
  "sc_ha_mode": "enable",
  "sc_num_of_interfaces": 2,
  "sc_public_ips": "main",
  "sc_additional_ips": "",
  "sc_image_id": "ami-d771563c",
  "sc_instance_type": "r5.2xlarge",
  "sc_iam_role": "SBC-HA-3",
  "sc_disk_size": 100,
  "sc_ini_file_contains_admin_user": "true",

```

```

    "sc_ini_file_url": "",
    "mc_num_of_interfaces": 3,
    "mc_public_ips": "main",
    "mc_additional_ips": "",
    "mc_image_id": "ami-d771563c",
    "mc_instance_type": "m5.large",
    "mc_profile": "forwarding",
    "mc_max_pps_limit": "auto",
    "name_prefix": "",
    "manage_via_https": "enable"
  }

```

202 Accepted

Content-Type: application/json

```

{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}

```

POST /api/v1/stacks/stack1

Content-Type: multipart/form-data; boundary=-----WebKitFormBoundary7MA4YWxkTrZu0gW

```

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="stack1.cfg"
Content-Type: application/octet-stream

```

```

<configuration file>
-----WebKitFormBoundary7MA4YWxkTrZu0gW--

```

202 Accepted

Content-Type: application/json

```

{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}

```

5.8.1 Getting Stack Template

Method: GET

Path: /api/v1/template

Arguments:

- type – stack type; supported values:
 - sbc – Mediant VE
 - sbc-cluster – Mediant CE
 - vaic – Voice.AI Connect
 - win-server – Windows Server

- vim – virtual environment; supported values:
 - aws – Amazon Web Services
 - azure – Microsoft Azure
 - google – Google Cloud Platform
 - openstack - OpenStack

Description: Returns sample JSON dictionary that can be used for new stack creation.

```
GET /api/v1/template?type=sbc&vim=azure
Content-Type: application/json
{
  "accelerated_networking": "disable",
  "additional1_subnet_id": "voip1",
  "additional2_subnet_id": "",
  "additional_ips": "",
  "admin_password": "Admin#123456",
  "admin_username": "sbcadmin",
  "deployment_mode": "vm",
  "ha_mode": "disable",
  "ha_subnet_id": "",
  "image_id": "",
  "ini_file_contains_admin_user": "false",
  "ini_file_url": "",
  "ini_params": "",
  "instance_type": "Standard_DS2_v2",
  "location": "WestEurope",
  "main_subnet_id": "oam",
  "main_subnet_traffic": "all",
  "manage_via_https": "enable",
  "media_ports": "6000-65535/udp",
  "name_prefix": "",
  "num_of_interfaces": "2",
  "oam_ports": "22/tcp,80/tcp,443/tcp",
  "os_type": "8",
  "public_ips": "main",
  "signaling_ports": "5060/udp,5060/tcp,5061/tcp",
  "spot_instances": "disable",
  "stack_type": "sbc",
  "storage_account_type": "StandardSSD_LRS",
  "tags": "",
  "vim": "azure",
  "vnet_id": "StackMgrNetwork/StackMgrNetwork"
}
```

5.9 Checking Stack State and Configuration

Method:	GET
Path:	/api/v1/stacks/<stack_name>
Arguments:	<i>?no_status=True</i> Don't include real-time status information (connection status and media/dsp utilization) in the response.
Description:	Returns detailed information of the specific stack. Unless <i>?no-status=True</i> argument is provided, the command queries the active Signaling Component for real-time connection status and media/dsp utilization per Media Component. This might result in a delay in response (up to 30 seconds), for example, if connection with the active Signaling Component is unavailable.

```
GET /api/v1/stacks/stack1

200 OK
Content-Type: application/json
{
  "additional1_subnet_id": "subnet-1536d368",
  "additional2_subnet_id": "",
  "advanced_config": "",
  "alarms": [
    {
      "name": "mc-2-down",
      "raised_at": "Jul 02, 2020 09:47:56",
      "severity": "MINOR",
      "text": "Media component 'mc-2' is 'disconnected'"
    }
  ],
  "auto_heal": "enable",
  "auto_scale": "enable",
  "auto_scale_cooldown_time": 900,
  "auto_scale_in_step": 1,
  "auto_scale_out_step": 1,
  "aws_region": "eu-central-1",
  "cluster_subnet_id": "subnet-be6e8bc3",
  "comments": "",
  "common_network_config": [
    {
      "id": "subnet-be6e8bc3",
      "interface": "eth0",
      "status": "in-use",
      "subnet": "cluster"
    },
    {
      "id": "subnet-1536d368",
      "cron_scheduler": "eth1",
      "status": "in-use",
      "subnet": "main"
    }
  ]
}
```

```
    }
  ],
  "common_tags": 2,
  "connected_mc_num": 2,
  "created_at": "April 09, 2021 08:55:29",
  "deleted_at": "",
  "dsp_util_scale_in_threshold": 0,
  "dsp_util_scale_out_threshold": 0,
  "free_dsp_resources": -1,
  "free_media_resources": 200,
  "id": "stack1",
  "manage_via_https": "enable",
  "management_ip": "18.197.127.204",
  "max_mc_num": 10,
  "mc_additional_ips": "",
  "mc_image_id": "",
  "mc_instance_type": "m5.large",
  "mc_max_pps_limit": "auto",
  "mc_network_config": [
    {
      "id": "subnet-1536d368",
      "interface": "eth2",
      "status": "in-use",
      "subnet": "additional1"
    },
    {
      "id": "",
      "interface": "eth3",
      "status": "",
      "subnet": "additional2"
    }
  ],
  "mc_num": 2,
  "mc_num_of_interfaces": 3,
  "mc_profile": "forwarding",
  "mc_public_ips": "main",
  "media_components": [
    {
      "created_at": "April 09, 2021 08:55:59",
      "dsp_util": -1,
      "id": "mc-1",
      "instance_type": "m5.large",
      "ip": "172.31.67.240",
      "media_util": 0,
      "status": "connected",
      "version": "7.40A.005.314"
    },
    {
      "created_at": "April 09, 2021 08:55:59",
      "dsp_util": -1,
      "id": "mc-2",
```



```
        "instance_type": "m5.large",
        "ip": "172.31.67.15",
        "media_util": 0,
        "status": "connected",
        "version": "7.40A.005.314"
    }
],
"media_util_scale_in_threshold": 250,
"media_util_scale_out_threshold": 100,
"min_mc_num": 2,
"name_prefix": "",
"oam_subnet_id": "subnet-1536d368",
"sc_additional_ips": "",
"sc_ha_mode": "enable",
"sc_iam_role": "SBC-HA-3",
"sc_image_id": "ami-d771563c",
"sc_ini_file_contains_admin_user": "true",
"sc_ini_file_url": "",
"sc_instance_type": "r5.2xlarge",
"sc_network_config": [
    {
        "id": "",
        "interface": "eth2",
        "status": "",
        "subnet": "additional1"
    },
    {
        "id": "",
        "interface": "eth3",
        "status": "",
        "subnet": "additional2"
    }
],
"sc_num_of_interfaces": 2,
"sc_oam_ports": "22/tcp,80/tcp,443/tcp",
"sc_public_ips": "main",
"sc_signaling_ports": "5060/udp,5060/tcp,5061/tcp",
"signaling_components": [
    {
        "created_at": "April 09, 2021 08:57:19",
        "id": "sc-1",
        "instance_type": "r5.2xlarge",
        "ip": "172.31.71.211",
        "status": "running",
        "version": "7.40A.005.314"
    },
    {
        "created_at": "April 09, 2021 08:57:19",
        "id": "sc-2",
        "instance_type": "r5.2xlarge",
        "ip": "",
```

```

        "status": "running",
        "version": "7.40A.005.314"
    }
],
"ssh_key_pair": "aws_ssh_frankfurt_1",
"stack_type": "sbc-cluster",
"started_at": "April 09, 2021 08:46:59",
"state": "running",
"state_task_url": "",
"stopped_at": "",
"update_needed": false,
"update_reason": "",
"vim": "aws",
"vpc_id": "vpc-45f3152c"
}

```

5.9.1 Viewing IP Addresses of Stack Components

Method: GET

Path: /api/v1/stacks/<stack_name>/ips

Description: Returns IP addresses of stack components.

GET /api/v1/stacks/stack1/ips

200 OK

Content-Type: application/json

```

[
  {
    "addresses": [
      {
        "name": "eth1",
        "public_ip": "52.143.78.251"
      },
      {
        "name": "eth2",
        "public_ip": "40.91.126.121"
      }
    ],
    "component": "public-lb"
  },
  {
    "addresses": [
      {
        "name": "eth1",
        "private_ip": "10.23.0.27"
      },
      {
        "name": "eth2",
        "private_ip": "10.23.2.12"
      }
    ]
  }
]

```

```
    }
  ],
  "component": "sc-1"
},
{
  "addresses": [
    {
      "name": "eth1",
      "private_ip": "10.23.0.33"
    },
    {
      "name": "eth2",
      "private_ip": "10.23.2.13"
    }
  ],
  "component": "sc-2"
},
{
  "addresses": [
    {
      "name": "eth1",
      "private_ip": "10.23.0.24",
      "public_ip": "40.91.122.172"
    },
    {
      "name": "eth2",
      "private_ip": "10.23.2.11",
      "public_ip": "52.143.94.91"
    }
  ],
  "component": "mc-1"
},
{
  "addresses": [
    {
      "name": "eth1",
      "private_ip": "10.23.0.25",
      "public_ip": "52.137.89.84"
    },
    {
      "name": "eth2",
      "private_ip": "10.23.2.6",
      "public_ip": "51.143.105.157"
    }
  ],
  "component": "mc-2"
}
]
```

5.9.2 Checking Deployment Environment

Method: GET
Path: /api/v1/stacks/<stack_name>/check-env
Description: Checks deployment environment for Mediant VE and CE stacks deployed in AWS.

```
GET /api/v1/stacks/stack1/check-env
```

```
200 OK
```

```
Content-Type: application/json
```

```
{
  "description": "\n-----\nNetwork interfaces\n-----\nNetwork interface: eni-016cf2f8ceaba3318 (sbc-eth0)\n IP addresses: 172.31.35.181\n Security groups: sg-0872da39c1b6112d8, sg-038adbdf823381bf8, sg-0a3b1e2d2f039eec9\n\n-----\n..."
}
```

5.9.3 Checking Connectivity

Method: POST
Path: /api/v1/stacks/<stack_name>/check-connectivity
Description: Checks connectivity with stack
Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/check-connectivity
```

```
202 Accepted
```

```
Content-Type: application/json
```

```
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.9.4 Updating Connectivity

Method:	POST
Path:	/api/v1/stacks/<stack_name>/update-connectivity
Description:	Updates stack's connectivity parameters
Arguments:	none
Content:	
<i>ip</i>	IP address
<i>username</i>	username
<i>password</i>	password
Content type:	application/json
Response:	URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/update-connectivity
```

```
Content-Type: application/json
```

```
{
  "ip": "10.1.2.30",
  "username": "StackMgr",
  "password": "Password123456!"
}
```

```
202 Accepted
```

```
Content-Type: application/json
```

```
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```



For Stack Manager versions earlier than 3.5.0, this API passed parameters as request arguments.

5.10 Scaling Mediant CE Stack

5.10.1 Scale Out Operation

Method: POST

Path: /api/v1/stacks/<stack_name>/scale-out

Arguments:

- ?num=2* Defines the number of Media Components to add

Description: Scales out the stack

Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/scale-out

202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.10.2 Scale In Operation

Method: POST

Path: /api/v1/stacks/<stack_name>/scale-in

Arguments:

- ?num=2* Defines the number of Media Components to remove
- ?ids=mc-1,mc-2* Comma-separated list of IDs of Media Components to remove

Description: Scales in the stack

Response: URL of asynchronous task (as described in 5.2)

```
POST /api/v1/stacks/stack1/scale-in

202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.10.3 Scale To Operation

Method: POST

Path: /api/v1/stacks/<stack_name>/scale

Arguments:

?num=2 Defines the number of Media Components

Description: Scales the stack to the specified number of Media Components

Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/scale?num=2
```

```
202 Accepted
```

```
Content-Type: application/json
```

```
{  
  "description": "task accepted",  
  "url": "/api/v1/tasks/1"  
}
```

5.11 Modifying Stack Configuration

Method: PUT
Path: /api/v1/stacks/<stack_name>
Arguments: None
Content type: application/json
Content: Dictionary of parameter value/pairs
Description: Modifies the stack configuration

```
PUT /api/v1/stacks/stack1
Content-Type: application/json
{
  "auto_scale ": "enable",
  "media_util_scale_in_threshold": 230
}

200 OK
Content-Type: application/json
{
  "description": "stack configuration was modified"
}
```

Some modify actions require stack updates to be run to apply them. This is indicated using the *update_needed* attribute in the response. The 'update_needed' flag is set on the stack.

```
PUT /api/v1/stacks/stack1
Content-Type: application/json
{
  "max_mc_num": 10
}

200 OK
Content-Type: application/json
{
  "description": "stack configuration was modified; stack must
be updated to apply the changes",
  "update_needed": True,
  "url": "/api/v1/stacks/stack1/update"
}
```


5.11.1 Update Operation

Method: POST

Path: /api/v1/stacks/<stack_name>/update

Arguments:

?force=True Forces update even if it's not needed

?reset=True Resets 'update is needed' flag without performing the update

Description: Updates the stack

Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/update
```

```
202 Accepted
```

```
Content-Type: application/json
```

```
{  
  "description": "task accepted",  
  "url": "/api/v1/tasks/1"  
}
```

5.12 Stopping and Starting Stack

5.12.1 Stopping Stack

Method:	POST
Path:	/api/v1/stacks/<stack_name>/stop
Arguments:	<i>?ids=mc-1,mc-3</i> Comma-separated list of component IDs to be stopped
Description:	Stops stack components
Response:	URL of asynchronous task (as described in 5.2)

```
POST /api/v1/stacks/stack1/stop
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.12.2 Starting Stack

Method:	POST
Path:	/api/v1/stacks/<stack_name>/start
Arguments:	<i>?ids=mc-1,mc-3</i> Comma-separated list of component IDs to be started
Description:	Starts stack components
Response:	URL of asynchronous task (as described in 5.2)

```
POST /api/v1/stacks/stack1/start
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.13 Rebooting Stack Components

Method: POST

Path: /api/v1/stacks/<stack_name>/reboot

Arguments:

?ids=mc-1,mc-3 Comma-separated list of component IDs to be rebooted

Description: Reboots stack components

Response: URL of asynchronous task (as described in 5.2)

```
POST /api/v1/stacks/stack1/reboot
```

```
202 Accepted
```

```
Content-Type: application/json
```

```
{  
  "description": "task accepted",  
  "url": "/api/v1/tasks/1"  
}
```

5.14 Deleting Stack

Method:	DELETE
Path:	/api/v1/stacks/<stack_name>
Arguments:	none
Description:	Deletes stack
Response:	URL of asynchronous task (as described in 5.2)

```
DELETE /api/v1/stacks/stack1

202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.14.1 Purging Deleted Stack

Method:	PURGE
Path:	/api/v1/stacks/<stack_name>
Arguments:	none
Description:	Purges deleted stack

```
PURGE /api/v1/stacks/stack1

200 OK
Content-Type: application/json
{
  "description": "stack was purged"
}
```

5.15 Healing Stack

Method: POST
Path: /api/v1/stacks/<stack_name>/heal
Arguments: none
Description: Heals the stack
Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/heal
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.16 Rebuilding Stack

Method: POST
Path: /api/v1/stacks/<stack_name>/rebuild
Arguments:
?ids=mc-1,mc-3 Comma-separated list of component IDs to be rebuilt
Description: Rebuilds stack components
Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/rebuild?ids=mc-1
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.17 Managing Files

5.17.1 Listing Files

Method: GET
Path: /api/v1/files
Arguments: none
Description: Lists files in Files Repository

```
GET /api/v1/files

200 OK
Content-Type: application/json
{
  "files": [
    {
      "added_by": "Admin",
      "added_on": "Jul 11, 2024 09:08:45",
      "name": "HostedTP_CENTOS8_SIP_F7.40A.501.329.cmp",
      "status": "ok",
      "type": "cmp"
    }
  ]
}
```

5.17.2 Adding File

Method: POST
Path: /api/v1/files
Arguments: none
Content: file – configuration file as `multipart/form-data`
Content type: multipart/form-data
Description: Adds a new file to Files Repository.

```
POST /api/v1/files
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary7MA4YWxkTrZu0gW

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="test.cmp"
Content-Type: application/octet-stream

<CMP file>
-----WebKitFormBoundary7MA4YWxkTrZu0gW--

200 OK
```

```
Content-Type: application/json
{
  "description": "file was added"
}
```

5.17.3 Deleting File

Method: DELETE
Path: /api/v1/files
Arguments: none
Content:
 *filename*file name
Content type: application/json
Description: Deletes file from Files Repository.

```
DELETE /api/v1/files
Content-Type: application/json
{
  "filename": "test.cmp"
}

200 OK
Content-Type: application/json
{
  "description": "file was deleted"
}
```

5.18 Upgrading Stack

Method: POST

Path: /api/v1/stacks/<stack_name>/upgrade

Arguments:

?ids=sc.mc Comma-separated list of component types to be rebuilt (sc/mc)
&cmp_url=<URL> Publicly accessible HTTP/HTTPS URL with software load (CMP)
&graceful_timeout=60 Graceful timeout for media components upgrade (in seconds)
&mode=(hitless|reset) Mode for Signaling Components upgrade

Description: Upgrades stack components

Response: URL of asynchronous task (as described in Section 5.2)

```
POST
/api/v1/stacks/stack1/upgrade?ids=sc,mc&cmp_url=<URL>&graceful_timeout=60

202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```


5.18.1 Hosting Software Load (CMP) Files on Stack Manager

You can optionally use Stack Manager to host the software load (CMP) file used to upgrade the Mediant VE/CE components.

See Section 3.21.1, Hosting Software Load (CMP) Files on Stack Manager for more information.

1. Copy the CMP file to Stack Manager using the SCP/SFTP protocol.
2. Add the copied file to Files Repository using the **stack_mgr file-add** command.
3. Use the added file by specifying the following value for the **--cmp-url** parameter in the **stack_mgr upgrade** command:

```
http://<stack-mgr>/files/<filename>
```

```
$ stack_mgr file-list --help
usage: stack_mgr file-list [-h]

optional arguments:
  -h, --help  show this help message and exit

$ stack_mgr file-add --help
usage: stack_mgr file-add [-h] source_path name

positional arguments:
  source_path  source path
  name         file name

optional arguments:
  -h, --help  show this help message and exit

$ stack_mgr file-delete --help
usage: stack_mgr file-delete [-h] name

positional arguments:
  name         file name

optional arguments:
  -h, --help  show this help message and exit
```

5.18.2 Upgrading Software on Idle Media Components



This section is applicable only to Mediant CE stacks.

When the software upgrade of Media Components is done through the Mediant CE's Web interface (**Setup > IP Network > Cluster Manager Settings > Start Upgrade**), as described in the *Mediant Software User's Manual*, it applies only to the "active" Media Components (that are in "started" state).

To complete the upgrade for "idle" Media Components (that are in "stopped" state), use the following CLI command:

```
$ stack_mgr update --idle-mcs
```

5.19 Shelving and Unshelving Stack

5.19.1 Shelving Stack

Method: POST
Path: /api/v1/stacks/<stack_name>/shelve
Description: Reduces footprint of Mediant CE stack
Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/shelve
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.19.2 Unshelving Stack

Method: POST
Path: /api/v1/stacks/<stack_name>/unshelve
Description: Restores "shelved" stack to fully operational state
Response: URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/unshelve
```

```
202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

5.20 Resetting Stack Password

Method:	POST
Path:	/api/v1/stacks/<stack_name>/reset-password
Arguments:	none
Content:	
<i>username</i>	username
<i>password</i>	password
Content type:	application/json
Description:	Configures new credentials that can be used to log in to the Mediant VE/CE stack's Web interface
Response:	URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/reset-password
```

```
Content-Type: application/json
```

```
{
  "username": "sbcadmin",
  "password": "Password123456!"
}
```

```
202 Accepted
```

```
Content-Type: application/json
```

```
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```



For Stack Manager versions earlier than 3.5.0, this API passed parameters as request arguments.

5.21 Sending INI File

Method:	POST
Path:	/api/v1/stacks/<stack_name>/send-ini
Arguments:	none
Content:	file – configuration file as multipart/form-data
Content type:	multipart/form-data
Description:	Send incremental INI file to Mediant VE / CE stack
Response:	URL of asynchronous task (as described in Section 5.2)

```
POST /api/v1/stacks/stack1/send-ini
Content-Type: multipart/form-data; boundary=-----WebKitFormBoundary7MA4YWxkTrZu0gW

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="ini.txt"
Content-Type: application/octet-stream

<INI file>
-----WebKitFormBoundary7MA4YWxkTrZu0gW--

202 Accepted
Content-Type: application/json
{
  "description": "task accepted",
  "url": "/api/v1/tasks/1"
}
```

6 Operational Logs

Stack Manager stores its logs in the `/var/log/stack_mgr` directory. The following files are created:

- **stack_mgr.log:** Main application log file.
- **http.log:** Log of operations performed through Web interface and/or REST API.
- **http_access.log:** Log of HTTP/HTTPS requests processed by the Web interface and/or REST API.
- **auto_job.log:** Log of automatic scaling and healing jobs.
- **api.log:** Log of internal API server used to run jobs performed through Web interface and/or REST API.
- **upgrade.log:** Log of upgrades performed through Web interface.

Log files are rotated daily. Up to seven copies of each file are stored.

In addition to above logs, Stack Manager maintains Activity Log that records summary of all operations and configuration changes.

To view logs through the Web interface, open the Logs page, and then choose the corresponding log.

Figure 6-1: Viewing logs in Web Interface

The screenshot shows the Stack Manager Web Interface. The top navigation bar includes 'stack_mgr', 'Stacks', 'Configuration', 'Logs', and 'About'. The 'Logs' page is selected, and the user 'Alex Agranov' is logged in. Below the navigation bar, there are tabs for 'Activity', 'Application', 'Auto job', 'HTTP server', 'HTTP access', and 'API server'. A 'Refresh' button is on the left, and a 'Lines' input field is set to 100. The main content area displays a list of log entries with timestamps, severity levels, and messages.

```
[2022-01-16 06:36:35,154679] [MAJOR] Raise 'rest-api' alarm on stack 'orenu-nqm' - Cannot connect to SBC via REST API
[2022-01-16 06:37:40,578501] [CLEARED] Clear 'rest-api' alarm on stack 'orenu-nqm' - Successfully connected to SBC via REST API
[2022-01-16 06:51:24,914897] [MAJOR] Raise 'rest-api' alarm on stack 'orenu-nqm' - Cannot connect to SBC via REST API
[2022-01-16 06:51:57,999404] [CLEARED] Clear 'rest-api' alarm on stack 'orenu-nqm' - Successfully connected to SBC via REST API
[2022-01-17 17:33:28,706302] [INFO] Start stack 'garyd-aws-ve-1' - Alex Agranov, https, 172.18.110.11
[2022-01-17 17:35:16,949122] [INFO] Heal stack 'garyd-aws-ve-1' - auto-job, internal
[2022-01-17 17:35:30,639569] [DONE] Heal stack 'garyd-aws-ve-1' - done
[2022-01-17 17:39:51,538544] [INFO] Heal stack 'garyd-aws-ve-1' - Alex Agranov, https, 172.18.110.11
[2022-01-17 17:40:03,264757] [DONE] Heal stack 'garyd-aws-ve-1' - done
[2022-01-17 17:40:19,843708] [INFO] Stop stack 'garyd-aws-ve-1' - Alex Agranov, https, 172.18.110.11
[2022-01-17 17:42:06,548630] [DONE] Stop stack 'garyd-aws-ve-1' - done
[2022-01-19 12:04:55,710839] [INFO] Create stack 'dmitryh-ce-test-1' - type: Mediant CE, min_mc_num: 2, max_mc_num: 5 - Dmitry Halpern, https, 172.18.110.214
[2022-01-19 12:16:12,109850] [INFO] Create stack 'ariel-test-sbc' - type: Mediant VE - Ariel Mannes, https, 172.18.110.18
[2022-01-19 12:17:13,767672] [FAILED] Create stack 'ariel-test-sbc' - failed
[2022-01-19 12:17:54,898331] [INFO] Delete stack 'ariel-test-sbc' - Ariel Mannes, https, 172.18.110.18
[2022-01-19 12:18:37,281247] [DONE] Delete stack 'ariel-test-sbc' - done
[2022-01-19 12:20:25,795081] [DONE] Create stack 'dmitryh-ce-test-1' - done
[2022-01-19 12:21:00,240188] [INFO] Create stack 'ariel-test-sbc' - type: Mediant VE - Ariel Mannes, https, 172.18.110.18
[2022-01-19 12:25:40,557028] [DONE] Create stack 'ariel-test-sbc' - done
```

To view logs through CLI, use the following command:

```
$ stack_mgr log --help
usage: stack_mgr log [-h] [--name {activity_log,stack_mgr,
auto_job,http,http_access}] [--lines LINES]

optional arguments:
  -h, --help                show this help message and exit
  --name {activity_log,stack_mgr,auto_job,http,http_access,api}
                             log name
  --lines LINES              number of lines
```

```
$ stack_mgr log --name activity_log --lines 10
[2020-12-14 17:58:03] [INFO] Delete stack 'test-ce-2'
[2020-12-14 18:00:17] [DONE] Delete stack 'test-ce-2' - done
[2020-12-14 18:03:12] [INFO] Delete stack 'test-ce-3'
[2020-12-14 18:03:33] [DONE] Delete stack 'test-ce-3' - done
[2020-12-15 12:54:54] [INFO] Start stack 'test-ve-1'
[2020-12-15 12:55:19] [DONE] Start stack 'test-ve-1' - done
[2020-12-15 12:55:25] [INFO] Modify stack 'test-ve-1'
configuration - auto_heal: enable
[2020-12-15 12:55:37] [INFO] Start stack 'test-ce-1'
[2020-12-15 12:56:56] [DONE] Start stack 'test-ce-1' - done
[2020-12-15 12:57:36] [INFO] Modify stack 'test-ce-1'
configuration - auto_scale: enable
```

6.1 Web Server Logs

Stack Manager uses an NGINX Web server, which stores its logs in the `/var/log/nginx` directory. The following files are created:

- **access.log:** Log of all client requests.
- **error.log:** Log of encountered issues and errors.

7 Stacks Management

7.1 Automatic Stop / Start / Shelve

Stack Manager can be configured to automatically perform *Stop / Start / Shelve* operations at pre-defined time / day. This can be useful for lab environments where stacks need to be running only during specific work hours and automatically stopping / shelving stacks at night helps to save the costs.

Automatic stop / start / shelve behavior is controlled via the following parameters under *Advanced* section in the Stack Manager configuration screen:

- **Auto Stop Time**
- **Auto Start Time**
- **Auto Shelve Time**

Use the following syntax to configure the parameters:

- 08:00 – specific time (24h)
- 1/08:00 - weekday and time (weekday: 0=SUN, 1=MON, ... 6=SAT)
- 0,1,2/08:00 - multiple weekdays and time
- 0-5/08:00 - range of weekdays and time
- 0,1/08:00|2-4/09:00 - multiple statements

It's also possible to specify different auto stop / start / shelve time for specific stack by using **auto_stop_time / auto_start_time / auto_shelve_time** advanced config parameters at stack level. Use the same syntax as above; use “disabled” value to disable corresponding operation.

7.2 Tagging Stack Resources

You can configure Stack Manager to add global tag to all created stack's resources. This, for example, can be useful if you have multiple Stack Manager instances in your account and want to indicate the instance that was used to create specific stack / resource.

To enable tagging for all created stack's resources, navigate to **Configuration** screen and configure **Stack Manager Tag** parameter. You must specify tag name and value as follows: `<tag_name>=<tag_value>. %IP%` element in `<tag_value>` is expanded to Stack Manager's IP address.

For example:

```
stack_mgr_ip=%IP%
```

In addition to the global tag, you can configure per-stack tags using the corresponding Advanced Config parameters (e.g., "sbc_tags" or "sc_tags"). The name and format of these configuration parameters depends on the stack type and deployment environment. See Section 3.8.11, Advanced Configuration for more information.

7.3 Integration with Azure Application Insights

Stack Manager deployed in a Microsoft Azure environment can be configured to publish active alarms and basic stack metrics to Azure Application Insights.

Active alarms are published every 10 minutes as `customEvents` in the following structure:

- name: "sm_alarm"
- customDimensions:
 - name: Alarm name (as reported to OVOC)
 - description: Alarm description
 - entity: Stack name
 - source: Name of stack element to which alarm applies
 - severity: Alarm severity
 - time: Time when alarm was raised
 - id: Unique alarm ID
 - stack_mgr_ip: IP address of the Stack Manager

For example:

```
"name": "sm_alarm",
"customDimensions": {
  "name": "acSmDown",
  "description": "Signaling component 'sc-2' is 'down'",
  "entity": "byoc-ce-1",
  "source": "sc-2",
  "severity": "Minor",
  "time": "2022-01-26T06:57:48.0000000Z",
  "id": "1642",
  "stack_mgr_ip": "10.4.2.4"
```



```
}
```

Stack metrics are published every minute as `customMetrics` in the following structure:

- name: "sm_<metric>"
- customDimensions:
 - entity: Stack name
 - stack_mgr_ip: IP address of the Stack Manager

For example:

```
"name": "sm_CEMediaUsage",  
"valueSum": 0,  
"customDimensions": {  
  "entity": "byoc-ce-1",  
  "stack_mgr_ip": "10.4.2.4"  
}
```

To enable integration with Azure Application Insights, navigate to the **Configuration** screen and then configure the following parameters:

- **Application Insights Connection String:** Specify the connection string, as shown in the Application Insights' Overview screen.
- **Application Insights Mode:** Choose whether you want to generate reports for all stack or for specific stacks only. For the latter case, you must configure the `app_insights` advanced configuration parameter for specific stacks to one of the following values:
 - **enable:** Sends both alarms and metrics
 - **alarms:** Sends alarms only
 - **metrics:** Sends metrics only

8 Secure Deployment

Stack Manager is a user-space application that is deployed on a standard Linux OS distribution.

To ensure secure deployment in a production environment, the following steps are recommended:

- Change the default Web user credentials, as described in Section 2.5, Accessing the Web Interface.
- For each person who will be working with Stack Manager, create a dedicated user with the appropriate access level. See Section 3.3, Managing Users for more information.
- Secure the connection to the Web interface, as described in Section 3.5, Securing Connection to Web Interface.
- Configure cloud network security groups / firewall to limit access to the Web and SSH interfaces to the specific IP addresses or range of IP addresses.
- Regularly apply security updates to your Linux OS distribution.

Most Linux OS distribution are configured to perform automatic unattended updates. If your image is not, consider enabling them via the following commands:

- **Debian / Ubuntu:** `sudo apt update; sudo apt install unattended-upgrades`
- **RHEL / CentOS / Rocky Linux / Alma Linux / Amazon Linux:** `sudo yum install dnf-automatic; sudo systemctl enable --now dnf-automatic-install.timer`

You can also apply updates manually via the following commands:

- **Debian / Ubuntu:** `sudo apt update; sudo apt upgrade`
- **RHEL / CentOS / Rocky Linux / Alma Linux / Amazon Linux:** `sudo yum upgrade`
- (Optional) Update the default Web server (nginx) configuration.

The configuration file is located at:

- **Debian / Ubuntu:** `/etc/nginx/sites-available/stack_mgr`
- **RHEL / CentOS / Rocky Linux / Alma Linux / Amazon Linux:** `/etc/nginx/nginx.conf`

The following configuration parameters might need to be adjusted to match your company's security policies:

- `ssl_protocols`: Specifies the supported SSL/TLS protocol versions
- `ssl_ciphers`: Specifies the supported ciphers

Starting from version 3.5.0, Stack Manager automatically adds the following security headers to its responses:

`Content-Security-Policy, Strict-Transport-Security, Referrer-Policy, X-Frame-Options, X-XSS-Protection, X-Content-Type-Options`

If you're using earlier versions, consider adding these headers using the `add_header nginx` configuration parameter.

Stack Manager configures nginx to exclude its version in the HTTP responses. However, the Server header is still present (with "nginx" value). If you're using **Debian / Ubuntu** you can completely remove the Server header from HTTP responses, as follows:

- a. Install the "http-headers" nginx module using the following command:


```
sudo apt install libnginx-mod-http-headers-more-filter
```
- b. Add the following after the "`server_tokens off`" line in the nginx configuration file:

```
more_clear_headers Server;
```

Refer to Nginx documentation at <http://nginx.org/en/docs> for more information.

Restart the Web server via the following command to apply the changes: `systemctl restart nginx`

- (Optional) Update the default SSH server (sshd) configuration.

The configuration file is located at `/etc/ssh/sshd_config`

The following configuration parameters might need to be adjusted to match your company's security policies:

- `Ciphers`: Specifies symmetric encryption algorithms.
- `HostKeyAlgorithms`: Specifies host key algorithms.
- `KexAlgorithms`: Specifies key exchange algorithms.

Refer to Sshd documentation https://www.ssh.com/academy/ssh/sshd_config for more information.

Restart the SSH server via the following command to apply the changes: `systemctl restart sshd`

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-28967

