AudioCodes Mediant<sup>™</sup> Family of Session Border Controllers

# **Stack Manager**

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

Version 7.4



**C**C audiocodes

## **Table of Contents**

| 1 | Intro | oducti  | on  | 13       |
|---|-------|---------|---|----------|
| 2 | Dep   | loyme   | nt  | 15       |
|   | 2.1   | Opera   | ational Environment   |          |
|   | 2.2   | Netwo   | ark Topology  | 15       |
|   | 2.2   | Install | lation Prerequisites  | 16       |
|   | 2.0   | 2 3 1   | Installation Prerequisites for Amazon Web Services (AWS) Environment              | 10       |
|   |       | 2.0.1   | 2.3.1.1 IAM Role for Stack Manager  |          |
|   |       |         | 2.3.1.2 Subnet and Elastic IP Addresses   | 19       |
|   |       | 2.3.2   | Installation Prerequisites for Microsoft Azure Environment                        | 20       |
|   |       | 0.0.0   | 2.3.2.1 Subnet and Public IP Addresses  | 20       |
|   |       | 2.3.3   | Installation Prerequisites for Google Cloud Environment                           | 21<br>21 |
|   |       | 2.3.4   | Installation Prerequisites for OpenStack Environment                              |          |
|   |       |         | 2.3.4.1 Provider Versus Self-Service Networks                                     | 22       |
|   |       |         | 2.3.4.2 Subnet and Floating IP Addresses  | 22       |
|   | 2.4   | Install | ation   | 23       |
|   |       | 2.4.1   | Overview  | 23       |
|   |       | 2.4.2   | Creating Amazon Web Services (AWS) Instance                                       | 24       |
|   |       | 2.4.3   | Deploying Stack Manager on Microsoft Azure  | 27       |
|   |       | 2.4.4   | Creating OpenStack Instance   | 32<br>34 |
|   |       | 2.4.6   | Installing Stack Manager Application  |          |
|   | 2.5   | Acces   | ssing the Web Interface   |          |
|   | 2.6   | Acces   | ssing the CLI   | 39       |
|   | 27    | Unora   | ading Stack Manager   | 39       |
|   | 2.8   | Post-i  | nstallation Configuration   | 00<br>   |
|   | 2.0   | 281     | Post-installation Configuration on Amazon Web Services (AWS)                      |          |
|   |       | 2.0.1   | 2.8.1.1 Enabling Access to AWS API via IAM Role (Recommended Method               | 1)41     |
|   |       |         | 2.8.1.2 Enabling Access to AWS API via AWS Access Key (Alternative Met<br>41      | thod)    |
|   |       | 2.8.2   | Post-Installation Configuration on Microsoft Azure                                | 42       |
|   |       |         | 2.8.2.1 Configuring the Azure Subscription ID                                     | 42       |
|   |       |         | (Recommended Method)  | 43       |
|   |       |         | 2.8.2.3 Enabling Access to Azure APIs via Service Principal (Alternative Me<br>50 | ethod)   |
|   |       | 2.8.3   | Post-Installation Configuration on Google Cloud                                   | 51       |
|   |       |         | 2.8.3.1 Configuring Google Project ID   | 51       |
|   |       |         | 2.8.3.2 Enabling APIs in Project  | 51       |
|   |       |         | 2.8.3.4 Enabling Access to Google Cloud APIs via Service Account                  |          |
|   |       |         | (Recommended Method)  | 53       |
|   |       |         | 2.8.3.5 Enabling Access to Google Cloud APIs via Configuration File (Altern       | native   |
|   |       |         | Method) 53  |          |
|   |       | 2.8.4   | Post-Installation Configuration on OpenStack                                      | 54       |
|   | 20    | 2.0.J   | veniying coniguration   | 55       |
|   | 2.9   |         | Storing Puntime Data on AWS S2  | 50       |
|   |       | 2.9.1   | Storing Runtime Data on Avvo So   | 50<br>58 |
|   |       | 2.9.3   | Storing Runtime Data on Google Cloud Storage Service                              | 58       |
|   |       | 2.9.4   | Storing Runtime Data on OpenStack Object Storage Service                          | 59       |
|   |       | 2.9.5   | Migrating Runtime Data from Local Disk to Storage Service                         | 59       |

|   | 2.10 | Resourc   | e Naming   | 59             |
|---|------|-----------|--|----------------|
|   | 2.11 | Backup a  | and Restore  | 60             |
|   | 2.12 | Migrating | g to a New Virtual Machine                                       | 60             |
|   | 2.13 | Providing | z<br>Debua File for Troubleshootina                              | 61             |
| 3 | Web  | Interfac  | ;e   | 63             |
|   | 31   | Accessir  | ng the Web Interface   | 63             |
|   | 3.2  |           |  | 64             |
|   | 0.2  | Monogin   |  | 0 <del>4</del> |
|   | 3.3  |           | g USels  | 05             |
|   |      | 3.3.1 U   | assword Complexity   | 07<br>67       |
|   |      | 3.3.3 P   | assword Reuse  | 68             |
|   | 3.4  | Global C  | configuration  | 69             |
|   | 0    | 3.4.1 G   | General Configuration Parameters                                 |                |
|   |      | 3.4.2 C   | hange Password Block   | 70             |
|   |      | 3.4.3 S   | ecurity Configuration Parameters                                 | 70             |
|   |      | 3.4.4 N   | licrosoft Azure Parameters                                       | 72             |
|   |      | 3.4.5 A   | mazon AWS Parameters   | 73             |
|   |      | 3.4.0 G   | oogle Cloud Parameters   | 14<br>74       |
|   |      | 3.4.8 D   | ebug File Parameters   | 74             |
|   |      | 3.4.9 A   | dvanced Parameters   | 75             |
|   | 3.5  | Securing  | Connection to Web Interface                                      | 76             |
|   |      | 3.5.1 C   | onfiguring Hostname for Stack Manager Virtual Machine            | 77             |
|   |      | 3.5.2 A   | cquiring Certificate from Certificate Authority                  | 78             |
|   |      | 3.5.3 Ir  | stalling Let's Encrypt Certificates                              | 79             |
|   |      | 3.5.4 E   | nforcing Secure Connection                                       | 79             |
|   | 3.6  | Login via | a Azure Entra ID   | 80             |
|   | 3.7  | Resetting | g Web Interface Credentials                                      | 83             |
|   | 3.8  | Creating  | a New Stack  | 84             |
|   |      | 3.8.1 C   | reating Mediant CE in Amazon Web Services (AWS) Environment      | 85             |
|   |      | 3         | .8.1.1 Troubleshooting   | 88             |
|   |      | 3.8.2 C   | Preating Mediant CE in Azure Environment                         | 89             |
|   |      | 383 0     | .8.2.1 Troubleshooling   | 92             |
|   |      | 3.8.4 C   | reating Mediant CE in OpenStack Environment                      | 90             |
|   |      | 3.8.5 C   | reating Mediant VE in Amazon Web Services (AWS) Environment      | 98             |
|   |      | 3         | .8.5.1 Troubleshooting   | 100            |
|   |      | 3.8.6 C   | reating Mediant VE in Azure Environment                          | 101            |
|   |      | 3         | .8.6.1 I roubleshooting  | .103           |
|   |      | 3.8.7 U   | reating VoiceAI Connect in Amazon Web Services (AWS) Environment | 104            |
|   |      | 3.8.9 C   | reating VoiceAl Connect in Azure Environment.                    | 109            |
|   |      | 3.8.10 C  | reating VoiceAl Connect in Google Cloud Environment              | 112            |
|   |      | 3.8.11 A  | dvanced Configuration  | 114            |
|   |      | 3         | .8.11.1 Advanced Configuration for Mediant CE                    | 114            |
|   |      | 3         | .8.11.2 Advanced Configuration for Mediant VE                    | .135           |
|   | 2.0  | Chooking  | a Stock State and Configuration                                  | . 140          |
|   | 5.9  |           | jouing ID Addresses of Stack Components                          | 100            |
|   |      | 392 C     | iewing in Addresses of Stack Components                          | 160            |
|   |      | 3.9.3 C   | hecking Connectivity   |                |
|   |      | 3.9.4 U   | pdating Connectivity   | 161            |
|   | 3.10 | Active A  | arms   | .161           |
|   | 3.11 | Performi  | ng Operations on Stack   | .162           |
|   |      |           |  |                |

|   | 3.12    | Scaling Mediant CE Stack   | 163  |
|---|---------|--|------|
|   |         | 3.12.1 Scale Out Operation   | 163  |
|   |         | 3.12.2 Scale In Operation  | 164  |
|   |         | 3.12.3 Scale To Operation  | 164  |
|   | 3.13    | Automatic Scaling  | 165  |
|   |         | 3.13.1 Cool Down Period  | 166  |
|   |         | 3.13.2 Auto Scale Step   | 166  |
|   | 2 1 /   | Modifying Stock Configuration  | 167  |
|   | 5.14    | 2 14 1 Undete Operation  | 160  |
|   |         | 3.14.1 Opuale Operation  | 170  |
|   |         | 3.14.3 Modifiable Parameters for Mediant VE                                | 173  |
|   | 3.15    | Stopping and Starting Stack  | 174  |
|   | 3.16    | Rebooting Stack Components   | 174  |
|   | 3.17    | Healing Stack  | 175  |
|   |         | 3.17.1 Automatic Healing   | 175  |
|   | 3 18    | Deleting Stack   | 176  |
|   | 3 19    | Rebuilding Stack   | 176  |
|   | 3 20    | Managing Files   | 177  |
|   | 3.21    | Ungrading Stack  | 177  |
|   | 5.21    | 3 21 1 Hosting Software Load (CMP) Files on Stack Manager                  | 170  |
|   |         | 3.21.2 Upgrading Software on Idle Media Components.                        | 180  |
|   | 3 22    | Shelving and Unshelving Stack  | 180  |
|   | 3.23    | Resetting Stack Password   | 181  |
|   | 3 24    | Sending INI File   | 181  |
|   | 3 25    | Stack Deployment Details   | 182  |
|   | 0.20    | 3 25 1 Deployment Method   | 182  |
|   |         | 3.25.2 Adjusting Security Groups   | 183  |
|   |         | 3.25.2.1 Modifying Security Groups Created by Stack Manager in Azure       |      |
|   |         | Environment  | 183  |
|   |         | 3.25.3 Using Pre-Defined Public IP Addresses                               | 184  |
|   |         | 3.25.4 Using Pre-Defined Virtual IP Addresses                              | 100  |
|   |         | 3.25.6 Using Pre-Defined IPv6 Addresses                                    | 191  |
|   | <u></u> |  | 400  |
| 4 | CLI     | Interface  | .193 |
|   | 4.1     | Accessing CLI Interface  | 193  |
|   | 4.2     | Invocation   | 193  |
|   | 4.3     | Usage Information  | 193  |
|   | 4.4     | Managing Users   | 194  |
|   | 4.5     | Global Configuration   | 196  |
|   | 4.6     | Listing Available Stacks   | 196  |
|   | 4.7     | Creating a New Stack   | 197  |
|   |         | 4.7.1 Creating Stack Configuration File via SBC Cluster Configuration Tool |      |
|   |         | (Recommended Method)   | 197  |
|   |         | 4.7.2 Creating Stack Configuration File Manually (Alternative Method)      | 203  |
|   |         | 4.7.2.1 Sample Configuration File  | 204  |
|   | 10      | 4.1.3 Creating a new Stack   | 211  |
|   | 4.Ŏ     | A 9.1 Checking Idle Media Components                                       |      |
|   |         | 4.0.1 Checking late media Components                                       | 215  |
|   |         |  |      |

|   |   | 4.8.3 Chee<br>4.8.4 Chee<br>4.8.5 Under  | cking Deployment Environment<br>cking Connectivity            | .216<br>.217   |
|---|---|--|---|--|
|   | 10  | Scaling Me   | diant CE Stack  | 218  |
|   | 7.5   | 4 9 1 Scal   |   | 218  |
|   |   | 4.9.2 Scal   | e In Operation  | .219   |
|   |   | 4.9.3 Scal   | e To Operation  | .219   |
|   | 4.10  | Modifying S  | Stack Configuration   | 220  |
|   |   | 4.10.1 Upda  | ate Operation   | .221   |
|   | 4.11  | Stopping an  | nd Starting the Stack   | 223  |
|   |   | 4.11.1 Stop<br>4.11.2 Star   | pping Stack<br>ting Stack                                     | .223<br>223  |
|   | 4 12  | Rebooting  | Stack Components  | 224  |
|   | 4 13  | Deleting St  | ack   | 224  |
|   |   | 4.13.1 Puro  | uing Deleted Stack  | 225  |
|   | 4.14  | Healing Sta  | ack   | 225  |
|   | 4 15  | Rebuilding   | Stack   | 226  |
|   | 4 16  | Managing F   | Files   | 227  |
|   | 4 17  | Upgrading  | Stack   | 228  |
|   | 4.17  | 4 17 1 Host  | ting Software Load (CMP) Files on Stack Manager               | 229  |
|   |   | 4.17.2 Upg   | rading Software on Idle Media Components                      | .229   |
|   | 4.18  | Shelving ar  | nd Unshelving the Stack                                       | 230  |
|   |   | 4.18.1 Shel  | lving Stack   | .230   |
|   |   | 4.18.2 Unsl  | helving Stack   | .230   |
|   | 4.19  | Resetting S  | Stack Password  | 231  |
|   | 4.20  | Sending IN   | I File  | 231  |
|   |   |  |   |  |
|   | 4.21  | Multiple Op  | erations  | 232  |
| 5 | 4.21<br><b>RES</b>  | Multiple Op<br>T API   | perations   | 232<br>233   |
| 5 | 4.21<br><b>RES</b><br>5.1   | Multiple Op<br><b>T API</b><br>Overview  | perations   | 232<br>233<br>233  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2  | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono  | perations   | 232<br>233<br>233<br>234   |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3   | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio  | perations   | <ul> <li>232</li> <li>233</li> <li>233</li> <li>234</li> <li>235</li> </ul>  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3   | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth  | perations   | 232<br>233<br>233<br>234<br>235<br>.235  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4  | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .   | perations   | <ul> <li>232</li> <li>233</li> <li>233</li> <li>234</li> <li>235</li> <li>235</li> <li>237</li> </ul>  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U   | berations<br>bus Tasks<br>on<br>horization via Azure Entra ID | 232<br>233<br>234<br>235<br>.235<br>237<br>238   |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin   | berations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>.238  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi   | berations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>.238<br>.238  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod  | berations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>.238<br>.238<br>.238<br>.239<br>240   |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele  | berations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>.238<br>.238<br>.238<br>.239<br>.240  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con  | berations   | 232<br>233<br>234<br>235<br>235<br>235<br>237<br>238<br>.238<br>.238<br>.238<br>.239<br>.240<br>240  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5                                     | Multiple Op<br><b>T API</b><br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Available   | berations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>239<br>240<br>240<br>241  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7                       | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avai<br>Creating Na   | berations   | 232<br>233<br>234<br>235<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>239<br>240<br>240<br>240<br>241  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8                | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avail<br>Creating Net   | perations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>240<br>240<br>241<br>241<br>241  |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8<br>5.0         | Multiple Op<br><b>T API</b><br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avail<br>Creating Net<br>5.8.1 Gett   | perations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>239<br>240<br>240<br>240<br>241<br>241<br>241<br>242<br>243   |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8<br>5.9         | Multiple Op<br><b>T API</b><br>Overview<br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avai<br>Creating Net<br>5.8.1 Gett<br>Checking S<br>5.9.1 View  | perations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>240<br>240<br>240<br>241<br>241<br>241<br>242<br>243<br>245<br>248                                    |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8<br>5.9         | Multiple Op<br><b>T API</b><br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avai<br>Creating Net<br>5.8.1 Gett<br>Checking S<br>5.9.1 View<br>5.9.2 Chec                             | perations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>240<br>240<br>241<br>241<br>241<br>242<br>243<br>245<br>248<br>250                      |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8<br>5.9         | Multiple Op<br><b>T API</b><br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avai<br>Creating Ne<br>5.8.1 Gett<br>Checking S<br>5.9.1 View<br>5.9.2 Chec<br>5.9.3 Chec                             | perations   | 232<br>233<br>234<br>235<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>240<br>240<br>240<br>240<br>241<br>241<br>242<br>242<br>243<br>245<br>.248<br>.250<br>.250            |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8<br>5.9         | Multiple Op<br><b>T API</b><br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avai<br>Creating Net<br>5.8.1 Gett<br>Checking S<br>5.9.1 View<br>5.9.2 Chec<br>5.9.3 Chec<br>5.9.4 Upda              | perations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>240<br>240<br>241<br>241<br>241<br>241<br>242<br>243<br>245<br>245<br>248<br>250<br>251        |
| 5 | 4.21<br><b>RES</b><br>5.1<br>5.2<br>5.3<br>5.4<br>5.5<br>5.6<br>5.7<br>5.8<br>5.9<br>5.10 | Multiple Op<br><b>T API</b><br>Asynchrono<br>Authorizatio<br>5.3.1 Auth<br>Discovery .<br>Managing U<br>5.5.1 Listin<br>5.5.2 Addi<br>5.5.3 Mod<br>5.5.4 Dele<br>Global Con<br>5.6.1 Upda<br>Listing Avai<br>Creating Ne<br>5.8.1 Gett<br>Checking S<br>5.9.1 View<br>5.9.2 Chec<br>5.9.3 Chec<br>5.9.4 Upda<br>Scaling Me | perations   | 232<br>233<br>234<br>235<br>235<br>237<br>238<br>238<br>238<br>238<br>238<br>238<br>238<br>239<br>240<br>240<br>241<br>241<br>241<br>242<br>243<br>245<br>243<br>245<br>248<br>250<br>251<br>252 |

|   |      | 5.10.2 Scale In Operation   | 252 |
|---|------|---|-----|
|   |      | 5.10.3 Scale To Operation   | 253 |
|   | 5.11 | Modifying Stack Configuration   | 254 |
|   |      | 5.11.1 Update Operation   | 255 |
|   | 5.12 | Stopping and Starting Stack   | 256 |
|   |      | 5.12.1 Stopping Stack   | 256 |
|   |      | 5.12.2 Starting Stack   | 256 |
|   | 5.13 | Rebooting Stack Components  | 257 |
|   | 5.14 | Deleting Stack  | 258 |
|   |      | 5.14.1 Purging Deleted Stack  | 258 |
|   | 5.15 | Healing Stack   | 259 |
|   | 5.16 | Rebuilding Stack  | 259 |
|   | 5.17 | Managing Files  | 260 |
|   |      | 5.17.1 Listing Files  | 260 |
|   |      | 5.17.2 Adding File  | 260 |
|   | E 40 | 5.17.3 Deleting File  | 201 |
|   | 5.18 | Upgrading Stack   | 202 |
|   |      | 5.18.1 Hosting Software Load (CMP) Files on Stack Manager<br>5.18.2 Upgrading Software on Idle Media Components | 263 |
|   | 5.19 | Shelving and Unshelving Stack   | 264 |
|   |      | 5.19.1 Shelving Stack   |     |
|   |      | 5.19.2 Unshelving Stack   | 264 |
|   | 5.20 | Resetting Stack Password  | 265 |
|   | 5.21 | Sending INI File  | 266 |
| 6 | Оре  | rational Logs   | 267 |
|   | 6.1  | Web Server Logs   | 268 |
| 7 | Stac | ks Management   | 269 |
|   | 7.1  | Automatic Stop / Start / Shelve   | 269 |
|   | 7.2  | Tagging Stack Resources   | 270 |
|   | 7.3  | Integration with Azure Application Insights   | 270 |
| 8 | Seci | ure Deployment  | 273 |
| - |      |   |     |



This page is intentionally left blank.

#### **Notice**

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

This document is subject to change without notice.

Date Published: June-04-2025

#### Security Vulnerabilities

All security vulnerabilities should be reported to <u>vulnerability@audiocodes.com</u>.

#### **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 <a href="https://www.audiocodes.com/services-support/maintenance-and-support">https://www.audiocodes.com/services-support/maintenance-and-support</a>.

#### Stay in the Loop with AudioCodes



#### **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 <a href="https://online.audiocodes.com/documentation-feedback">https://online.audiocodes.com/documentation-feedback</a>.

#### **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   |
|-------|---|
| 28931 | Initial document release for Version 7.4  |
| 28932 | New sections - Migrating to a New Virtual Machine; Secure Connection to Web<br>Interface; Installing Let's Encrypt Certificates; Login via Azure Active Directory;<br>Resetting Web Interface Credentials; Authentication via Azure Active Directory  |
| 28933 | Typos   |
| 28935 | Creating a new stack updates; Mediant VE for Azure/AWS/Google updates   |
| 28936 | Update to acquiring access token using REST client  |
| 28937 | Debian Linux 10 and 11 support; updates to messages in creating new stack and checking stack state and configuration (REST API)   |
| 28938 | Debian 11; backup and restore; configuring hostname for stack manager VM;<br>acquiring certificate from CA; enforcing secure connection; creating VoiceAI Connect<br>in AWS/Azure; resetting stack password; shelving and unshelving stack; viewing IP<br>addresses of stack components; stacks management (automatic stop / start / shelve,<br>tagging stack resources, integration with Azure Application Insights) |
| 28939 | Typo in redirect URL; typo in Storage > Storage Admin   |
| 28950 | Commands updated for changing Web interface credentials; trademark and USA address updated  |
| 28951 | Mediant VE HA in Azure support.   |
| 28952 | Creating Debug file; common_tags; tags (Azure - VE); VoiceAI Connect commands (center_tags, sm_tags, sbc_tags)  |
| 28953 | Updates to 'Using Pre-Defined Private IP Addresses' section   |
| 28954 | Google cloud prerequisites; management ports updated; automatic update URL for<br>"worker" and front-end SBCs; bot dialplan; creating VoiceAl Connect stack in Google<br>cloud; new advanced parameters auto_shelve_time / shelve_delete_ips /<br>auto_start_time / auto_stop_time; hosting software load (cmp) file on Stack Manager   |
| 28954 | Typo in path ("/var/stack_mgr/bin/stack_mgr")   |
| 28955 | Miscellaneous   |
| 28956 | Supported operating systems and versions; uploading CMP file  |
| 28957 | AWS multi-zone deployment updates; component name for pre-defined public IP address   |
| 28958 | "Microsoft.ManagedIdentity/userAssignedIdentities/*" added; Advanced configuration parameters added (main_nsg_id, nsg_id_sc_ethX, nsg_id_mc_ethX, nsg_id_ethX, resource_group)  |
| 28959 | Significant updates – mainly to configuration parameters, IPv6 parameters, secure deployment (new section).   |
| 28965 | Significant updates (e.g., OS versions; prerequisites; installation; web interface (accessing, levels, managing users, passwords, global parameters; upgrading; Azure Entra ID replaced AD; debug file; securing web connection)  |
| 28966 | IAM roles updated (cloudformation:CreateChangeSet,<br>cloudformation:DeleteChangeSet, cloudformation:DescribeChangeSet, and<br>cloudformation:ExecuteChangeSet)   |

| LTRT  | Description  |
|-------|--|
| 28968 | Section Creating Amazon Web Services (AWS) Instance updated with t3.medium, Standard_B2s and Standard_B2ls_v2, 4 GB memory for e2-medium |

#### **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 <a href="https://online.audiocodes.com/documentation-feedback">https://online.audiocodes.com/documentation-feedback</a>.



This page is intentionally left blank.

## **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



This page is intentionally left blank.

## 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 <u>http://aws.amazon.com/</u>.
- 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 (<u>https://console.aws.amazon.com/iam</u>).
- 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: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"
```



**Note:** 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 <u>http://azure.microsoft.com</u>.
- 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 <u>http://cloud.google.com</u>.
- 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



**Note:** 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

| aws | Services                                   | <b>Q</b> Search  |                  |                   |                          | [AI           | t+S]  | D 4 0  |
|-----|--|--|------------------|-------------------|--------------------------|---------------|---|--|
| •   | EC2 🕻 Instan                               | ces > Launch ar  | n instance       |                   |                          |               |   | ▼ Summary  |
|     | Launch<br>Amazon EC2 a<br>following the s  | an instan<br>lows you to create<br>imple steps below.                | Virtual machine  | es, or instances, | that run on the <i>i</i> | AWS Cloud. C  | uickly get started by   | Number of instances Info   |
|     | Name ar                                    | d tags Info  |                  |                   |                          |               |   | Software Image (AMI)<br>Amazon Linux 2023 AMI 2023.1.2read more<br>ami-051f7e7f6c2f40dc1   |
|     | Name<br>e.g. My W                          | b Server   |                  |                   |                          | 4             | dd additional tags  | Virtual server type (instance type)<br>t2.micro  |
|     | T Appli                                    | ation and OS   | Images (Am       | azon Machi        |                          | 4-            |   | Firewall (security group)<br>New security group  |
|     | An AMI is a                                | template that cont   | tains the softwa | re configuration  | (operating system        | em. applicati | on server, and  | Storage (volumes)<br>1 volume(s) - 8 GiB   |
|     | application<br>below<br>Q Search<br>My AMI | a) required to launce<br>our full catalog in<br>Quick Start<br>macOS | th your instance | Search or Brow    | d OS images              | SUSE Li       | that you are looking for  | ● Free tier: In your first year includes<br>750 hours of t2.micro (or t3.micro in<br>the Regions in which t2.micro is<br>unavailable) instance usage on free<br>tier AMS per month, 30 GiB of EBS<br>storage, 2 million 105, 1 GB of<br>snapshots, and 100 GB of bandwidth<br>to the internet. |
|     | aws  | Mac  | ubuntu®          | Microsoft         | 🝓 Red Hat                | SUS           | Browse more AMIs<br>Including AMIs from<br>AWS, Marketplace and | Cancel Launch instance   |

- **3.** In the 'Name' field, type the name of the EC2 Instance (e.g., "STACK-MGR").
- 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 **t3.medium**.
- 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.

8. 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)<br>172.31.0.0/16   | (default) 🔻 | C |                     |
| Subnet Info   |             |   |                     |
| subnet-a704e9fe<br>VPC: vpc-f8b7159d Owner: 516086831279 Availability Zone: us-east-1c<br>IP addresses available: 4077 CIDR: 172.31.0.0/20) | main<br>T   | C | Create new subnet 🗹 |
| Auto-assign public IP Info  |             | _ |                     |
| Enable  |             | , |                     |

**9.** 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

| Type Info   | Protocol Info  | Port range Info   |
|---|--|---|
| ssh   | ТСР  | 22  |
| Source type Info  | Source Info  | Description - optional Info   |
| Anywhere  | ▼ Q Add CIDR, prefix list or security  | e.g. SSH for admin desktop  |
|   | 0.0.0.0/0 🗙  |   |
| Security group rule 2 (TCP, 80,   | , 0.0.0.0/0)   | Remove  |
|   |  |   |
| Type Info   | Protocol Info  | Port range Info   |
| Type Info<br>HTTP   | Protocol Info<br>▼ TCP   | Port range Info<br>80   |
| Type Info<br>HTTP<br>Source type Info   | Protocol Info     TCP     Source Info  | Port range Info<br>80<br>Description - optional Info  |
| Type Info<br>HTTP<br>Source type Info<br>Anywhere   | Protocol Info     TCP     Source Info     Q. Add CIDR, prefix list or security   | Port range Info<br>80<br>Description - optional Info<br>e.g. SSH for admin desktop  |
| Type Info<br>HTTP<br>Source type Info<br>Anywhere   | <ul> <li>Protocol Info</li> <li>TCP</li> <li>Source Info</li> <li>Q. Add CIDR, prefix list or security</li> <li>0.0.0/0 ×</li> </ul>   | Port range Info<br>80<br>Description - optional Info<br>e.g. SSH for admin desktop  |
| Type Info<br>HTTP<br>Source type Info<br>Anywhere<br>Security group rule 3 (TCP, 44   | Protocol Info  TCP  Source Info  Add CIDR, prefix list or security  0.0.0.0/0 ×  3, 0.0.0/0)   | Port range Info<br>80<br>Description - optional Info<br>e.g. SSH for admin desktop<br>Remove  |
| Type Info<br>HTTP<br>Source type Info<br>Anywhere<br>Security group rule 3 (TCP, 44<br>Type Info                              | Protocol Info  TCP  Source Info  Add CIDR, prefix list or security  0.0.0.0/0 ×  3, 0.0.0/0)  Protocol Info  | Port range Info<br>80<br>Description - optional Info<br>e.g. SSH for admin desktop<br>Remove  |
| Type Info<br>HTTP<br>Source type Info<br>Anywhere<br>Security group rule 3 (TCP, 44<br>Type Info<br>HTTPS                     | <ul> <li>Protocol Info</li> <li>TCP</li> <li>Source Info</li> <li>Q. Add CIDR, prefix list or security</li> <li>0.0.0.0/0 ×</li> <li>3, 0.0.0.0/0</li> <li>Protocol Info</li> <li>TCP</li> </ul> | Port range Info<br>80<br>Description - optional Info<br>e.g. SSH for admin desktop<br>Remove<br>Port range Info<br>443                                |
| Type Info<br>HTTP<br>Source type Info<br>Anywhere<br>Security group rule 3 (TCP, 44<br>Type Info<br>HTTPS<br>Source type Info | Protocol Info  TCP  Source Info  Add CIDR, prefix list or security  0.0.0.0/0 ×  3,0.0.U/0)  Protocol Info  TCP  Source Info   | Port range Info<br>80<br>Description - optional Info<br>e.g. SSH for admin desktop<br>Remove<br>Port range Info<br>443<br>Description - optional Info |

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

**11.** 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

| ▼ Advanced details Info  |   |                        |
|--|---|------------------------|
| Purchasing option Info   |   |                        |
| Request Spot Instances   |   |                        |
| Domain join directory Info                                       |   |                        |
| Select   | G | Create new directory   |
| IAM instance profile Info  |   |                        |
| STACK_MGR<br>arn:awsiam::516086831279:instance-profile/STACK_MGR | G | Create new IAM profile |

#### **12.** Click Launch instance.

- **13.** Wait until the instance is successfully launched.
- **14.** 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         |

- **15.** 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.
- **16.** 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 <u>https://portal.azure.com/</u>.
- 2. Navigate to Azure Marketplace (All services > Marketplace).
- **3.** Search for the product "Mediant CE Session Border Controller (SBC)" published by AudioCodes.

| ≡ Microsoft Azure     | P Search resources, services, and docs (G+/)   | ? 😳 |
|-----------------------|--|-----|
| Home > Marketplace    |  |     |
| Marketplace           |  | Ŕ   |
| My Saved List         | P mediant ce X Pricing : All Operating System : All Publisher : All  |     |
| Recently created      |  |     |
| Service Providers     | Showing All Results  |     |
| Categories            |  |     |
| Get Started           |  |     |
| AI + Machine Learning | Mediant CE Session Border<br>Controller (SBC)  |     |
| Analytics             | AudioCodes   |     |
| Blockchain            | Direct SIP connectivity to enable<br>voice services in Microsoft Teams or<br>Crister de Microsoft |     |
| Compute               | atype ini obaliteas  |     |
| Containers            |  |     |

#### Figure 2-6: Azure Marketplace

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



| Home > Marketplace > Mediant CE Session Border Controller (SBC)  |      |
|--|------|
| Mediant CE Session Border Controller (SBC)<br>AudioCodes   | \$ X |
| Mediant CE Session Border Controller (SBC)       Save for later         AudioCodes       Create  |      |
| Overview Plans   |      |
| Looking to enable Microsoft Teams Direct Routing or connect SIP trunks to Microsoft Skype for Business?  |      |
| AudioCodes' Mediant Session Border Controlliers (SECS) deliver seamless connectivity, enhanced security and quality assurance for enterprise and service provider VoIP<br>networks. By running on Azure virtual machines within the enterprise environment, AudioCodes' SBCs provide an effective demarcation point between Microsoft Teams or<br>Stype for Dusiness and the SIP trunk.  |      |
| AudioCodes' SBCs perform SIP protocol mediation, translation and media handling (better known as interoperability), while also securing the enterprise VoIP network. In addition, AudioCodes' SBCs can connect virtually any existing IP-PBX to Microsoft Teams or Skype for Business to enable coexistence and migration options.   |      |
| AudioCodes' SBCs are certified for Microsoft Teams Direct Routing and Skype for Business. They provide complete coverage of unique enterprise requirements with a highly<br>scalable portfolio to provide the necessary interoperability and reliability requirements. Please click here for more information on AudioCodes' SBC session capacities, media<br>handling and capabilities.   |      |
| Mediant Cloud Edition (CE) Session Border Controller (SBC) leverages the advantages of cloud agility to allow enterprises and service providers to fully realize the potential of<br>virtual environments by offering full cloud elasticity that rapidly adjusts to changing needs. The Mediant CE automatically provides extra capacity when required and scales<br>back when demand drops. Its microservices architecture, combined with a scalable media cluster, enables new revenue-generating communications services to be introduced<br>simply and cost-effectively. |      |
| This offer creates Stack Manager virtual machine that is used to deploy Mediant CE cluster and manage its complete lifecycle. Refer to the Mediant CE Installation Manual and<br>Stack Manager User Manual for detailed instructions on how to proceed with the Mediant CE deployment.   |      |
| Useful Links<br>Mediant Cloud Edition (CE) SBC - Installation Manual<br>Stack Manager for Mediant CE SBC - User Manual   |      |

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

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



- **a.** In the 'Virtual Machine name' field, enter a unique name for the new virtual machine.
- b. 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.

**Note:** 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.
  - **c.** From the 'Subscription' drop-down list, select a proper subscription for your deployment.
  - **d.** Under 'Resource group', click **Create new**, and then enter a new Resource Group name for your deployment.
  - e. From the 'Location' drop-down list, select a proper location for your deployment.
  - f. 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

| Home > Marketplace > Mediant CE Session Borde                      | er Controller (SBC) > Create Mediant CE Session Borde                               | er Controller (SBC) $>$ Virtual Machine Settings $>$ Subne |
|--|---|--|
| Create Mediant CE Session Bo $\times$                              | Virtual Machine Settings $\qquad 	imes$   | Subnet 🗆 🗙   |
| 1 Basics ~   | Virtual machine size * ①<br>1x Standard B1ms<br>1 vcpu, 2 GiB memory<br>Change size | Subnet *   |
| 2 Virtual Machine Settings ><br>Configure virtual machine settings | *Virtual network ① > StackMgrNetwork  |  |
| 3 Summary ><br>Mediant CE Session Border Contr                     | Subnet ()<br>Configure subnets  |  |
| 4 <sup>Buy</sup> >   | *Public IP Address () > (new) stack-mgr-ip  |  |
|  | Public DNS Prefix ①<br>stack-mgr-83892bb079 ✓<br>westeurope.cloudapp.azure.com      |  |
|  | ОК  | ОК   |

- a. Choose the Virtual machine size. Standard\_B2s or Standard\_B2ls\_v2 instances are 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

| Home > Marketplace > Mediant CE Session Border Controller (SBC) > Create Mediant CE Session Border Controller (SBC |
|--|
| Create public IP address   |
| News   |
| Name ^   |
| stack-mgr-ip   |
| sku 🛈  |
| 🔿 Basic 🔘 Standard   |
|  |
|  |
|  |
|  |

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

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



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

#### Figure 2-12: Buy – Step 4

| te Mediant CE Session Bo $	imes$ Create    |   | ] |
|--|---|---|
| 1 Basics  V Done                           | Mediant CE Session Border Controller (SBC)<br>by AudioCodes<br>Terms of use   privacy policy  |   |
| $2  \  \  \  \  \  \  \  \  \  \  \  \  \$ | Deploying this template will result in various actions being performed, which may include<br>the deployment of one of more Azure resources or Marketplace offerings and/or<br>transmission of the information you provided as part of the deployment process to one or<br>more parties, as specified in the template. You are responsible for reviewing the text of<br>the template to determine which actions will be performed and which resources or |   |
| 3 Summary  Mediant CE Session Border Contr | offerings will be deployed, and for locating and reviewing the pricing and legal terms<br>associated with those resources or offerings.<br>The legal terms associated with any Marketplace offering may be found in the Azure   |   |
| <b>4</b> <sup>Buy</sup> >                  | portal. For pricing information and to determine which offerings may be purchased using<br>monetary commitment funds or subscription credits, please contact your reseller. If any<br>Microsoft products are included in a Marketplace offering (e.g., Windows Server or SQL<br>Server), such products are licensed by Microsoft and not by any third party.  |   |
|  | Template deployment is intended for advanced users only. If you are uncertain which<br>actions will be performed by this template, which resources or offerings will be deployed,<br>or what prices or legal terms pertain to those resources or offerings, do not deploy this<br>template.   |   |
|  | Terms of use  |   |

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

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

| ■ Microsoft Azure                   | $\mathcal{P}$ Search resources, services, and docs (G+/) |  | >_ 🗣 🗳 🏶 ? 😳                    |
|-------------------------------------|--|--|---------------------------------|
| Home > audiocodes.mediant_ce_sbc-20 | 0191119093113 - Overview > stack-mgr                     |  |                                 |
| stack-mgr                           |  |  | \$ X                            |
|                                     | Stop Start C Restart Stop Stop Capture                   | 🕽 Delete 💍 Refresh   |                                 |
| Overview                            | Resource group (change) : StackMgrRG                     | Size : Standard B1m  | ns (1 vcpus, 2 GiB memory)      |
| Activity log                        | Status : Running   | Public IP address : 51.105.185.25<br>Private IP address : 10.3.1.6 | 32                              |
| Access control (IAM)                | Subscription (change) : SBC Lab                          | Virtual network/subnet : StackMgrNet                               | work/oam                        |
| 🔷 Tags                              | Subscription ID : 4ad554cf-0b4e-4a65-8a14-2b6951a3d1d    | B DNS name : stack-mgr-83  | 892bb079.westeurope.cloudapp.az |
| Diagnose and solve problems         | Computer name : stack-mgr                                | Scale Set : N/A  |                                 |
| Settings                            | Operating system : Linux (ubuntu 18.04)                  |  |                                 |
| A Networking                        | Tags (change) : Click here to add tags                   | *  |                                 |
| a Disks                             |  |  |                                 |
| 📮 Size                              | Show data for last: 1 hour 6 hours 1                     | 2 hours 1 day 7 days 30 days                                       |                                 |
| Security                            |  |  |                                 |
| 🛃 Extensions                        | CPU (average)  | Network (total)  | 1                               |
| 🐔 Continuous delivery (Preview)     | 100%   | 1008   |                                 |
| Availability + scaling              | 80%  | 80B  |                                 |
| Configuration                       | · · · · · · · · · · · · · · · · · · ·                    |  | -                               |

#### Figure 2-13: Determining Public IP Address

#### **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

| stack-mgr - Networking      |                 |   |                  |                       |                        |                  |                  |          |
|-----------------------------|-----------------|---|------------------|-----------------------|------------------------|------------------|------------------|----------|
| ○ Search (Ctrl+/) 《         | 🔊 Attach netv   | work interface $\mathcal{B}^{q}$ Detach network int | erface           |                       |                        |                  |                  |          |
| Overview                    | Network I       | nterface: stack-mgr-nic Effectiv                    | e security rule  | s Topology            |                        |                  |                  |          |
| Activity log                | Virtual network | k/subnet: StackMgrNetwork/oam N                     | IC Public IP: 51 | .105.185.252 NI       | C Private IP: 10.3.1.6 | Accelerated netw | orking: Disabled | I        |
| Access control (IAM)        | Inbound por     | rt rules Outbound port rules Ap                     | plication secu   | rity groups Load b    | alancing               |                  |                  |          |
| Tags                        | Natwork co      |   | d to potwork i   | storface, stack mar i | aid)                   |                  |                  |          |
| Diagnose and solve problems | Impacts 0 si    | ubnets, 1 network interfaces                        | I to hetwork i   | ntenace, stack-myr-r  |                        |                  | Add inbound po   | ort rule |
| ttings                      | Priority        | Name  | Port             | Protocol              | Source                 | Destination      | Action           |          |
| Networking                  | 500             | 🔺 ssh   | 22               | TCP                   | Any                    | Any              | Allow            |          |
| Disks                       | 510             | http  | 80               | TCP                   | Any                    | Any              | Allow            |          |
| Size                        | 520             | https   | 443              | TCP                   | Any                    | Any              | Allow            |          |
| Security                    | 65000           | AllowVnetinBound                                    | Any              | Any                   | VirtualNetwork         | VirtualNetwork   | Allow            | ••       |
|                             | 65001           | AllowAzureLoadBalancerinBound                       | Any              | Any                   | AzureLoadBalan         | Any              | Allow            | •        |
| Extensions                  |                 |   |                  |                       |                        |                  |                  |          |

**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 <a href="https://console.cloud.google.com/compute">https://console.cloud.google.com/compute</a>.
- 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-medium** (2 shared vCPUs, 4-GB memory).
- 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.

| ÷  | Create an instance  |  | E HELP ASSISTANT  |
|----|---|--|---|
| To | reate a VM instance, select one of the options:   | Name *   | Monthly estimate  |
| Ħ  | New VM instance<br>Create a single VM instance from scratch   | stackmgr 😧                                     | \$13.23<br>That's about \$0.02 hourly                         |
| Ŧ  | New VM instance from template<br>Create a single VM instance from an existing<br>template           | ADD LABELS                                     | Pay for what you use: No upfront costs and per second billing |
| =  | New VM instance from machine image<br>Create a single VM instance from an existing<br>machine image | Region is permanent     Cone 's permanent      |   |
|    | Marketplace<br>Deploy a ready-to-go solution onto a VM instance                                     | <section-header><form></form></section-header> |   |

Figure 2-15: Create Google Cloud Instance

- **12.** 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.
- **13.** 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

| etails         | Please provide the initial hostname for the instance, to<br>count. Increase the Count to create multiple instance | the availability zone where it will be deployed, and the instance<br>is with the same settings. |
|----------------|---|---|
| ource *        | Instance Name *   | Total Instances   |
| lavor *        | stack-mgr   | (40 Max)  |
| etworks *      | Description   | 23%   |
| etwork Ports   | Availability Zone   | 8 Current Usage   |
| ecurity Groups | nova  | ▼ 31 Remaining  |
| ey Pair        | Count *   |   |
| onfiguration   | 1   |   |
| erver Groups   |   |   |
| cheduler Hints |   |   |
| etadata        |   |   |

- 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

| Details *       | Instance source is<br>snapshot), a volum | the template used to create a<br>le or a volume snapshot (if el | in instance. Y<br>nabled). You | rou can use a<br>can also choc | n image, a sna<br>se to use persi | pshot of an instar<br>stent storage by ( | creating a |
|-----------------|--|---|--------------------------------|--------------------------------|-----------------------------------|--|------------|
| Source          | Select Boot Source                       | ce  |                                | Create Nev                     | Volume                            |  |            |
| Flavor *        | Image                                    |   | ~                              | Yes N                          | o                                 |  |            |
| Networks *      | Volume Size (GB)                         | •   |                                | Delete Volu                    | ime on Instan                     | ce Delete                                |            |
| Network Ports   | 10                                       |   | \$                             | Yes                            | 0                                 |  |            |
| Security Groups | Allocated                                |   |                                |                                |                                   |  |            |
| Key Pair        | Name                                     | Updated   | Siz                            | e                              | Туре                              | Visibility                               |            |
| Configuration   | > debian-11                              | 1/25/22 10:02 AM  | 310                            | 0.25 MB                        | qcow2                             | Public                                   | •          |
| Server Groups   | ✓ Available 1                            | 3   |                                |                                |                                   |  | Select on  |
| Scheduler Hints | Q debian                                 |   |                                |                                |                                   |  | ×          |
| Metadata        | Name                                     | Updated   | Size                           | Тур                            | e                                 | Visibility                               |            |

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

| Details         | Flavors manage<br>Allocated | the sizing for | r the comput | le, memory and | I storage capacity | of the instance. |        |          |
|-----------------|-----------------------------|----------------|--------------|----------------|--------------------|------------------|--------|----------|
| Source          | Name                        | VCPUS          | RAM          | Total Disk     | Root Disk          | Ephemeral Disk   | Public |          |
| Flavor          | > m1.small                  | 1              | 2 GB         | 20 GB          | 20 GB              | 0 GB             | Yes    | ¥        |
| Networks *      | ✓ Available                 | 11             |              |                |                    |                  |        | Select o |
| Network Ports   | <b>Q</b> m1                 |                |              |                |                    |                  |        | :        |
| Security Groups | Name                        | VCPUS          | RAM          | Total Disk     | Root Disk          | Ephemeral Disk   | Public |          |
| Key Pair        | > m1.tiny                   | 1              | 512 MB       | 1 GB           | 1 GB               | 0 GB             | Yes    | 1        |
| Configuration   | > m1.medium                 | 2              | 4 GB         | 40 GB          | 40 GB              | 0 GB             | Yes    | 1        |
| Server Groups   |                             |                |              |                |                    |                  |        |          |
| Scheduler Hints | > m1.large                  | 4              | 8 GB         | 80 GB          | 80 GB              | 0 GB             | Yes    | 1        |
| Metadata        | > m1.xlarge                 | 8              | 16 GB        | 160 GB         | 160 GB             | 0 GB             | Yes    | ↑        |

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

| Launch Instance |                           |                                 |                   |                |                    | ×        |
|-----------------|---------------------------|---------------------------------|-------------------|----------------|--------------------|----------|
| Details         | Networks provide the comm | nunication channels for instant | ces in the cloud. | Colect potwork | from these liste   |          |
| Source          | Network                   | Subnets Associated              | Shared            | Admin State    | Status             | I Delow. |
| Flavor          | ♦1 > flat network         | flat subnet                     | Yes               | Up             | Active             | •        |
| Networks        | _                         | -                               |                   |                |                    |          |
| Network Ports   | ✓ Available ④             |                                 |                   | Se             | elect at least one | network  |
| Security Groups | Q Click here for filters. |                                 |                   |                |                    | ×        |
| Key Pair        | Network                   | Subnets Associated              | Shared            | Admin State    | Status             |          |
| Configuration   | > private_network         | private_subnet                  | No                | Up             | Active             | •        |
| Server Groups   | > private6_network        | private6_subnet                 | No                | Up             | Active             | 1        |
| Scheduler Hints | > public_network          | public_subnet                   | Yes               | Up             | Active             | •        |
| Metadata        | > internal_network        | internal_subnet                 | Yes               | Up             | Active             | *        |
|                 |                           |                                 |                   |                |                    |          |
| × Cancel        |                           |                                 |                   | < Back Next >  | ▲ Launch Ins       | tance    |

- **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. |

| letails        | Select the security groups | to launch the instance in. |                  |
|----------------|----------------------------|----------------------------|------------------|
| ource          | Name                       | Description                |                  |
| lavor          | > default                  | Default security group     | 4                |
| letworks       | > all                      |                            | 4                |
| letwork Ports  |                            |                            |                  |
| ecurity Groups |                            |                            | Select one or mo |
| ev Pair        | Q Click here for filters   | 5.                         | >                |
|                | Name                       | Description                |                  |
| onfiguration   | > sbc                      |                            | 1                |
| erver Groups   |                            |                            |                  |
| cheduler Hints |                            |                            |                  |
| letadata       |                            |                            |                  |

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

**13.** Click **Next**; the Key Pair wizard page appears.

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

| Launch Instance |   |  |   | ×           |
|-----------------|---|--|---|-------------|
| Details         | A key pair allows you to SSH<br>pair, or generate a new key p | into your newly created instance. You air. | may select an existing key pair, import a | a key 🕜     |
| Source          | + Create Key Pair   | mport Key Pair                             |   |             |
| Flavor          | Allocated   |  |   |             |
| Networks        | Name Finge  | erprint                                    |   |             |
| Network Ports   | > admin 0b:7c   | :9d:f4:36:29:7b:6e:b2:2f:d8:e4:72:20:7     | <sup>72-75</sup>                          | •           |
| Security Groups | Dianta ditem  |  |   |             |
| Key Pair        | Displaying Titem  |  |   |             |
| Configuration   | ✓ Available ①   |  |   | Select one  |
| Server Groups   | Q Click here for filters.                                     |  |   | ×           |
| Scheduler Hints | Displaying 1 item   |  |   |             |
| ocheddior filma | Name  | Fingerprint                                |   |             |
| Metadata        | > aws-ssh-frankfurt-1   | ba:6c:f8:46:4f:37:40:02:2c:33              | :74:59:cb:fd:ad:fd                        | <b>^</b>    |
|                 | Displaying 1 item   |  |   |             |
|                 |   |  |   |             |
| × Cancel        |   |  | < Back Next > 🔓 Laund                     | ch Instance |

#### 14. Click Launch Instance.

**15.** 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.



**Note:** 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 <u>https://tools.audiocodes.com/install/index.html</u> 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

C stack\_mgr

#### **Stack Manager**

| Admin |       |  |
|-------|-------|--|
| ••••• |       |  |
|       | Login |  |

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.



**Note:** 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.

| Modify user                              |                          |
|--|--------------------------|
| Name                                     | Admin                    |
| Password                                 | *****                    |
| Туре                                     | Security Administrator 💙 |
| Status                                   | Active 🗸                 |
| Password Expiration<br>In Days (0=never) | 0                        |
| Modify Cancel                            |                          |

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.



**Note:** 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.



**Note:** 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 <u>AWS documentation</u>.
- 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 <u>https://portal.azure.com/</u>.
- 2. Navigate to Subscriptions (All services > Subscriptions).
- **3.** Locate your Azure Subscription ID.

#### Figure 2-24: Locating Subscription ID

| SBC Lab                       |   |   |
|-------------------------------|---|---|
|                               | « 🗹 Manage - 🔿 Two (for - 🏛 Cancel subscription | ✓ Rename → Change directory             |
| <ul> <li>Overview</li> </ul>  | Subscription ID<br>4ad554cf-0                   | Subscription name<br>SBC Lab            |
| Access control (IAM)          | audiocodes itd (audiocodes365.onmicrosoft.com)  | Current billing period<br>Not available |
| X Diagnose and solve problems | Owner<br>Offer                                  | Not available<br>Status                 |
| Security (Preview)            | Enterprise Agreement<br>Offer ID                | Active                                  |

- 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.

|                | (                   | Configuration        |                 |
|----------------|---------------------|----------------------|-----------------|
|                | General             |                      | Microsoft Azure |
| Name Prefix    |                     | Tenant ID            |                 |
| dmin Username  | Admin               | Client ID            |                 |
| Admin Password |                     | Secret               |                 |
|                |                     | Subscription ID      | 4ad554cf-       |
|                | Amazon Web Services | Blob Account<br>Name |                 |
| Access Key     |                     | Blob Account Key     |                 |
| Secret Key     |                     | Blob Container       |                 |
| S3 Bucket      |                     |                      |                 |
| S3 Prefix      |                     |                      | Google          |
|                |                     | Project              |                 |
|                | Openstack           | Credentials          |                 |
| Cloud Name     |                     | Storage Bucket       |                 |
| Container      |                     | Storage Prefix       |                 |

Figure 2-25: Configuring Azure Subscription ID

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 <u>http://portal.azure.com</u>.
- 2. Navigate to the Virtual Machines page.
- 3. Select the Stack Manager virtual machine.

- 4. 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:
- 1. Open the Azure portal at <u>http://portal.azure.com.</u>
- 2. Navigate to the Subscriptions page.
- **3.** Select your subscription.
- 4. In the Navigation menu, click Access Control (IAM), and then click Add a role assignment:
  - a. From the 'Role' drop-down list, select Contributor.
  - 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.

| Add role assignment  | 2 |
|--|---|
| Role (j  |   |
| Contributor V  |   |
| Assign access to 🕕   |   |
| Virtual Machine V  |   |
| Subscription *   |   |
| SBC Lab 🗸  |   |
| Select ①   |   |
| Search by name   |   |
| alex-ubuntu-1<br>/subscriptions/4ad554cf-0b4e-4a65-8a14-2b695  |   |
| Selected members:  |   |
| alex-stack-mgr-2<br>/subscriptions/4ad554cf-0b4e-4a65-8 Remove |   |

#### 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<br>only (e.g., virtual networks and subnets). This is needed for displaying<br>"Create new stack" Web UI dialog and validating stack configuration<br>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).<br>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,<br>modify and delete stack resources (e.g., virtual machines, network<br>interfaces, load balancers).  |
|                 | The role is assigned only for specific Resource Group(s) that must be pre-created prior to stack deployment.  |



**Note:** 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**.



```
"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}"
   1,
    "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 <u>Azure documentation</u> for detailed instructions on how to create custom IAM roles.

- 2. Open the Azure portal at <u>http://portal.azure.com.</u>
- **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.
- 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.

#### 2.8.2.2.2.1 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/<br>virtualNetworks/read   | 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.  |
|  | Network name prior to stack deployment. If the wrong name is provided, stack deployment will fail.  |
| Microsoft.Network/<br>virtualNetworks/subnets/read                                       | 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.   |
|  | Stack Manager will not be able to validate Subnet names<br>prior to stack deployment. If wrong names are provided,<br>stack deployment will fail.   |
|  | You must specify the CIDR for each subnet through the<br>Advanced Config parameters: cluster_subnet_cidr,<br>main_subnet_cidr, additional1_subnet_cidr,<br>additional2_subnet_cidr. Otherwise, Stack Manager will<br>not be able to properly configure network interfaces for<br>deployed stack components. |
| Microsoft.Network/<br>publicIPAddresses/read   | Stack Manager will not be able to validate predefined<br>Public IP addresses provided via <b>public_ip_*</b> Advanced<br>Config parameters. If wrong names are provided, stack<br>deployment will fail.   |
| Microsoft.Compute/<br>images/read  | Stack Manager will not be able to validate custom VM images provided via <b>sc_image_id</b> / <b>mc_image_id</b> Advanced Config parameters. If wrong names are provided, stack deployment will fail.   |
| Microsoft.Compute/<br>skus/read  | Stack Manager will not be able to validate instance types<br>(VM sizes) provided via <b>sc_instance_type</b> /<br><b>mc_instance_type</b> Advanced Config parameters. If wrong<br>names are provided, stack deployment will fail.   |
| Microsoft.MarketplaceOrdering/<br>offertypes/publishers/offers/<br>plans/agreements/read | Stack Manager will not be able to automatically accept<br>publisher agreement for Mediant VE/CE Marketplace offer.<br>You need to manually accept the agreement prior to new<br>stack deployment through the following CLI command:   |
| Microsoft.MarketplaceOrdering/   | az vm image terms accept \  |
| offertypes/publishers/offers/  | publisher audiocodes \  |
|  | sku mediantvesbcazure   |
|  |   |

| Permission | What happens when it's dropped   |
|------------|--|
|            | If you are deploying SBC version based on CentOS 6, use -<br>-sku mediantvirtualsbcazure in the above command. |
|            | 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/<br>networkSecurityGroups/* | You must precreate network security groups and provide them via<br>cluster_nsg_id / oam_nsg_id / signaling_nsg_id /<br>media_nsg_id Advanced Config parameters during the new<br>stack creation.                                  |
|   | Stack Manager VM must be granted with<br><b>Microsoft.Network/networkSecurityGroups/actions/join</b><br>permission in the Resource Group where network security groups<br>reside.   |
| Microsoft.Network/<br>publicIPAddresses/*     | You must precreate public IP addresses and provide them via <b>public_ip_*</b> Advanced Config parameters during the new stack creation.  |
|   | Stack Manager VM must be granted with<br><b>Microsoft.Network/publicIPAddresses/read</b> and<br><b>Microsoft.Network/publicIPAddresses/actions/join</b><br>permissions in the Resource Group where public IP addresses<br>reside. |
| Microsoft.Storage/<br>storageAccounts/*       | You must precreate diagnostics Storage Account and provide it via <b>diag_account</b> Advanced Config parameters during the new stack creation.   |
|   | Stack Manager VM must be granted with <b>Microsoft.Storage/storageAccounts/read</b> 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:

- Create an Azure Service Principal, as described in the <u>Azure documentation</u>. 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:
- In Google Cloud Platform Console, go to the Home > Dashboard (<u>https://console.cloud.google.com/home/dashboard</u>), and then determine your project ID.

| $\equiv$ Google Cloud Platform <b>*</b>   | AudioCodes Prod 👻 🔍      | - 206:  | 9      |
|---|--------------------------|---|--------|
| DASHBOARD ACTIVITY  |                          | CUST  | FOMIZE |
| <ul> <li>Project info</li> <li>Project name</li> <li>AudioCodes Prod</li> </ul> | : Dompute Engine         | : Google Cloud Platform status All services normal  | •      |
| Project ID<br>audiocodes-prod<br>Project number<br>40111111111                  |                          | Go to Cloud status dashboard<br>Billing   | :      |
| → Go to project settings  | No data is available for | or the selected time frame     Estimated charges     USD \$121.       0.2     For the billing period Nov 1 - 20, 2019 | 98     |
| Resources   | 3:45 4 PM                | 4:15 0 → View detailed charges  |        |
| Compute Engine     11 instances     Storage     4 buckets                       | -> Go to Compute Engine  | Error Reporting  No sign of any arrors. Have your set up Error  | :      |

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:

- In the Google Cloud Platform Console, go to the API & Services > Dashboard page (<u>https://console.cloud.google.com/apis/dashboard</u>).
- 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 (<u>https://console.cloud.google.com/iam-admin/serviceaccounts</u>).
- 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 (<u>https://console.cloud.google.com/iam-admin/iam</u>).
- 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:
- In the Google Cloud Platform Console, go to the Compute Engine > VM Instances page (<u>https://console.cloud.google.com/compute/instances</u>).
- 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:
- In the Google Cloud Platform Console, go to the IAM & admin > Service Accounts page (<u>https://console.cloud.google.com/iam-admin/serviceaccounts</u>).
- 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 <u>http://docs.openstack.org/openstacksdk</u> for more information.

- 3. Place the file in one of the following locations:
  - /var/stack\_mgr/.config/openstack
  - /etc/openstack

Make 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.

#### $\succ$ 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.

|                         |                     | Configuration     |                 |
|-------------------------|---------------------|-------------------|-----------------|
|                         | General             |                   | Microsoft Azure |
| Name Prefix             | alex-               | Tenant ID         |                 |
| Admin Username          | Admin               | Client ID         |                 |
| Admin Password          |                     | Secret            |                 |
|                         |                     | Subscription ID   | 4ad554cf-       |
|                         | Amazon Web Services | Blob Account Name | sbc1            |
| Access Key              |                     | Blob Account Key  | MqknD2G0PYUh    |
| Secret Key              |                     | Blob Container    | stackmgr        |
| S3 Bucket               |                     |                   |                 |
| S3 Prefix               |                     |                   | Google          |
|                         |                     | Project           |                 |
|                         | Openstack           | Credentials       |                 |
| Cloud Name              |                     | Storage Bucket    |                 |
| Container               |                     | Storage Prefix    |                 |
|                         |                     |                   |                 |
|                         |                     | Vpdate Venty      |                 |
| Verifying configuration |                     |                   |                 |

#### 2 20. Varifying Stack Ma \_.. \_ .....

# 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.



**Note:** *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 <u>http://console.aws.amazon.com/s3</u>.
- 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



 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 <u>https://portal.azure.com/</u>.
- 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.



**Note:** 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:
- In the Google Cloud Platform Console, go to the Storage > Browser page (<u>https://console.cloud.google.com/storage/browser</u>).
- 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 (<u>https://console.cloud.google.com/iam-admin/iam</u>). Assign the following IAM role to the Stack Manager service account: Storage > Storage Admin.
- 5. Log in to the Stack Manager Web interface.
- 6. Open the Configuration page.
- 7. In the 'Google Storage Bucket' field, enter the value.
- 8. 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/").
- 9. Click Update.
- **10.** 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.



**Note:** 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).
- 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.



**Note:** 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 <u>Services Portal</u>, 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

|                            | Debu       | g |
|----------------------------|------------|---|
| Debug File                 | 🛓 Download |   |
| Debug File Access<br>Level | Operator   | ~ |

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.



This page is intentionally left blank.

# 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

| Stack Manager |       |
|---------------|-------|
|               |       |
|               |       |
|               |       |
| Login         |       |
|               | Login |

- 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.



**Note:** 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<br>Administrator | Administrator | Operator     | Monitor      |
|--|---------------------------|---------------|--------------|--------------|
| Change global configuration parameters | $\checkmark$              |               |              |              |
| Manage users                           | ✓                         |               |              |              |
| Perform upgrade via Web<br>interface   | $\checkmark$              |               |              |              |
| Create new stacks                      | ✓                         | ×             |              |              |
| Manage existing stacks                 | ✓                         | ✓             | $\checkmark$ |              |
| Monitor existing stacks                | ✓                         | ✓             | $\checkmark$ | $\checkmark$ |

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



**Note:** 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

| stack_mgr  | Stacks Users C | Configuration Lo | gs Files About                    | secadmin      | Logout |
|------------|----------------|------------------|-----------------------------------|---------------|--------|
|            |                |                  |                                   |               |        |
| + Add user |                |                  |                                   | Q Search      |        |
|            |                |                  | Heeve                             |               |        |
|            |                |                  | Users                             |               |        |
| Name       | Туре           | 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 |        |
|            |                |                  |                                   |               |        |

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

| Add user                                 |                          |   |
|--|--------------------------|---|
| Name                                     | Admin                    |   |
| Password                                 | •••••                    | ۲ |
| Туре                                     | Security Administrator 💙 |   |
| Status                                   | Active 🗸                 |   |
| Password Expiration<br>In Days (0=never) | 0                        |   |
| Add 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 Y | our Own Password |
|------------------------|------------------|
|------------------------|------------------|

| C stack_mgr | Stacks | Configuration | Logs | Files | About | operator | Logout | ? |
|-------------|--------|---------------|------|-------|-------|----------|--------|---|
|             |        |               |      |       |       |          |        |   |

Configuration

| Change Password |  |
|-----------------|--|
|                 |  |
|                 |  |
|                 |  |

| New Password       |          |  |
|--------------------|----------|--|
| New Password       |          |  |
| (One More<br>Time) |          |  |
|                    |          |  |
|                    |          |  |
|                    | Security |  |
| Number Of          | 0        |  |
| Other Sessions     |          |  |
|                    |          |  |

#### **Password Complexity** 3.3.2

Current Password

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

Update

- 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

| CC stack_mgr | Stacks | Users | Configuration | Logs | Files | About | secadmin | Logout | ? |
|--------------|--------|-------|---------------|------|-------|-------|----------|--------|---|
|              |        |       |               |      |       |       |          |        |   |

| eral    |                      | Microsoft  | Azure  |  |
|---------|----------------------|--|--|--|
|         | Subscription         |  |  |  |
|         | Cloud                | Public   | ~  |  |
|         | Tenant ID            |  |  |  |
|         | Client ID            |  |  |  |
|         | Secret               |  |  |  |
| assword | Blob Account<br>Name |  |  |  |
|         | Blob Account         |  |  |  |
|         | assword              | eral<br>Subscription<br>ID<br>Cloud<br>Tenant ID<br>Client ID<br>Secret<br>assword<br>Blob Account<br>Name<br>Blob Account | eral Microsoft<br>Subscription<br>ID<br>Cloud Public<br>Cloud Public<br>Client ID<br>Client ID<br>Secret I<br>Secret I<br>Blob Account<br>Name | eral Microsoft Azure<br>Subscription<br>ID<br>Cloud Public<br>Cloud Public<br>Client ID<br>Client ID<br>Secret<br>Secret<br>Blob Account<br>Name<br>Blob Account<br>Secret |

#### Configuration

| Summary                      | Regular 💙 |  |                            | Debug         |   |
|------------------------------|-----------|--|----------------------------|---------------|---|
| Use                          | Enable 🗸  |  | Debug File                 | L Download    |   |
| onfiguration<br>Package      |           |  | Debug File<br>Access Level | Administrator | ~ |
| Docker<br>Deployment<br>Mode | Disable 🗸 |  |                            |               |   |

# 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<br>for specific user, IP address from where login was<br>performed and number if unsuccessful login attempts (if<br>there were such). |
|                       | It may also contain custom text defined via <b>Welcome Screen Custom Text</b> parameter.  |
| Welcome Screen Custom | Defines custom text to be displayed on welcome screen.  |
| Text                  | 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<br>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.<br>When hostname is defined, access to the Stack Manager's<br>Web interface is allowed only via it, and not via the<br>corresponding IP address. |

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

| Parameter | Description  |
|-----------|--|
|           | 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   | Defines default Azure Subscription ID for deployed stacks.<br>For Stack Manager versions 3.2.5 and later you can<br>choose a different Azure Subscription ID as part of stack<br>creation dialog. So this global configuration parameter only<br>defines the default subscription value.<br>For Stack Manager versions prior to 3.2.5 all stacks were<br>deployed to the Subscription ID defined via this global<br>configuration parameter.  |
| Cloud   | Defines Azure cloud where deployments are done.<br>Supported values: "Public", "US Government".   |
| Tenant ID<br>Client ID<br>Secret  | Leave these parameters blank if you are using the<br>recommended deployment method, where Stack Manager<br>VM is granted access to Azure APIs via managed system<br>identity – as described in Section 2.8.2.2, Enabling Access<br>to Azure APIs via Managed Service Identity<br>(Recommended Method).<br>However if you choose to use Service Principal instead, as<br>described in Section 2.8.2.3, Enabling Access to Azure<br>APIs via Service Principal (Alternative Method), you need<br>to configure these parameter with values that match your<br>Service Principal. |
| Blob Account Name<br>Blob Account Key<br>Blob SAS Token<br>Blob Container | Configure these parameters if you want Stack Manager to<br>store its runtime data on Microsoft Azure Storage Service.<br>See Section 2.9.2, Storing Runtime Data on Azure Storage<br>Service for detailed instructions.   |
| Application Insights<br>Connection String                                 | Configure this parameter if you want Stack Manager to<br>send metrics and alarms data to Azure Application<br>Insights.<br>See Section 7.3, Integration with Azure Application Insights<br>for more information.  |
| Application Insights Mode   | <ul> <li>Defines data that will be sent to Azure Application Insights.</li> <li>Supported values: <ul> <li>"all" – both metrics and alarms data will be sent for all stacks</li> <li>"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 /</li> </ul> </li> </ul>  |
| Parameter        | Description  |
|------------------|--|
|                  | alarms / all; refer to Section 3.8.11, Advanced<br>Configuration for detailed description.   |
| Keep-Alive Alarm | Defines whether Stack Manager periodically sends "keep-<br>alive" alarm to Azure Application Insights. The alarm may<br>be used to determine that Stack Manager application is<br>alive.<br>Supported values: "Enable", "Disable"  |
| Azure Login      | Defines whether Stack Manager's Web interface uses<br>Azure Entra ID for user authorization.<br>Supported values: "Enable", "Optional", "Disable".<br>See Section 3.6, Login via Azure Entra ID for detailed<br>description.   |
| Login Authority  | Defines login authority for authorization via Azure Entra ID.<br>See Section 3.6, Login via Azure Entra ID for detailed<br>description.  |
| INI Repository   | Defines Azure Storage Account that contains INI files<br>repository. When repository is defined, Stack Manager<br>allows choosing files from it as part of "Send INI File"<br>dialog for Mediant VE and Mediant CE stacks.<br>Syntax: "account= <account-name>;container=<container-<br>name&gt;:key=<account-key>"</account-key></container-<br></account-name> |
|                  | It is possible to specify "token" (SAS token) instead of<br>"key".   |

## 3.4.5 Amazon AWS Parameters

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

| Parameter                | Description   |
|--------------------------|---|
| Access Key<br>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<br>described in Section 2.8.1.2, Enabling Access to AWS API<br>via AWS Access Key (Alternative Method) you need to<br>configure these parameter with proper values.                         |
| S3 Bucket<br>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.<br>See Section 2.8.3.1, Configuring Google Project ID for<br>more information.   |  |  |  |
| Credentials                      | Leave these parameters blank if you are using the<br>recommended deployment method, where Stack Manager<br>VM is granted access to Googel Cloud APIs via Service<br>Account – as described in Section 2.8.3.4, Enabling<br>Access to Google Cloud APIs via Service Account<br>(Recommended Method). |  |  |  |
|                                  | However if you choose to use Configuration File instead,<br>as described in Section 2.8.3.5, Enabling Access to<br>Google Cloud APIs via Configuration File (Alternative<br>Method), you need to configure these parameter with<br>corresponding value.   |  |  |  |
| Storage Bucket<br>Storage Prefix | Configure these parameters if you want Stack Manager to<br>store its runtime data on Google Cloud Storage Service.<br>See Section 2.9.3, Storing Runtime Data on Google Cloud<br>Storage Service for detailed instructions.   |  |  |  |

# 3.4.7 Openstack Parameters

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

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

## 3.4.8 Debug File Parameters

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

| Parameter  | Description  |
|------------|--|
| Debug File | Downloads Stack Manager Debug File used for bug<br>reporting and troubleshooting.<br>See Section 2.13, Providing Debug File for<br>Troubleshooting for more information. |

| Parameter               | Description  |
|-------------------------|--|
| Debug File Access Level | Defines minimum access level that is allowed to download Debug File.               |
|                         | Supported values: "Security Administrator", "Administrator", "Operator", "Monitor" |

## 3.4.9 Advanced Parameters

The following table describes global advanced configuration parameters.

| Parameter   | Description   |  |  |  |
|---|---|--|--|--|
| Stack Manager Tag                                     | Defines global tag applied to all cloud resources (e.g.<br>virtual machines and network security groups) created<br>during stack creation.<br>Syntax: " <tag_name>=<tag_value>"<br/>You may use %IP% element in <tag_value> that will be</tag_value></tag_value></tag_name> |  |  |  |
|   | expanded to Stack Manager IP address.<br>See Section 7.2, Tagging Stack Resources for more  |  |  |  |
|   | information.  |  |  |  |
| Auto Stop Time<br>Auto Start Time<br>Auto Shelve Time | Define time for automatic stop / start / shelve operations.<br>See Section 7.1, Automatic Stop / Start / Shelve for more<br>information.  |  |  |  |
| Delete Public IPs During<br>Shelve                    | Defines whether public IPs are deleted during "shelve"<br>operation.<br>Supported valued: "Enable", "Disable"   |  |  |  |
| Stop / Start Individual<br>Components                 | Defines whether Stack Manager's "stop" and "start" operations allow user to choose specific component to be stopped / started.  |  |  |  |
|   | Supported values: "Enable", "Disable"   |  |  |  |
| Default Instance Types                                | Defines default instance types used by Stack Manager in<br>"stack create" dialog.   |  |  |  |
|   | Supported values:   |  |  |  |
|   | <ul> <li>Production – default instance type are suitable for<br/>production deployments</li> </ul>  |  |  |  |
|   | <ul> <li>Development – minimal (and typically burstable)<br/>instance types are used by defult; note that these<br/>instance types are not suitable for production<br/>deployment and software is not guaranteed to behave<br/>correctly on them</li> </ul>                 |  |  |  |
| Update Protocol                                       | Defines transport protocol used during Stack Manager software update.   |  |  |  |
|   | Supported values:   |  |  |  |
|   | <ul> <li>HTTP or HTTPS – software automatically determines<br/>transport type based on current network conditions</li> <li>HTTPS – use secure transport type only</li> </ul>  |  |  |  |
| Obfuscation Algorithm                                 | Defines algorithm used for sensitive data obfuscation in stack descriptors.   |  |  |  |

| Parameter                 | Description   |  |  |  |
|---------------------------|---|--|--|--|
|                           | <ul> <li>Supported values:</li> <li>Simple – simple transposition algorithm; this was the only algorithm supported in Stack Manager versions prior to 3.2.4</li> <li>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</li> <li>Custom Key – data is encrypted using AES256 cipher with custom key.</li> </ul> Custom key must be created using the following CLI command: stack_mgr_obfuscate-keygenerate |  |  |  |
|                           | Use the following command to view the generated key:<br>stack_mgr obfuscate-keyshow<br>And the following command to set it to specific value:<br>stack_mgr obfuscate-keyset <key></key>   |  |  |  |
| Summary Format            | <ul> <li>Defines format for stack summary list in Web interface's Stacks screen.</li> <li>Supported values:</li> <li>Regular – summary list includes regular fields</li> <li>Verbose – summary list includes Subcription ID and Comments</li> </ul>   |  |  |  |
| Use Configuration Package | Use configuration package for backup/restore of SBC configuration during "rebuild" and "update" operations on Mediant VE and Mediant CE stacks.<br>Supported values: "Enable", "Disable"  |  |  |  |
| Docker Deployment Mode    | Enables experimental Docker deployment mode for<br>Mediant VE and CE stacks on Azure.   |  |  |  |

# 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:

- 1. Configure a hostname for the Stack Manager's virtual machine.
- 2. 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 <u>https://portal.azure.com/</u>.
- 2. Navigate to Virtual Machines.
- 3. Select the Stack Manager virtual machine.
- 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.

Figure 3-6: Configuring DNS Name in Azure

7. 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".

| $\equiv$ Microsoft Azure         | Search resources, services, and docs (G+/) Opilot  |
|----------------------------------|--|
| Home > Virtual machines > stack- | mgr > stack-mgr-ip   |
| stack-mgr-ip   Co                | nfiguration * ··· ×  |
|                                  | « 🗟 Save 🗙 Discard 🖒 Refresh   |
| 🚾 Overview                       |  |
| Activity log                     | () This public IP address can't be updated because it is associated to the IP configuration 'ipconfig1',<br>in the network interface 'alex-stack-mor-nic'. |
| Access control (IAM)             |  |
| 🗳 Tags                           | IP address assignment  |
| $\checkmark$ Settings            | O Dynamic () Static  |
| a Configuration                  | IP address ①   |
| Properties                       | 20.115.155.27  |
| Locks                            | Idle timeout (minutes) ①   |
| > Monitoring                     |  |
|                                  | DNS name label (optional) 🕕  |
| > Automation                     | stack-mgr 🗸 🗸  |
| > Help                           | .westus2.cloudapp.azure.com  |
|                                  |  |
|                                  | You can use the IP address as your 'A' DNS record or DNS label as your 'CNAME' record.   |

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 (<u>https://letsencrypt.org</u>) 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 <u>https://certbot.eff.org</u>
- 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.

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
- 6. Restart the NGINX server:

sudo systemctl restart nginx

7. 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 <a href="https://chostname">https://chostname</a> 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.



**Note:** 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 <u>https://portal.azure.com/</u>.
- 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:

#### https://<hostname>/azureToken

Replace <hostname> with the hostname of the Stack Manager virtual machine.

- 8. Click **Register** to register the new application.
- 9. In the app's Overview screen, find and note the Application (client) ID and Directory (tenant) ID.

Switch to the Authentication screen and under the **Implicit grant and hybrid flows** group, enable **ID tokens**. Click **Save** to apply the changes.

- **10.** Open a new web browser tab.
- **11.** Log in to the Stack Manager Web interface.
- **12.** Open the Configuration page.
- **13.** 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

#### 14. Click Update.

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'.

- **15.** On the Azure app's registration page, switch to the App roles screen.
- **16.** 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
- 17. 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



**Note:** 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).

- **18.** 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
- **19.** 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
- 20. Navigate to the Enterprise applications page.
- **21.** 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.
- 22. Click the application.
- 23. Select Users and groups.
- 24. 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.



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

- **25.** Open a new browser tab.
- 26. Navigate to the Stack Manager's Web interface.
- 27. Click the Login with Azure button.

Enter your Azure credentials and then verify that you can successfully log in.

Figure 3-7: Login via Azure Entra ID

CC stack\_mgr

### Stack Manager

| Login with Microsoft Azure<br>account |
|---------------------------------------|
| 🗧 Login with Azure                    |

# 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:
  - 1. List existing users: \$ stack mgr user-list
  - 2. 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.



| CC st | tack_mgr          | Stacks | Configuration | Logs | About       |       |            | Logout |
|-------|-------------------|--------|---------------|------|-------------|-------|------------|--------|
|       |                   |        |               |      |             |       |            |        |
|       | + Create new stac | k      |               |      |             |       |            |        |
|       |                   |        |               |      |             |       |            |        |
|       |                   |        |               |      | Stacks      |       |            |        |
|       |                   |        |               | _    |             |       |            |        |
|       | Name              |        |               | Туре | Environment | State | IP Address |        |
|       | No stacks found   |        |               |      |             |       |            |        |

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

# Figure 3-9: Create New Stack Dialog Create new stack Name Stack type Mediant CE • Environment -- select -- • Create Cancel

- **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.



**Note:** 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 <u>https://www.audiocodes.com/library/technical-documents.</u>

## 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.
  - '1<sup>st</sup> and 2<sup>nd</sup> 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.

#### Notes:



- 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)
- 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.

| Create new stack |                                    |
|------------------|------------------------------------|
| Name             | ce1                                |
| Stack type       | Mediant CE 🔹 🗸                     |
| Environment      | AWS 🗸                              |
| Region           | EU (Frankfurt)                     |
| Key Pair         | aws_ssh_frankfurt_1                |
| IAM Role         | SBC-HA-3                           |
| Networking       |                                    |
| VPC              | vpc-45f3152c (DefaultVPC)          |
| Cluster Subnet   | subnet-0496039603680f5a2 (cluster) |

Figure 3-10: Configuring Mediant CE in AWS 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 3-11: Creating Mediant CE in AWS environment

| 🕻 stack_mgr   | Stacks Configuration Logs                                    | About       |            |            | Logout |
|---|--|-------------|------------|------------|--------|
| + Create new st   | ack  |             |            |            |        |
| Creating stack<br>Initializing AW<br>Creating netwo<br>Creating media | 'ce1'<br>S client done<br>ork resources done<br>a components |             |            |            |        |
|   |  | Stack       | S          |            |        |
| Name<br>ce1   | Type<br>Mediant CE   | Environment | State      | IP Address |        |
|   |  |             | er ea unig |            |        |

## 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 | <b>Mediant CE</b> | Stack C | Creation in | AWS Env | /ironment |
|------------|-----------------|-------------------|---------|-------------|---------|-----------|
|            |                 |                   |         |             | -       |           |

| Problem   | Reason   | Solution   |
|---|--|--|
| Mediant CE stack creation<br>freezes at the "Creating media<br>components" step for more<br>than 10 minutes. No Media | You haven't subscribed to the<br>Mediant VE offer in AWS<br>Marketplace.     | Subscribe to Mediant VE offer<br>in AWS Marketplace, as<br>described in <i>Mediant Cloud</i><br><i>Edition Installation Manual</i> .   |
| Component instances are shown in the AWS dashboard.   | The IAM role specified during<br>Mediant CE stack creation<br>doesn't exist. | Create an IAM role for Mediant<br>CE, as described in <i>Mediant</i><br><i>Cloud Edition Installation</i><br><i>Manual</i> and specify its name<br>in the Mediant CE Create<br>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.
- 1<sup>st</sup> and 2<sup>nd</sup> 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.

# **C**audiocodes

 '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.

Note: Azure imposes the following limitations on the username and password:

#### • Username:

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

#### Password:

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

| Name            | ce1          |  |
|-----------------|--------------|--|
| Stack type      | Mediant CE 🗸 |  |
| Environment     | Azure 🗸      |  |
| Region          | West US 2    |  |
| Networking      |              |  |
| Virtual Network | VnetWestUS2  |  |
| Cluster Subnet  | cluster 🗸    |  |
| Main Subnet     | oam 🗸        |  |
| 1st Additional  | voip1 🗸      |  |

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.



| 🗨 stack_mgr  | Stacks                                       | Configuration  | Logs | About |  |          |            | Logout |
|--|--|----------------|------|-------|--|----------|------------|--------|
| + Create new stack   |  |                |      |       |  |          |            |        |
| Creating stack 'ce<br>Initializing Azure of<br>Accepting market<br>Creating common | 1'<br>client dor<br>place licen<br>resources | ne<br>ise done |      |       |  |          |            |        |
| Name   | Stacks                                       |                |      |       |  |          |            |        |
| ce1  | Me   | ediant CE      |      | Azure |  | creating | IP Address |        |



**Note:** 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.

|--|

| Problem  | Reason   | Solution   |
|--|--|--|
| Mediant CE stack creation<br>fails with error message<br>"Legal terms have not<br>been accepted for this item<br>on this subscription" | You haven't subscribed to<br>the Mediant VE offer in Azure<br>Marketplace.   | Subscribe to Mediant VE offer in Azure<br>Marketplace, by deploying a demo<br>instance of it. Refer to <i>Mediant Cloud</i><br><i>Edition Installation Manual</i> for detailed<br>description.                                     |
| Mediant CE stack creation<br>fails with error message<br>"Creating resource group<br><stack_name> failed"</stack_name>                 | Stack Manager creates a<br>new Resource Group for<br>each stack with the same<br>name as the stack name<br>(unless <b>resource_group</b><br>advanced config parameter<br>is used). If your subscription<br>already has such a resource<br>group, stack creation will fail. | Use a different stack name that doesn't<br>match the name of any existing<br>Resource Group in your subscription.<br>Alternatively, you can configure the<br>'Name Prefix' parameter in the Stack<br>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.
  - 1<sup>st</sup> and 2<sup>nd</sup> 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:

<sup>6</sup> '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)
- 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

| Create new stack |                  |   |
|------------------|------------------|---|
| Name             | ce1              | * |
| Stack type       | Mediant CE 🗸     | l |
| Environment      | Google 💙         | l |
| Region           | us-central1      | ł |
| Zones            | a,b              |   |
| Image            | sbc-7-20-256-110 |   |
| Networking       |                  |   |
| Cluster Subnet   | cluster 🗸        |   |
| Main Subnet      | oam 🗸            | • |
| 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.

| Figuro 3-15   | Croating | Modiant | CE in | Googla | Cloud | Environment |
|---------------|----------|---------|-------|--------|-------|-------------|
| i igule 5-15. | oreating | Weulant |       | Google | olouu | LINNOTHIER  |

| stack_mgr   | Stacks Configuration                                       | Logs About  |          |            | Logout |  |
|---|--|-------------|----------|------------|--------|--|
| + Create new stack  |  |             |          |            |        |  |
| Creating stack 'ce<br>Initializing Googl<br>Creating common<br>Creating media c | 1'<br>e Cloud client done<br>n resources done<br>omponents |             |          |            |        |  |
| Stacks  |  |             |          |            |        |  |
| Name  | Туре   | Environment | State    | IP Address |        |  |
| ce1   | Mediant CE   | Google      | creating |            |        |  |

95

## 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.
  - '1<sup>st</sup> and 2<sup>nd</sup> 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.

| Create new stack         | :                 |  |
|--------------------------|-------------------|--|
| Name                     | ce1               |  |
| Stack type               | Mediant CE 🔹      |  |
| Environment              | OpenStack 💙       |  |
| Image                    | sbc-7-20-256-110  |  |
| Key Pair                 | admin 🗸           |  |
| Networking               |                   |  |
| Cluster Subnet           | internal_subnet 💙 |  |
| Main Subnet              | flat_subnet       |  |
| 1st Additional<br>Subnet | none 🗸            |  |
| Create Cancel            |                   |  |

Figure 3-16: Configuring Mediant CE in OpenStack 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.



| 🗙 stack_mgr  | Stacks  | Configuration     | Logs | About       |  |          |            | Logout |
|--|---|-------------------|------|-------------|--|----------|------------|--------|
| + Create new stack   |   |                   |      |             |  |          |            |        |
| Creating stack 'ce<br>Initializing Opens<br>Creating network<br>Creating media c | 1'<br>tack client.<br>resources.<br>omponent: | done<br>done<br>5 |      |             |  |          |            |        |
| Stacks   |   |                   |      |             |  |          |            |        |
| Name   | Ту  | pe                |      | Environment |  | State    | IP Address |        |
| ce1  | Me  | diant 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.
  - '1<sup>st</sup> and 2<sup>nd</sup> 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.

#### Notes:



- 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 commaseparated 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)
- or contract
- <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.

| Create new stack |                     |  |
|------------------|---------------------|--|
| Name             | ce1                 |  |
| Stack type       | Mediant VE 🗸        |  |
| Environment      | AWS 🗸               |  |
| Region           | EU (Frankfurt)      |  |
| Key Pair         | aws_ssh_frankfurt_1 |  |
| IAM Role         | SBC-HA-3            |  |
| Compute          |                     |  |
| HA Mode          | enable 🗸            |  |
| VM Type          | r4.large 🗸          |  |
| Create Cancel    |                     |  |

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.

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

| Table 3-3. | Troubloshooting | Modiant VE | Stack Croation | in AWS   | Environmont |
|------------|-----------------|------------|----------------|----------|-------------|
| Table 3-3: | Troubleshooting |            | Slack Creation | III AVV3 |             |

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.
- 1<sup>st</sup> and 2<sup>nd</sup> 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.

•

**Note:** Azure imposes the following limitations on the username and password:

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

#### Password:

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

| Create new stack |                   |   |
|------------------|-------------------|---|
| Name             | ve1               | ĺ |
| Stack type       | Mediant VE 🗸 🗸    |   |
| Environment      | Azure 🗸           |   |
| Region           | West US 2         |   |
| Compute          |                   |   |
| VM Туре          | Standard_DS1_v2 💙 |   |
| Networking       |                   |   |
| Virtual Network  | VnetWestUS2       |   |
| Main Subnet      | oam 🗸             |   |
| Create           |                   |   |

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.



**Note:** 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.

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

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

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.
- 1<sup>st</sup> and 2<sup>nd</sup> 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

| Create new stack |                  |   |
|------------------|------------------|---|
| Name             | ve1              | Î |
| Stack type       | Mediant VE 🗸     | l |
| Environment      | Google 🗸         | l |
| Region           | us-central1 🗸    | ļ |
| Image            | sbc-7-20-256-110 |   |
| Compute          |                  |   |
| HA Mode          | enable 🗸         |   |
| VM Туре          | n1-standard-2 🗸  |   |
| Networking       |                  | • |
| Create 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 VoiceAl Connect in Amazon Web Services (AWS) Environment



**Note:** VoiceAl 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.Al Connect**.
- 'Environment': AWS.
- (Region': Defines the region where VoiceAl 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 VoiceAl Connect stack is to be deployed.
  - 'Main Subnet': Defines the main subnet within the VPC to which all VoiceAl 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



Note: 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.

| Create new stack |                    |   |   |
|------------------|--------------------|---|---|
| Name             | vaic1              |   |   |
| Stack type       | Voice.Al Connect 💙 |   |   |
| Environment      | AWS 🗸              |   |   |
| Region           | EU (Frankfurt)     | ~ |   |
| Key pair         | Bahir              | ~ |   |
| Networking       |                    |   |   |
| VPC              | select             |   | ~ |
|                  |                    |   |   |
| Create Cancel    |                    |   |   |

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

Once you have configured all the above parameters, click **Create** to create the VoiceAl Connect stack instance. The operation progress is displayed at the top of the page.
## 3.8.9 Creating VoiceAl Connect in Azure Environment



**Note:** 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.Al Connect**.
- 'Environment': **Azure**.
- (Region': Defines the region where VoiceAl 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 VoiceAl 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.

# **C**audiocodes

### 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.

**Note:** Azure imposes the following limitations on the username and password:

### • Username:

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



## Password:

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

| Lreate new stack |                    |
|------------------|--------------------|
| Name             | vaic1              |
| Stack type       | Voice.Al Connect 💙 |
| Environment      | Azure 🗸            |
| Region           | West US 2          |
| Networking       |                    |
| Virtual network  | select 🗸           |

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

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 VoiceAl Connect in Google Cloud Environment



**Note:** The VoiceAl 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.Al 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 VoiceAl 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 f(ur)/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.Al 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.Al Connect in Google Cloud Environment

| Create new stack | :                  |
|------------------|--------------------|
| Name             | vaic1              |
| Stack type       | Voice.Al Connect 🗸 |
| Environment      | Google 🗸           |
| Region           | us-central1        |
| Zones            | a,b                |
| Networking       |                    |
| Main subnet      | default 🗸          |
| Public subnet    | none 💙             |
| Create Cancel    |                    |

Once you have configured all the above parameters, click **Create** to create the VoiceAl 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.



| Create new stack | <u>(</u>   |
|------------------|--|
| Min Number       | 2 🗸  |
| Max Number       | 5 🗸  |
| Admin User       |  |
| Username         | sbcadmin   |
| Password         |  |
| Advanced         |  |
| Advanced Config  | sc_public_ips = eth1.eth2<br>mc_public_ips = eth1.eth2 |
| 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<br>Environme<br>nt | Description   | Apply<br>Mode |
|---|-------------------------------|---|---------------|
| accelerated_networking  | Azure                         | Enables accelerated networking on D_v2,<br>Dds_v3 and Dds_v4 instances.<br>Note: Dds_v5 instances always have<br>accelerated networking enabled and<br>therefore, this parameter is not applicable.<br>Supported values:<br>disable (Default)<br>enable<br>Example:<br>accelerated_networking =<br>anable | Update        |
| additional3_subnet_id,<br>additional4_subnet_id,<br>additional5_subnet_id,<br>additional6_subnet_id | AWS,<br>Azure                 | Defines subnet IDs for Additional 3 to<br>Additional 6 subnets (connected to eth4 to<br>eth8, respectively).<br>For Azure, specify subnet name, e.g.,:<br>additional3_subnet_id =<br>voip3<br>For AWS, specify subnet ID, e.g.,:  | Instant       |

| Parameter   | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode |
|---|-------------------------------|--|---------------|
|   |                               | additional3_subnet_id =<br>subnet-12345<br><b>Note:</b> Subnet IDs that are currently "in use"<br>can't be modified. If you want to change the<br>subnet ID of an existing network interface,<br>first reduce the number of network<br>interfaces, update the corresponding subnet<br>ID, and then restore the number of<br>interfaces.  |               |
| additional1_subnet_cidr,<br>additional2_subnet_cidr,<br>additional3_subnet_cidr,<br>additional4_subnet_cidr,<br>additional5_subnet_cidr,<br>additional6_subnet_cidr | Azure                         | Defines the CIDR for "additional 1",<br>"additional 2", etc. subnets. This can be<br>used to overcome Stack Manager's lack of<br>permissions to read current subnet<br>configuration.<br>Syntax: same as main_subnet_cidr.<br>See also cluster_subnet_cidr<br>parameter.   | Rebuild       |
| additional_route_tables   | AWS                           | Applicable to multi-zone AWS deployments.<br>Defines additional route tables that should<br>be updated with virtual IP addresses.<br>Syntax: comma-separated list of <interface<br>name&gt;:<route id="" table="">; multiple route table<br/>IDs can be specified using pipe ( ) delimiter.<br/>Example:<br/>additional_route_tables = eth1<br/>:rtb-123,eth2:rtb-567 rtb-890</route></interface<br>   | Update        |
| app_insights  | Azure                         | <ul> <li>Defines the type of data that is reported to Azure Application Insights.</li> <li>Supported values: <ul> <li>disable: (Default) Don't report any data for this stack.</li> <li>enable: Report alarms and metrics (PMs).</li> <li>alarms: Report alarms only.</li> <li>metrics: Report metrics (PMs) only.</li> </ul> </li> <li>Example: <ul> <li>app_insights = enable</li> </ul> </li> <li>See Section 7.3, Integration with Azure Application Insights for more information.</li> </ul> | Instant       |
| auto_shelve_time  | AWS,<br>Azure,<br>Google      | <ul> <li>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).</li> <li>If defined, it overrides the global Auto Shelve Time configuration parameter.</li> <li>Supported syntax:</li> <li>08:00: Time of day (24h).</li> </ul>   | Instant       |



| Parameter           | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode |
|---------------------|-------------------------------|---|---------------|
|                     |                               | <ul> <li>1/08:00: Weekday (0 is Sunday, 1 is<br/>Monday, 2 is Tuesday, and so on) and<br/>time.</li> <li>0,1,2/08:00: Multiple weekdays and<br/>time.</li> <li>0-5/08:00: Range of weekdays and time.</li> <li>0,1/08:00 2-4/09:00: Multiple<br/>statements.</li> <li>Example:<br/>auto_shelve_time = 08:00</li> </ul>            |               |
| auto_start_time     | All                           | Defines the time of day when the stack is<br>automatically started. If defined, it overrides<br>the global <b>Auto Start Time</b> configuration<br>parameter.<br>Syntax is identical to auto_shelve_time<br>parameter.<br>Example:<br>auto_start_time = 08:00   | Instant       |
| auto_stop_time      | All                           | Defines the time of day when the stack is<br>automatically stopped. If defined, it<br>overrides the global <b>Auto Stop Time</b><br>configuration parameter.<br>Syntax is identical to auto_shelve_time<br>parameter.<br>Example:<br>auto_stop_time = 22:00   | Instant       |
| availability_zones  | Azure                         | <ul> <li>Defines availability zones where Mediant<br/>CE components are deployed.</li> <li>Syntax: <ul> <li>Two availability zone names separated<br/>by a comma.</li> <li>"none" - components are deployed in to<br/>an availability set.</li> </ul> </li> <li>Example: <ul> <li>availability_zones = 1,2</li> </ul> </li> </ul> | Update        |
| cluster_subnet_cidr | Azure                         | Defines the CIDR for the "cluster" subnet.<br>This can be used to overcome Stack<br>Manager's lack of permissions to read<br>current subnet configuration.<br>Syntax: same as main_subnet_cidr.<br>See also the<br>additionalX_subnet_cidr parameter.   | Rebuild       |
| cluster_nsg_id      | AWS, Azure                    | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.  | Update        |

| Parameter                   | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode                                   |
|-----------------------------|-------------------------------|---|---|
|                             |                               | <ul> <li>Refer to the Security Groups chapter in the<br/>Mediant CE for AWS / Azure Installation<br/>Manual for a detailed list of rules that should<br/>be included in the specific NSG.</li> <li>Syntax: <ul> <li>AWS: Security Group ID, e.g.,:</li> <li>cluster_nsg_id = sg-11223344</li> </ul> </li> <li>Azure: Resource Group name / NSG<br/>name, e.g.,:</li> <li>cluster_nsg_id =<br/>rg1/cluster-nsg</li> <li>See also the main_nsg_id,<br/>media_nsg_id, oam_nsg_id and<br/>signaling_nsg_id parameters.</li> </ul> |   |
| common_tags                 | AWS                           | <pre>Defines tags that are assigned to created<br/>network security groups.<br/>Syntax: comma-separated list of<br/>name=value pairs.<br/>Example:</pre>  | Rebuild   |
| diag_account                | Azure                         | Defines the name of the existing Storage<br>Account.<br>If not empty, the specified Storage Account<br>stores the VM's diagnostics data instead of<br>creating a new one.<br>Syntax: Resource Group name / account<br>name<br>Example:<br>diag_account = rg1/account1   | Can't be<br>modified<br>after stack<br>creation |
| disk_encryption_set         | Azure                         | Defines the Disk Encryption Set to<br>implement server-side encryption with<br>customer-managed key (SSE-CMK) for<br>managed disks.<br>Syntax: Resource Group name / Disk<br>Encryption Set name<br>Example:<br>disk_encryption_set = rg1/des1  | Rebuild   |
| eip_depends_on_instanc<br>e | AWS                           | Defines the dependency relation in Cloud<br>Formation scripts used by Stack Manager to<br>create / update stack resources.<br>Prior to Version 3.1.0, Stack Manager<br>defaulted to <b>disable</b> . In the middle of 2023,<br>AWS implemented a change in Cloud<br>Formation behavior that started causing<br>"instance is not in a valid state" error during  | Instant   |



| Parameter                               | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode                                   |
|---|-------------------------------|--|---|
|   |                               | <ul> <li>stack creation / update. To overcome this problem, this parameter was introduced and the default was changed to enable for newly created stacks.</li> <li>Supported values: <ul> <li>disable: EC2 instances depend on Elastic IPs.</li> <li>enable: (Default) Elastic IPs depend on EC2 instances.</li> </ul> </li> <li>Example: <ul> <li>eip_depends_on_instance = enable</li> </ul> </li> </ul>   |   |
| ha_nlb                                  | AWS                           | <ul> <li>Enables the use of AWS Network Load<br/>Balancer for 1+1 HA implementation of<br/>Signaling Components.</li> <li>Supported values: <ul> <li>internal – use internal NLB instead of<br/>Virtual IPs</li> <li>public – use public NLB instead of<br/>Elastic IPs</li> <li>all – use internal / public NLB instead of<br/>Virtual / Elastic IPs</li> </ul> </li> <li>Example:<br/>ha_nlb = internal</li> </ul>   | Can't be<br>modified<br>after stack<br>creation |
| ini_incremental                         | All                           | Defines additional configuration parameters<br>(in INI file format) for Signaling Components<br>during stack creation / rebuild.<br>Syntax: a single line with \n as line delimiter,<br>e.g.,:<br>ini_incremental = EnableSyslog<br>= 1\nSyslogServerIP = 10.1.2.3<br>Specified configuration is applied via REST<br>API and therefore, has no size limit.<br>Therefore, this parameter is preferred over<br>the sc_ini_params parameter that is<br>applied via cloud-init mechanism and is<br>therefore, limited by instance user-data<br>size. | Rebuild   |
| imds                                    | AWS                           | <ul> <li>Defines the version of the AWS meta-data instance for the deployed EC2 instances.</li> <li>Supported values:</li> <li>any: Allow both IMDSv1 and IMDSv2.</li> <li>v2: (Default) Enforce IMDSv2.</li> <li>Example:</li> <li>imds = v2</li> </ul>   | Update  |
| ipv6_ip_sc-X_eth*,<br>ipv6_ip_mc-X_eth* | AWS                           | Defines pre-defined IPv6 addresses.<br>See also the private_ip_* and<br>public ip * parameters.  | Update  |

| Parameter        | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode |
|------------------|-------------------------------|---|---------------|
| ipv6_virtual_ips | AWS                           | <pre>Defines whether "virtual IPs" are used for<br/>IPv6 addresses in multi-zone AWS<br/>deployments.<br/>"Virtual IPs" apply only to internal traffic<br/>(within the subnet or via Transit Gateway)<br/>and therefore, have very limited application.<br/>Correspondingly, this parameter defaults to<br/>"disable" and it's recommended to use AWS<br/>Load Balancer or DNS-based methods for<br/>traffic distribution.<br/>Supported values:<br/>• enable: Use "virtual IP" addresses for<br/>IPv6 addresses.<br/>• disable: (default) Don't use "virtual IP"<br/>addresses for IPv6 addresses.<br/>Example:<br/>ipv6_virtual_ips = enable<br/>When enabled, the virtual IP address can<br/>be specified via the virtual IP address can<br/>be specified via the virtual_ip_sc_eth*<br/>parameter, for example:<br/>virtual_ip_sc_eth2 =<br/>fd00::a0dc:1</pre> | Update        |
| kms_key_id       | AWS                           | <pre>Identifier of the AWS KMS key for Amazon<br/>EBS disk encryption. You can specify the<br/>key via one of the following:     Key ID     Key alias     Key ARN     Alias ARN Example:     kms-key-id = arn:aws:kms:us-<br/>east-1: 012345678910:1234abcd-<br/>12ab6ef-1234567890ab</pre>   | Rebuild       |
| main_subnet_cidr | Azure                         | Defines the CIDR for the "main" subnet.<br>This can be used to overcome Stack<br>Manager's lack of permissions to read<br>current subnet configuration.<br>Syntax: Subnet IP / Prefix Length.<br>Example:<br>mian_subnet_cidr = 10.2.3.0/24   | Rebuild       |
| main_nsg_id      | Azure                         | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the Security Groups chapter in the<br>Mediant CE for AWS / Azure Installation<br>Manual for a detailed list of rules that should<br>be included in the specific NSG.   | Update        |



| Parameter         | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode |
|-------------------|-------------------------------|---|---------------|
|                   |                               | Syntax is similar to the cluster_nsg_id<br>parameter.<br>When modifying the security group, make<br>sure that it includes rules that enable Stack<br>Manager to access deployed instances via<br>the HTTPS protocol (TCP/443).<br>See also the media nsg_id com nsg_id  |               |
|                   |                               | and<br>signaling_nsg_id parameters.   |               |
| manage_via_https  | All                           | Defines the protocol used by Stack<br>Manager when connecting to the deployed<br>stack's management interface.  | Instant       |
|                   |                               | <ul> <li>Supported values:</li> <li>enable: (Default) Use HTTPS.</li> <li>disable: Use HTTP.</li> <li>Example:</li> <li>manage via https = disable</li> </ul>   |               |
| media_nsg_id      | AWS,<br>Azure                 | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the Security Groups chapter in the<br>Mediant CE for AWS / Azure Installation<br>Manual for a detailed list of rules that should<br>be included in the specific NSG.<br>Syntax is similar to the cluster_nsg_id<br>parameter.<br>See also the main_nsg_id, oam_nsg_id<br>and<br>signaling_nsg_id parameters. | Update        |
| mc_additional_ips | All                           | Defines the network interface names of<br>Media Components for which additional<br>private IP addresses are allocated and<br>optionally, the number of corresponding IP<br>addresses.<br>Refer to the sc_additional_ips<br>parameter for more information.<br>After stack creation, use the <b>MC Additional</b><br><b>IPs</b> parameter in the <b>Modify</b> dialog to<br>change the parameter value.  | Update        |
| mc_image_id       | AWS,<br>Azure                 | Defines the local image for Media<br>Components (instead of the Marketplace<br>image).<br>Refer to the sc_image_id parameter for<br>more information.<br>After stack creation, use the <b>MC Image ID</b><br>parameter in the <b>Modify</b> dialog to change<br>the parameter value.  | Update        |

| Parameter        | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode                         |
|------------------|-------------------------------|---|---------------------------------------|
|                  |                               | See also the sbc_image_id parameter.  |                                       |
| mc_image_url     | AWS,<br>Azure,<br>Google      | Defines the URL that contains a plain-text<br>file with the name of the local image for<br>Media Components (instead of the<br>Marketplace image).<br>Refer to the sc_image_url parameter for<br>more information.<br>See also the sbc_image_url parameter.   | Rebuild                               |
| mc_ini_params    | All                           | Defines additional configuration parameters<br>(in INI file format) for Media Components<br>during stack creation / rebuild.<br>Refer to the sc_ini_params parameter for<br>more information.   | Rebuild                               |
| mc_ipv6_ips      | AWS,<br>Azure                 | Defines the network interface names of<br>Media Components for which IPv6<br>addresses are allocated.<br>Refer to the sc_ipv6_ips parameter for<br>more information.  | Update                                |
| mc_max_pps_limit | All                           | <ul> <li>Defines the Media Component's maximum forwarding capacity (in packets per second).</li> <li>Supported values:</li> <li>auto: (Default) Stack Manager automatically configures Media Component forwarding capacity based on the cloud environment and instance type used</li> <li><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</number></li> <li>Example:</li> </ul> | Rebuild                               |
| mc_public_ips    | All                           | Defines the network interfaces of Media<br>Components for which public IP addresses<br>are allocated and optionally, the number of<br>corresponding IP addresses.<br>Refer to the sc_public_ips parameter for<br>more information.  | Update                                |
| mc_tags          | AWS,<br>Azure,<br>Google      | <ul> <li>Defines tags that are assigned to the following Media Components' resources:</li> <li>AWS: Tags are assigned to EC2 instance, volume, network interfaces and Elastic IPs.</li> </ul>   | Azure:<br>Update<br>Other:<br>Rebuild |



| Parameter                           | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode |
|-------------------------------------|-------------------------------|---|---------------|
|                                     |                               | <ul> <li>Azure and Google Cloud: Tags are<br/>assigned to VM instances.</li> <li>Refer to the sc_tags parameter for more<br/>information.</li> <li>See also the tags and common_tags<br/>parameters.</li> </ul>   |               |
| mc_user_data                        | All                           | Defines additional cloud-init configuration<br>parameters for Media Components.<br>Refer to the sc_user_data parameter for<br>more information.   | Rebuild       |
| nat_ip_sc_ethX,<br>nat_ip_mc-X_ethY | AWS,<br>Azure,<br>Google      | <ul> <li>Defines custom entries in the NAT<br/>Translation table of Signaling Components<br/>and Media Components.</li> <li>The parameter is useful when Public IP<br/>addresses are provided by external firewall /<br/>NAT gateway, and not attached directly to<br/>resources deployed by Stack Manager.</li> <li>Syntax: comma-separated list of IP<br/>addresses</li> <li>Example:</li> <li>nat_ip_sc_eth1 = 10.1.1.5<br/>nat_ip_mc-1_eth1 = 10.1.1.10<br/>nat_ip_mc-2_eth1 = 10.1.1.11</li> <li>Note:</li> <li>If a single IP address is specified, it's<br/>configured:</li> <li>Azure, Google Cloud and multi-<br/>zone AWS deployments: For<br/>primary address, e.g., "eth1".</li> <li>Single-zone AWS deployments:<br/>For first "usable" address, e.g.,<br/>"eth1:1" on Signaling Components<br/>and "eth1" on Media Components.</li> <li>If multiple IP addresses are specified,<br/>they are configured for additional /<br/>secondary addresses. You can leave<br/>irrelevant list elements empty, e.g., the<br/>following:</li> <li>nat_ip_mc-1_eth2 =<br/>,10.1.1.10,10.1.1.11</li> <li>configures NAT translation for mc-1<br/>on "eth2:1" and "eth2:2".</li> </ul> | Update        |
| nsg_id_mc_ethX                      | AWS,<br>Azure                 | Defines the name of the existing Network<br>Security Group (NSG) for a specific Media<br>Components' network interface.   | Update        |
|                                     |                               | Mediant CE for AWS / Azure Installation   |               |

| Parameter      | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode  |
|----------------|-------------------------------|---|--|
|                |                               | Manual for a detailed list of rules that should<br>be included in the specific NSG.<br>Syntax is similar to the nsg_id_sc_ethX<br>parameter.  |  |
| nsg_id_sc_ethX | AWS,<br>Azure                 | <ul> <li>Defines the name of the existing Network<br/>Security Group (NSG) for a specific<br/>Signaling Components' network interface.</li> <li>Refer to the Security Groups chapter in the<br/>Mediant CE for AWS / Azure Installation<br/>Manual for a detailed list of rules that should<br/>be included in the specific NSG.</li> <li>When modifying the security group that<br/>contains "management" rules, make sure<br/>that it includes rules that enable Stack<br/>Manager to access deployed instances via<br/>the HTTPS protocol (TCP/443).</li> <li>Syntax:</li> <li>AWS: Security Group ID. Multiple<br/>groups can be specified as a comma-<br/>separated string, e.g.,:<br/>nsg_id_sc_eth1 = sg-123, sg-345</li> <li>Azure: Resource Group name / NSG<br/>name, e.g.,:<br/>nsg_id_mc_eth2 = rg1/cluster-<br/>nsg</li> </ul> | Update   |
| oam_nsg_id     | AWS                           | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the Security Groups chapter in the<br>Mediant CE for AWS / Azure Installation<br>Manual for a detailed list of rules that should<br>be included in the specific NSG.<br>Syntax is similar to the cluster_nsg_id<br>parameter.<br>When modifying the security group, make<br>sure that it includes rules that enable Stack<br>Manager to access deployed instances via<br>the HTTPS protocol (TCP/443).<br>See also the main_nsg_id,<br>media_nsg_id and signaling_nsg_id<br>parameters.  | Update   |
| oam_ip         | AWS,<br>Azure,<br>Google      | <ul> <li>Defines an IP address on Signaling<br/>Components for management traffic (Web,<br/>SSH, SNMP).</li> <li>Syntax:</li> <li>"default": Use primary IP address on the<br/>"main" network interface for<br/>management traffic. In Azure and</li> </ul>   | Update<br>(Azure)<br>or<br>Rebuild<br>(AWS,<br>Google) |



| Parameter                              | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode                                   |
|--|-------------------------------|---|---|
|  |                               | <ul> <li>Google Cloud, Signaling Components reside behind Load Balancer and management traffic should be sent to the frontend addresses on the corresponding Load Balancer.</li> <li>"internal": Implies the following configuration:</li> <li>SC Public IPs: main</li> <li>SC Additional IPs: main oam_ip = internal</li> <li>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.</li> <li>AWS: Creates an additional private IP address on the "main" network interface.</li> <li>"additional1" / "additional2": (Google Cloud only) Use the primary IP address on the address on the "main" network interface connected to the "Additional 1" / "Additional 2" subnet correspondingly for management traffic.</li> </ul> |   |
| proximity_placement_gro<br>up_vm_sizes | Azure                         | Defines the optional "intent" parameter of<br>proximity placement groups created as part<br>of stack deployment. Refer to<br>https://learn.microsoft.com/en-<br>us/azure/virtual-machines/co-location for<br>details.<br>Syntax: comma-separated list of VM sizes.<br>Examples:<br>proximity_placement_group_vm_siz<br>es =<br>Standard_D2ds_v5, Standard_D4ds_v<br>5<br>Typically used together with the<br>proximity_placement_group_vm_zon<br>e parameter (see below).   | Can't be<br>modified<br>after stack<br>creation |
| proximity_placement_gro<br>up_vm_zone  | Azure                         | Defines the optional "zone" parameter of<br>proximity placement groups created as part<br>of stack deployment. Refer to<br>https://learn.microsoft.com/en-  | Can't be<br>modified<br>after stack<br>creation |

| Parameter  | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode |
|--|-------------------------------|---|---------------|
|  |                               | <pre>us/azure/virtual-machines/co-location for<br/>details.<br/>Enables deployment of Mediant CE into an<br/>availability set located in the specific<br/>availability zone.<br/>Examples:<br/>proximity_placement_group_vm_zon<br/>e = 1<br/>Due to limitations of Azure API, if you<br/>specify this parameter you must also<br/>specify the<br/>proximity_placement_group_vm_siz<br/>es parameter.</pre> |               |
| private_ip_sc-X_eth*,<br>private_ip_mc-X_eth*                    | AWS,<br>Azure,<br>Google      | Defines pre-defined private IP addresses.<br>See Section 3.25.3 for more information.<br>See also the ipv6_ip_* and<br>public_ip_* parameters.  | Update        |
| public_ip_sc_eth*,<br>public_ip_mc-X_eth*                        | AWS,<br>Azure,<br>Google      | Defines pre-defined public addresses. See<br>Section 3.25.3 for more information.<br>See also the ipv6_ip_* and<br>private_ip_* parameters.   | Update        |
| public_ip_prefix_sc,<br>public_ip_prefix_mc,<br>public_ip_prefix | Azure                         | <pre>Defines a Public IP Prefix from which public<br/>IP addresses for Signaling Components or<br/>Media Components are allocated.<br/>Syntax: comma-separated list of elements:</pre>  | Update        |
| remote_interfaces  | All                           | When set to "refresh", it triggers a refresh of<br>the Remote Media Interfaces table upon the<br>next "update" action.<br>Example:<br>remote_interfaces = refresh   | Update        |



| Parameter         | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode                                   |
|-------------------|-------------------------------|---|---|
| resource_group    | Azure                         | Defines the name of the existing Resource<br>Group.<br>If not empty, stack resources are deployed<br>into this Resource Group instead of creating<br>a new one. The Resource Group must be<br>empty prior to stack creation.<br>Example:<br>resource_group = SbcGroup1  | Can't be<br>modified<br>after stack<br>creation |
| sbc_image_id      | AWS,<br>Azure                 | Defines the local image for Signaling<br>Components and Media Components<br>(instead of the Marketplace image).<br>Contrary to the sc_image_id and<br>mc_image_id parameters, this parameter<br>is applied via the <b>Rebuild</b> (and not <b>Update</b> )<br>operation.<br>For Azure, specify the resource group name<br>followed by the image name, e.g.,:<br>sc_image_id = rg1/image1<br>For AWS, specify the AMI ID, e.g.,:<br>sc_image_id = ami-9a50cff5   | Rebuild   |
| sbc_image_url     | AWS,<br>Azure,<br>Google      | Defines the URL that contains a plain-text<br>file with the name of the local image for<br>Signaling Components and Media<br>Components (instead of the Marketplace<br>image).<br>Example:<br>sc_image_url = https://<br>company.com/123456<br>See also the mc_image_url and<br>sc image url parameter.   | Rebuild   |
| sc_additional_ips | All                           | Defines the network interface names of<br>Signaling Components for which additional<br>private IP addresses are allocated and<br>optionally, the number of corresponding IP<br>addresses.<br>Additional IP addresses are allocated <i>on top</i><br>of any private IP addresses created by<br>Stack Manager by default and/or due to the<br>public IP addresses assigned to the specific<br>network interface.<br>Syntax: comma-separated list of subnet<br>names: "main", "additional1", "additional2",<br>etc.; e.g.,:<br>sc_additional_ips =<br>additional1, additional2<br>You can also use "all" to specify all subnets.<br>If more than one additional private IP<br>address is required on the specific network<br>interface, this can be specified as | Update  |

| Parameter   | Applicable<br>Environme<br>nt | Description   | Apply<br>Mode     |
|-------------|-------------------------------|---|-------------------|
|             |                               | <pre>"<name>:<num>", where <num> is the total<br/>number of additional private IP addresses to<br/>be created; e.g.,:<br/>sc_additional_ips =</num></num></name></pre>                            |                   |
|             |                               | additional1:2<br>Alternatively, you can specify interface<br>names "ethX" instead of subnet names,<br>e.g.,:  |                   |
|             |                               | <pre>sc_additional_ips = eth1,eth2</pre>  |                   |
|             |                               | Notes:  |                   |
|             |                               | <ul> <li>On Azure, for Signaling Components,<br/>internal IP addresses are allocated on<br/>Internal Load Balancer, and<br/>corresponding network interfaces are<br/>placed behind it.</li> </ul> |                   |
|             |                               | <ul> <li>On Google Cloud, for Signaling<br/>Components, internal IP addresses are<br/>allocated on Network Load Balancer,<br/>and the "main" interface is placed behind<br/>it.</li> </ul>        |                   |
|             |                               | After stack creation, use the <b>SC Additional</b><br><b>IPs</b> parameter in the <b>Modify</b> dialog to<br>change the parameter value.  |                   |
|             |                               | parameter.  |                   |
| sc_ha_mode  | All                           | Defines the number of Signaling<br>Components.  | Azure:<br>Update  |
|             |                               | Supported values:   |                   |
|             |                               | <ul> <li>enable: (Default) Two Signaling<br/>Components are created and operate in<br/>1+1 HA mode.</li> </ul>  | Other:<br>Rebuild |
|             |                               | <ul> <li>disable: One Signaling Component is<br/>created.</li> </ul>  |                   |
|             |                               | Example:<br>sc_ha_mode = disable  |                   |
| sc_image_id | AWS,<br>Azure                 | Defines the local image for Signaling<br>Components (instead of the Marketplace<br>image).  | Update            |
|             |                               | For Azure, specify the resource group name followed by the image name, e.g.,:   |                   |
|             |                               | <pre>sc_image_id = rgl/imagel</pre>   |                   |
|             |                               | For AVVS, specify the AMI ID, e.g.,:  |                   |
|             |                               | After stack creation use the SC Image ID  |                   |
|             |                               | parameter in the <b>Modify</b> dialog to change<br>the parameter value.   |                   |



| Parameter     | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode |
|---------------|-------------------------------|--|---------------|
|               |                               | See also the sbc_image_id and mc_image_id parameters.  |               |
| sc_image_url  | AWS,<br>Azure,<br>Google      | Defines the URL that contains a plain-text<br>file with the name of the local image for<br>Signaling Components (instead of the<br>Marketplace image).<br>Example:<br>sc_image_url = https://<br>company.com/123456<br>See also the mc_image_url and<br>sbc_image_url parameter.   | Rebuild       |
| sc_ini_params | All                           | Defines additional configuration parameters<br>(in INI file format) for Signaling Components<br>during stack creation / rebuild.<br>Syntax: a single line with \n as line delimiter,<br>e.g.,:<br>ini_incremental = EnableSyslog<br>= 1\nSyslogServerIP = 10.1.2.3<br>Specified configuration is applied via cloud-<br>init mechanism and therefore, is limited by<br>instance user-data size (for example, in<br>AWS it's limited by 16 KB). For long<br>configurations, use the ini_incremental<br>parameter instead.<br>See also the mc_ini_params parameter.   | Rebuild       |
| sc_ipv6_ips   | AWS,<br>Azure                 | Defines the network interface names of<br>Signaling Components for which IPv6<br>addresses are allocated.<br>For AWS, IPv6 addresses are a <i>new</i> type of<br>addresses that are globally routable. They<br>are assigned <i>in addition</i> to IPv4 addresses<br>specified via the sc_public_ips and<br>sc_additional_ips parameters. If the<br>interface is connected to a subnet that has<br>both IPv4 and IPv6 address ranges, the<br>primary address will be IPv4, and IPv6 will<br>be assigned as a secondary address with<br>the 'ethX:100' name. If the interface is<br>connected to IPv6-only subnet (i.e., subnet<br>that lacks IPv4 address range), only IPv6<br>addresses are allocated and assigned with<br>regular 'ethX' names. Pre-defined IPv6<br>addresses can be specified via the<br>ipv6_ip_* parameters.<br><b>For Azure</b> , IPv6 is a <i>property</i> of normal<br>internal and public addresses. Therefore,<br>this parameter does not add new<br>addresses, but changes the type of those<br>that are specified via the sc_public_ips<br>or sc_additional_ips parameters. | Update        |

| Parameter     | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode |
|---------------|-------------------------------|--|---------------|
|               |                               | Primary addresses can't be IPv6 and<br>therefore, this parameter by default<br>configures IPv6 on the "1st secondary"<br>(second) IP address. For example, the<br>following creates two public addresses –<br>IPv4 and IPv6 – on Signaling Components'<br>interfaces connected to "additional 1"<br>subnet:<br>SC Public IPs: additional1:2<br>sc_ipv6_ips = additional1 |               |
|               |                               | You can specify secondary address index<br>(1="1st secondary", 2="2nd secondary", and<br>so on) to configure IPv6 on a different<br>secondary address. For example, the<br>following configures IPv6 on the "2nd<br>secondary" (third) IP address of Signaling<br>Components' interfaces connected to<br>"additional 1" subnet:  |               |
|               |                               | SC Public IPs: additional1<br>SC Additional IPs:<br>additional1:2  |               |
|               |                               | sc ipv6 ips = additional1:2  |               |
|               |                               | Pre-defined IPv6 addresses can be specified via the private_ip_* and public ip * parameters.   |               |
|               |                               | Syntax: comma-separated list of subnet<br>names: "main", "additional1", "additional2",<br>etc., for example:   |               |
|               |                               | <pre>sc_ipv6_ips = additional1,additional2</pre>   |               |
|               |                               | Alternatively, you can specify interface<br>names "ethX" instead of subnet names,<br>e.g.,:  |               |
|               |                               | <pre>sc_ipv6_ips = eth1,eth2</pre>   |               |
|               |                               | Notes:   |               |
|               |                               | <ul> <li>Management is always done via IPv4<br/>addresses.</li> </ul>  |               |
|               |                               | <ul> <li>For multi-zone deployment in AWS,<br/>"virtual IPs" are disabled by default for<br/>IPv6 addresses because they work only<br/>within the subnet and therefore, have<br/>very limited application. Use the<br/>ipv6_virtual_ips parameter to<br/>change this behavior.</li> </ul>  |               |
| sc_public_ips | All                           | Defines the network interfaces of Signaling<br>Components for which public IP addresses<br>are allocated and optionally, the number of<br>corresponding IP addresses.<br>During stack creation (via Web interface).  | Update        |
|               |                               | Stack Manager lets you specify which   |               |



| Parameter | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode |
|-----------|-------------------------------|--|---------------|
|           |                               | <pre>subnets (and corresponding network<br/>interfaces) are assigned with public (Elastic)<br/>IP addresses using the Public IPs<br/>parameter in the Networking section.<br/>When the sc_public_ips and/or<br/>mc_public_ips advanced configuration<br/>parameters are specified, they override any<br/>value configured by the Public IPs<br/>parameter. Typically, you use these<br/>parameters when:<br/>• You need to create multiple IP<br/>addresses on the same network<br/>interface.<br/>• You need to configure IP addresses<br/>differently for Signaling Components and<br/>Media Components.</pre><br>Syntax: comma-separated list of subnet<br>names: "main", "additional1", "additional2",<br>etc.; e.g.,:<br>sc_public_ips =<br>additional1, additional2<br>You can also use "all" to specify all subnets.<br>If more than one public IP address is<br>required on the specific network interface,<br>this can be specified as " <name>:<num>",<br/>where <num> is the total number of public<br/>IP addresses to be created; e.g.,:<br/>sc_public_ips =<br/>additional1:2<br/>Alternatively, you can specify interface<br/>names "ethX" instead of subnet names,<br/>e.g.,:<br/>sc_public_ips = eth1,eth2</num></num></name> |               |
|           |                               | <ul> <li>Notes:</li> <li>On Azure, for Signaling Components, public IP addresses are allocated on Public Load Balancer, and corresponding network interfaces are placed behind it.</li> <li>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.</li> <li>Stack Manager implicitly creates all private IP addresses required for public IP address assignment.</li> </ul>   |               |

| Parameter                   | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode                         |
|-----------------------------|-------------------------------|--|---------------------------------------|
|                             |                               | After stack creation, use the SC Public IPs<br>parameter in the Modify dialog to change<br>the parameter value.<br>See also the mc_public_ips parameter.   |                                       |
| sc_tags                     | AWS,<br>Azure,<br>Google      | <ul> <li>Defines tags assigned to the following<br/>Signaling Components' resources:</li> <li>AWS: Tags are assigned to EC2<br/>instance, volume, network interfaces and<br/>Elastic IPs.</li> <li>Azure and Google Cloud: Tags are<br/>assigned to VM instances.</li> <li>Syntax:</li> <li>AWS or Azure: comma-separated list of<br/>name=value pairs, e.g.,:<br/>sc_tags = type=sbc, role=sc</li> <li>Google: comma-separated list of tags,<br/>e.g.,:<br/>sc_tags = sbc, sc</li> <li>See also the tags, mc_tags, and<br/>common_tags parameters.</li> </ul> | Azure:<br>Update<br>Other:<br>Rebuild |
| sc_user_data                | All                           | Defines additional cloud-init configuration<br>parameters for Signaling Components.<br>Syntax: a single line with \n as line delimiter.<br>Example:<br>sc_user_data = #customer-<br>id\n123456\n#license-<br>key\nokRTr5top  | Rebuild                               |
| sc1_ha_name,<br>sc2_ha_name | All                           | Defines the name of the first / second<br>Signaling Component in the SBC's Web<br>interface's Monitor page.<br>Example:<br>$sc1_ha_name = sc-1$<br>$sc2_ha_name = sc-2$  | Rebuild                               |
| shelve_delete_ips           | AWS,<br>Azure                 | <ul> <li>Defines whether public IP addresses are deleted during "shelve" operation. It overrides the global "Delete Public IPs During Shelve" configuration parameter.</li> <li>Supported values: <ul> <li>enable: Delete public IPs during "shelve" operation.</li> <li>disable: Don't delete public IPs during "shelve" operation.</li> <li>empty: (Default) Use global configuration parameter.</li> </ul> </li> <li>Example: <ul> <li>shelve delete ips = enable</li> </ul> </li> </ul>  | Instant                               |

| Parameter            | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode                                   |
|----------------------|-------------------------------|--|---|
| signaling_nsg_id     | AWS,<br>Azure                 | <ul> <li>Defines the name of the existing Network<br/>Security Group (NSG) to be used instead of<br/>default security groups created by Stack<br/>Manager.</li> <li>Refer to the Security Groups chapter in the<br/>Mediant CE for AWS / Azure Installation<br/>Manual for a detailed list of rules that should<br/>be included in the specific NSG.</li> <li>Syntax is similar to cluster_nsg_id<br/>parameter.</li> <li>See also the main_nsg_id,<br/>media_nsg_id and oam_nsg_id<br/>parameters.</li> </ul> | Update  |
| spot_instances       | Azure                         | <ul> <li>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.</li> <li>Supported values: <ul> <li>enable: Use Spot instances.</li> <li>disable: (Default) Use regular instances.</li> </ul> </li> <li>Example: <ul> <li>spot_instances = enable</li> </ul> </li> </ul>  | Can't be<br>modified<br>after stack<br>creation |
| storage_account_type | Azure                         | Defines the storage account type for<br>managed disks.<br>Valid values include:<br>• Standard_LRS<br>• Premium_LRS<br>• StandardSSD_LRS<br>Example:<br>storage_account_type =<br>Premium_LRS   | Rebuild   |
| tags                 | Azure                         | Defines tags assigned to all created stack<br>resources.<br>Syntax: comma-separated list of<br>name=value pairs.<br>Example:<br>tags = type=sbc,role=sc<br>You can also use the sc_tags and<br>mc_tags parameters to define additional<br>tags that will be assigned to respective VMs<br>only.  | Azure:<br>Update<br>Other:<br>Rebuild           |
| update_needed        | All                           | When set to "reset", it turns off the<br>update_needed flag without applying any<br>changes upon the next "update" action.<br>Example:<br>update_needed = reset  | Update  |

| Parameter                         | Applicable<br>Environme<br>nt | Description  | Apply<br>Mode                                   |
|-----------------------------------|-------------------------------|--|---|
| use_availability_set              | Azure                         | <ul> <li>Defines whether an availability set is created for 'availability set' deployment topology.</li> <li>Supported values: <ul> <li>enable: (Default) Create availability set.</li> <li>disable: Don't create availability set. Virtual machines are deployed without any redundancy constraints.</li> </ul> </li> <li>Example: <ul> <li>use_availability_set = disable</li> </ul> </li> </ul> | Can't be<br>modified<br>after stack<br>creation |
| use_placement_group               | AWS                           | <ul> <li>Defines if Mediant CE components are deployed in the placement group.</li> <li>Supported values: <ul> <li>enable: (Default) Use placement group.</li> <li>disable: Don't use placement group.</li> </ul> </li> <li>Example: <ul> <li>use_placement_group = disable</li> </ul> </li> </ul>   | Can't be<br>modified<br>after stack<br>creation |
| use_proximity_<br>placement_group | Azure                         | <ul> <li>Defines if Mediant CE components are deployed in the proximity placement group.</li> <li>Supported values: <ul> <li>enable: (Default) Use proximity placement group.</li> <li>disable: Don't use proximity placement group.</li> </ul> </li> <li>Example: <ul> <li>use_proximity_placement_group = disable</li> </ul> </li> </ul>   | Can't be<br>modified<br>after stack<br>creation |
| virtual_ip_sc_eth*                | AWS                           | Applicable to multi-zone AWS deployments.<br>Defines virtual IP addresses for<br>communication in the VPC or via the Transit<br>Gateway.<br>Example:<br>virtual_ip_sc_eth2 = 10.5.1.10   | Update  |
| virtual_ips                       | AWS                           | Defines whether "virtual IPs" are used for<br>IPv4 addresses in multi-zone AWS<br>deployments.<br>Supported values:<br>• enable (default)<br>• disable<br>Example:<br>virtual_ips = disable  | Update  |
| volume_type                       | AWS                           | Defines the volume type for EBS disks.<br>Valid values include:<br>gp2   | Rebuild   |



| Parameter | Applicable<br>Environme<br>nt | Description                  | Apply<br>Mode |
|-----------|-------------------------------|------------------------------|---------------|
|           |                               | • gp3                        |               |
|           |                               | <ul> <li>io1</li> </ul>      |               |
|           |                               | <ul> <li>io2</li> </ul>      |               |
|           |                               | • sc1                        |               |
|           |                               | • sc2                        |               |
|           |                               | <ul> <li>standard</li> </ul> |               |
|           |                               | Example:                     |               |
|           |                               | <pre>volume_type = gp3</pre> |               |

## 3.8.11.2 Advanced Configuration for Mediant VE

The following table describes advanced parameters available for Mediant VE.

| Parameter                  | Applicable<br>Environment | Description   | Apply<br>Mode |
|----------------------------|---------------------------|---|---------------|
| accelerated_netwo<br>rking | Azure                     | Enables accelerated networking on D_v2,<br>Dds_v3 and Dds_v4 instances.<br>Note that Dds_v5 instances always have<br>accelerated networking enabled and therefore,<br>this parameter is not applicable.<br>Supported values:<br>• disable (Default)<br>• enable<br>Example:<br>accelerated_networking = enable  | Update        |
| additional_ips             | AI                        | <pre>Defines the network interface names for which<br/>additional private IP addresses are allocated and<br/>optionally, the number of corresponding IP<br/>addresses.<br/>Additional IP addresses are allocated on top of<br/>any private IP addresses created by Stack<br/>Manager by default and/or due to the public IP<br/>addresses assigned to the specific network<br/>interface.<br/>Syntax: comma-separated list of subnet names:<br/>"main", "additional1", "additional2"; e.g.,:<br/>additional_ips =<br/>additional1, additional2</pre><br>You can also use "all" to specify all subnets.<br>If more than one additional private IP address is<br>required on the specific network interface, this<br>can be specified as " <name>:<num>", where<br/><num> is the total number of additional private<br/>IP addresses to create; e.g.,:<br/>additional_ips = additional1:2<br/>Alternatively, you can specify interface names<br/>"ethX" instead of subnet names, e.g.,:<br/>additional_ips = eth1, eth2<br/>Notes:<br/>• For HA deployment in Azure, the<br/>additional_ips parameter specifies "logical" IP<br/>addresses. For each such "logical" address,<br/>two IP addresses are allocated on the VM -<br/>one is placed behind Internal Load Balancer<br/>and is used for signaling (SIP) traffic and<br/>another one is used for media (RTP) traffic.<br/>• For HA deployment in Google Cloud, internal<br/>IP addresses are allocated on Network Load<br/>Balancer and the "main" interface is placed<br/>behind it.</num></num></name> | Update        |



| Parameter   | Applicable<br>Environment | Description   | Apply<br>Mode |
|---|---------------------------|---|---------------|
|   |                           | <ul> <li>After stack creation, use the Additional IPs<br/>parameter in the Modify dialog to change the<br/>parameter value.</li> </ul>  |               |
| additional_route_t<br>ables   | AWS                       | Applicable to multi-zone AWS deployments.<br>Defines additional route tables that should be<br>updated with virtual IP addresses.<br>Syntax: comma-separated list of <interface<br>name&gt;:<route id="" table="">. Multiple route table IDs<br/>can be specified using pipe ( ) delimiter.<br/>Example:<br/>additional_route_tables = eth1<br/>:rtb-123,eth2:rtb-567 rtb-890</route></interface<br>  | Update        |
| additional3_subnet<br>_id,<br>additional4_subnet<br>_id,<br>additional5_subnet<br>_id,<br>additional6_subnet<br>_id | AWS,<br>Azure             | Defines subnet IDs for Additional 3 to Additional<br>6 subnets.<br>For Azure, specify the subnet name, e.g.,:<br>additional3_subnet_id = voip3<br>For AWS, specify the subnet ID, c:<br>additional3_subnet_id = subnet-<br>12345<br>Note: Subnet IDs that are currently "in use" can't<br>be modified. If you want to change the subnet ID<br>of an existing network interface, first reduce the<br>number of network interfaces, update the<br>corresponding subnet ID, and then restore the<br>number of interfaces.      | Instant       |
| app_insights  | Azure                     | <ul> <li>Defines the type of data that is reported to Azure Application Insights.</li> <li>Supported values: <ul> <li>disable: (Default) Don't report any data for this stack.</li> <li>enable: Report alarms and metrics (PMs).</li> <li>alarms: Report alarms only.</li> <li>metrics: Report metrics (PMs) only.</li> </ul> </li> <li>Example: <ul> <li>app_insights = enable</li> </ul> </li> <li>See Section 7.3, Integration with Azure Application Insights for more information.</li> </ul>                          | Instant       |
| auto_shelve_time  | All                       | <ul> <li>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.</li> <li>Supported syntax:</li> <li>08:00: Time of day (24h).</li> <li>1/08:00: Weekday (0 is Sunday, 1 is Monday, 2 is Tuesday, and so on) and time.</li> <li>0,1,2/08:00: Multiple weekdays and time.</li> <li>0-5/08:00: Range of weekdays and time.</li> </ul> | Instant       |

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



| Parameter              | Applicable<br>Environment | Description   | Apply<br>Mode |
|------------------------|---------------------------|---|---------------|
|                        |                           | <pre>image_id = ami-9a50cff5</pre>  |               |
|                        |                           | After stack creation, use the <b>Image ID</b> parameter in the <b>Modify</b> dialog to change the parameter value.  |               |
| image_url              | All                       | Defines the URL that contains a plain-text file<br>with the name of the local image (instead of the<br>Marketplace image).<br>Example:<br>image_url = https://  | Rebuild       |
| ini_incremental        | All                       | Defines additional configuration parameters (in   | Rebuild       |
|                        |                           | INI file format) that's applied during stack creation / rebuild.  |               |
|                        |                           | Syntax: a single line with \n as line delimiter,<br>e.g.,:  |               |
|                        |                           | <pre>ini_incremental = EnableSyslog = 1\nSyslogServerIP = 10.1.2.3</pre>  |               |
|                        |                           | Specified configuration is applied via REST API<br>and therefore, has no size limit. Therefore, this<br>parameter is preferred over the ini_params<br>parameter that is applied via the cloud-init<br>mechanism and therefore, is limited by instance<br>user-data size.  |               |
| imds                   | AWS                       | <ul> <li>Defines the version of the AWS meta-data instance for the deployed EC2 instances.</li> <li>Supported values: <ul> <li>any: Allow both IMDSv1 and IMDSv2.</li> <li>v2: (Default) Enforce IMDSv2.</li> </ul> </li> <li>Example: <ul> <li>imds = v2</li> </ul> </li> </ul>  | Update        |
| ini_params             | All                       | Defines additional configuration parameters (in<br>INI file format) that's applied during stack<br>creation / rebuild.<br>Syntax: a single line with \n as line delimiter,<br>e.g.,:<br>ini_incremental = EnableSyslog =<br>1\nSyslogServerIP = 10.1.2.3<br>Specified configuration is applied via the cloud-<br>init mechanism and is therefore limited by | Rebuild       |
|                        |                           | Instance user-data size (for example, in AWS it's limited by 16 KB). For long configurations, use the ini_incremental parameter instead.  |               |
| ipv6_ip_sbc-<br>X_eth* | AWS                       | Defines pre-defined IPv6 addresses.<br>See also the private_ip_* and<br>public ip * parameters.   | Update        |
| ipv6_ips               | AWS,<br>Azure             | Defines the network interface names for which<br>IPv6 addresses are allocated.  | Update        |
|                        |                           | <b>In AWS</b> , IPv6 addresses are a <i>new</i> type of addresses that are globally routable. They are  |               |

| Parameter | Applicable<br>Environment | Description   | Apply<br>Mode |
|-----------|---------------------------|---|---------------|
|           |                           | assigned <i>in addition</i> to IPv4 addresses specified<br>via the public_ips and additional_ips<br>parameters. If the interface is connected to a<br>subnet that has both IPv4 and IPv6 address<br>ranges, the primary address will be IPv4, and<br>IPv6 will be assigned as the secondary address<br>with the 'ethX:100' name. If the interface is<br>connected to an IPv6-only subnet (i.e., subnet<br>lacks IPv4 address range), only IPv6 addresses<br>are allocated and assigned with regular 'ethX'<br>names. Pre-defined IPv6 addresses can be<br>specified via ipv6_ip_* parameters.<br><b>In Azure</b> , IPv6 is a <i>property</i> of normal internal<br>and public addresses. Therefore, this parameter<br>does not add new addresses, but changes the<br>type of those that are specified via the<br>public_ips or additional_ips parameters.<br>Primary addresses cannot be IPv6 and<br>therefore, this parameter by default configures<br>IPv6 on the "1st secondary" (second) IP<br>address. For example, the following creates two<br>public addresses – IPv4 and IPv6 – on an<br>interface connected to the "additional 1" subnet:<br>Public IPs: additional1:2<br>ipv6_ips = additional1<br>You can specify the secondary address index (1<br>= "1st secondary", 2 = "2nd secondary", and so<br>on) to configure IPv6 on a different secondary<br>address. For example, the following configures<br>IPv6 on the "2nd secondary" (third) IP address of |               |
|           |                           | an interface connected to the additional 1<br>subnet:   |               |
|           |                           | Additional TPs, additional1.  |               |
|           |                           | ipv6 ips = additional1.2  |               |
|           |                           | Pre-defined IPv6 addresses can be specified via the private_ip_* and public_ip_* parameters.  |               |
|           |                           | Syntax: comma-separated list of subnet names:<br>"main", "additional1", "additional2", etc<br>Example:  |               |
|           |                           | <pre>ipv6_ips = additional1,additional2</pre>   |               |
|           |                           | Alternatively, you can specify interface names<br>"ethX" instead of subnet names, e.g.,:  |               |
|           |                           | <pre>ipv6_ips = eth1,eth2</pre>   |               |
|           |                           | Notes:  |               |
|           |                           | <ul> <li>Management is always done via IPv4<br/>addresses.</li> </ul>   |               |



| Parameter        | Applicable<br>Environment | Description   | Apply<br>Mode |
|------------------|---------------------------|---|---------------|
|                  |                           | <ul> <li>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.</li> <li>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 ipv6_virtual_ips parameter to change this behavior.</li> </ul>   |               |
| ipv6_virtual_ips | AWS                       | <pre>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. Supported values:     enable: Use "virtual IP" addresses for IPv6     addresses.     disable: (default) Don't use "virtual IP"     addresses for IPv6 addresses. Example:     ipv6_virtual_ips = enable When enabled, the virtual IP address can be specified via the virtual IP address can be specified via the virtual_ip_eth* parameter, for example:     virtual_ip_eth2 = fd00::a0dc:1</pre> | Update        |
| kms_key_id       | AWS                       | <pre>Identifier of the AWS KMS key for Amazon EBS disk encryption. You can specify the key via one of the following:     Key ID     Key alias     Key ARN     Alias ARN Example:     kms-key-id = arn:aws:kms:us-east- 1: 012345678910:1234abcd-12ab6ef- 1234567890ab</pre>   | Rebuild       |
| main_nsg_id      | Azure                     | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the Security Groups chapter in the<br>Mediant VE for Azure Installation Manual for a   | Update        |

| Parameter        | Applicable<br>Environment | Description   | Apply<br>Mode |
|------------------|---------------------------|---|---------------|
|                  |                           | detailed list of rules that should be included in<br>the specific NSG.<br>Syntax is similar to ha_nsg_id parameter.<br>When modifying the security group, make sure<br>that it includes rules that enable Stack Manager<br>to access deployed instances via the HTTPS<br>protocol (TCP/443).<br>See also the voip_nsg_id parameter.   |               |
| manage_via_https | All                       | <ul> <li>Defines the protocol used by Stack Manager when connecting to the deployed stack's management interface.</li> <li>Supported values: <ul> <li>enable: (Default) Use HTTPS protocol.</li> <li>disable: Use HTTP protocol.</li> </ul> </li> <li>Example: <ul> <li>manage_via_https = disable</li> </ul> </li> </ul>   | Instant       |
| media_nsg_id     | AWS                       | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the Security Groups chapter in the<br>Mediant VE for AWS Installation Manual for a<br>detailed list of rules that should be included in<br>the specific NSG.<br>Syntax is similar to the ha_nsg_id parameter.<br>See also the oam_nsg_id and<br>signaling_nsg_id parameters.   | Update        |
| nsg_id_ethX      | AWS,<br>Azure             | <ul> <li>Defines the name of the existing Network<br/>Security Group (NSG) for a specific network<br/>interface.</li> <li>Refer to the Security Groups chapter in the<br/>Mediant VE for AWS / Azure Installation Manual<br/>for a detailed list of rules that should be included<br/>in the specific NSG.</li> <li>When modifying the security group that contains<br/>"management" rules, make sure that it includes<br/>rules that enable Stack Manager to access<br/>deployed instances via the HTTPS protocol<br/>(TCP/443).</li> <li>Syntax:</li> <li>AWS: Security Group ID. Multiple groups can<br/>be specified as a comma-separated string,<br/>e.g.,:<br/>nsg_id_eth1 = sg-123, sg-345</li> <li>Azure: Resource Group name / NSG name,<br/>e.g.,:<br/>nsg_id_eth2 = rg1/cluster-nsg</li> </ul> | Update        |

| Parameter                                  | Applicable<br>Environment | Description   | Apply<br>Mode  |
|--|---------------------------|---|--|
| oam_ip                                     | All                       | <ul> <li>Defines an IP address for management traffic (Web, SSH, and SNMP). Applicable for HA deployments only.</li> <li>Syntax: <ul> <li>"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.</li> <li>"internal": Implies the following configuration: Public IPs: main</li> <li>Additional IPs: main oam_ip = internal</li> </ul> </li> <li>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 uses it for management traffic.</li> <li>AWS: Creates an additional private IP address on the "main" network interface and uses it for management traffic.</li> <li>"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.</li> </ul> | Update<br>(Azure)<br>or<br>Rebuild<br>(AWS,<br>Google) |
| proximity_placeme<br>nt_group_vm_size<br>s | Azure                     | Defines the optional "intent" parameter of<br>proximity placement groups created as part of<br>the stack deployment. Refer to<br><u>https://learn.microsoft.com/en-us/azure/virtual-</u><br><u>machines/co-location</u> for details.<br>Syntax: comma-separated list of VM sizes.<br>Example:<br>proximity_placement_group_vm_sizes<br>= Standard_D2ds_v5, Standard_D4ds_v5<br>Typically used together with the<br>proximity_placement_group_vm_zone<br>parameter (see below).  | Can't be<br>modified<br>after stack<br>creation        |
| proximity_placeme<br>nt_group_vm_zon<br>e  | Azure                     | Defines the optional "zone" parameter of proximity placement groups created as part of the stack deployment. Refer to <u>https://learn.microsoft.com/en-us/azure/virtual-machines/co-location for details.</u>  | Can't be<br>modified<br>after stack<br>creation        |

| Parameter                 | Applicable<br>Environment | Description  | Apply<br>Mode |
|---------------------------|---------------------------|--|---------------|
|                           |                           | Enables deployment of Mediant CE into an<br>availability set located in the specific availability<br>zone.<br>Example:   |               |
|                           |                           | <pre>proximity_placement_group_vm_zone = 1</pre>   |               |
|                           |                           | Due to limitations of Azure API, if you specify this parameter you must also specify the proximity_placement_group_vm_sizes parameter.   |               |
| private_ip_sbc-<br>X_eth* | All                       | Defines pre-defined private IP addresses. See Section 3.25.3 for more information.   | Update        |
|                           |                           | See also the ipv6_ip_* and public_ip_* parameters.   |               |
| public_ips                | All                       | Defines the Signaling Component's network<br>interface names for which public IP addresses<br>are allocated and optionally, the number of<br>corresponding IP addresses.   | Update        |
|                           |                           | During stack creation (via Web interface), Stack<br>Manager lets you specify which subnets (and<br>corresponding network interfaces) are assigned<br>with public (Elastic) IP addresses using the<br><b>Public IPs</b> parameter in the <b>Networking</b><br>section.        |               |
|                           |                           | When the public_ips advanced configuration<br>parameters is specified, it overrides any value<br>configured by the <b>Public IPs</b> parameter.<br>Typically, you will use this parameter when you<br>need to create multiple IP addresses on the<br>same network interface. |               |
|                           |                           | Syntax: comma-separated list of subnet names:<br>"main", "additional1", "additional2", etc<br>Example:   |               |
|                           |                           | <pre>public_ips = additional1,additional2</pre>  |               |
|                           |                           | You can also use "all" to specify all subnets.   |               |
|                           |                           | If more than one public IP address is required on<br>the specific network interface, this can be<br>specified as " <name>:<num>", where <num> is<br/>the total number of public IP addresses to be<br/>created, e.g.,:</num></num></name>                                    |               |
|                           |                           | Alternatively, you can specify the interface   |               |
|                           |                           | names "ethX" instead of subnet names, e.g.,:   |               |
|                           |                           | <pre>public_ips = eth1,eth2</pre>  |               |
|                           |                           | Notes:   |               |
|                           |                           | <ul> <li>For FA deployment on Azure, the<br/>public_ips parameter specifies "logical" IP<br/>addresses. For each such "logical" address,<br/>two IP addresses are allocated on the VM -</li> </ul>   |               |

| Parameter                                   | Applicable<br>Environment | Description   | Apply<br>Mode                                   |
|---|---------------------------|---|---|
|   |                           | <ul> <li>one is placed behind Public Load Balancer<br/>and is used for signaling (SIP) traffic and<br/>another one has public IP address attached<br/>directly to the VM and is used for media<br/>(RTP) traffic.</li> <li>For HA deployment on <b>Google Cloud</b>, public<br/>IP addresses are supported only on the<br/>"main" interface. They are allocated on<br/>Network Load Balancer and the "main"<br/>interface is placed behind it.</li> <li>Stack Manager implicitly creates all private IP<br/>addresses required for public IP address<br/>assignment.</li> <li>After stack creation, use the <b>Public IPs</b><br/>parameter in the <b>Modify</b> dialog to change the<br/>parameter value.</li> </ul> |   |
| public_ip_eth*,<br>public_ip_sbc-<br>X_eth* | All                       | Defines pre-defined public IP addresses. See<br>Section 3.25.3 for more information.<br>See also the ipv6_ip_* and private_ip_*<br>parameters.  | Update  |
| public_ip_prefix                            | Azure                     | <pre>Defines a Public IP Prefix from which public IP addresses are allocated. Syntax: comma-separated list of elements:</pre>   | Update  |
| resource_group                              | Azure                     | Defines the name of the existing Resource<br>Group.<br>If not empty, stack resources are deployed in<br>this Resource Group instead of creating a new<br>one. The Resource Group must be empty prior<br>to stack creation.<br>Example:<br>resource_group = SbcGroup1  | Can't be<br>modified<br>after stack<br>creation |
| sbc_image_id                                | AWS,<br>Azure             | Defines the local image (instead of the<br>Marketplace image).<br>In contrast to the image_id parameter, this<br>parameter is applied via the <b>Rebuild (</b> and not<br><b>Update)</b> operation.<br>For Azure, specify the resource group name<br>followed by the image name, e.g.,:<br>image_id = rg1/image1  | Rebuild   |
| Parameter                | Applicable<br>Environment | Description   | Apply<br>Mode                                   |
|--------------------------|---------------------------|---|---|
|                          |                           | For AWS, specify the AMI ID, e.g.,:<br>image_id = ami-9a50cff5  |   |
| sbc_tags                 | Azure                     | Defines tags that are assigned to VM instances.<br>These tags are added "on top" of tags specified<br>via the tags parameter, which are assigned to<br>all created resources.<br>Syntax: comma-separated list of name=value<br>pairs, e.g.,:<br>sbc_tags = duration=overnight   | Update  |
| signaling_nsg_id         | AWS                       | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the Security Groups chapter in the<br>Mediant VE for AWS Installation Manual for a<br>detailed list of rules that should be included in<br>the specific NSG.<br>Syntax is similar to ha_nsg_id parameter.<br>See also the media_nsg_id and oam_nsg_id<br>parameters. | Update  |
| spot_instances           | Azure                     | <ul> <li>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.</li> <li>Supported values: <ul> <li>enable: Use Spot instances.</li> </ul> </li> <li>disable: (Default) Use regular instances.</li> </ul> <li>Example: <ul> <li>spot_instances = enable</li> </ul> </li>                           | Can't be<br>modified<br>after stack<br>creation |
| storage_account_t<br>ype | Azure                     | <pre>Defines the storage account type for managed<br/>disks.<br/>Valid values include:<br/>• Standard_LRS<br/>• Premium_LRS<br/>• StandardSSD_LRS<br/>Example:<br/>storage_account_type = Premium_LRS</pre>   | Rebuild   |
| tags                     | All                       | <ul> <li>Defines tags that areassigned to the following stack resources:</li> <li>AWS: Tags are assigned to EC2 instance, volume, network interfaces and Elastic IPs.</li> <li>Azure: Tags are assigned to all created resources.</li> <li>Google Cloud: Tags are applied to VM instances.</li> <li>Syntax:</li> </ul>  | Azure:<br>Update<br>Other:<br>Rebuild           |



| Parameter                         | Applicable<br>Environment | Description  | Apply<br>Mode                                   |
|-----------------------------------|---------------------------|--|---|
|                                   |                           | <ul> <li>AWS or Azure: comma-separated list of name=value pairs, e.g.,:<br/>tags = type=sbc, product=ve</li> <li>Google: comma-separated list of tags tags = sbc, ve</li> <li>See also sbc_tags parameter.</li> </ul>  |   |
| update_needed                     | All                       | When set to "reset", it turns off the<br>update_needed flag without applying any<br>changes upon the next "update" action.<br>Example:<br>update_needed = reset  | Update  |
| user_data                         | All                       | Defines additional cloud-init configuration<br>parameters.<br>Syntax: a single line with \n as line delimiter.<br>Example:<br>sc_user_data = #customer-<br>id\n123456\n#license-<br>key\nokRTr5top   | Rebuild   |
| use_availability_se<br>t          | Azure                     | <ul> <li>Applicable for HA deployments. Defines whether availability set is created for 'availability set' deployment topology.</li> <li>Supported values: <ul> <li>enable: (Default) Create availability set.</li> <li>disable: Don't create availability set. Virtual machines are deployed without any redundancy constraints.</li> </ul> </li> <li>Example: <ul> <li>use_availability_set = disable</li> </ul> </li> </ul> | Can't be<br>modified<br>after stack<br>creation |
| use_placement_gr<br>oup           | AWS                       | <ul> <li>Applicable for HA deployments. Defines if two<br/>Mediant VE instances are deployed in the<br/>placement group.</li> <li>Supported values: <ul> <li>enable: (Default) Use placement group.</li> </ul> </li> <li>disable: Don't use placement group.</li> </ul> <li>Example: <ul> <li>use_placement_group = disable</li> </ul> </li>   | Can't be<br>modified<br>after stack<br>creation |
| use_proximity_<br>placement_group | Azure                     | <ul> <li>Applicable for HA deployments. Defines if two<br/>Mediant VE instances are deployed in the<br/>proximity placement group.</li> <li>Supported values: <ul> <li>enable: (Default) Use proximity placement<br/>group.</li> </ul> </li> <li>disable: Don't use proximity placement<br/>group.</li> </ul> <li>Example:<br/>use_proximity_placement_group =<br/>disable</li>  | Can't be<br>modified<br>after stack<br>creation |

| Parameter       | Applicable<br>Environment | Description   | Apply<br>Mode |
|-----------------|---------------------------|---|---------------|
| virtual_ip_eth* | AWS                       | Applicable to multi-zone AWS deployments.<br>Defines virtual IP addresses for communication<br>in the VPC or via the Transit Gateway.<br>Example:<br>virtual_ip_eth2 = 10.5.1.10  | Update        |
| virtual_ips     | AWS                       | <pre>Defines whether "virtual IPs" are used for IPv4 addresses in multi-zone AWS deployments. Supported values:     enable (default)     disable Example:     virtual_ips = disable</pre>   | Rebuild       |
| voip_nsg_id     | Azure                     | Defines the name of the existing Network<br>Security Group (NSG) to be used instead of<br>default security groups created by Stack<br>Manager.<br>Refer to the <i>Security Groups</i> chapter in the<br><i>Mediant VE for Azure Installation Manual</i> for a<br>detailed list of rules that should be included in<br>the specific NSG.<br>Syntax is similar to the ha_nsg_id parameter.<br>See also the main_nsg_id parameter. | Update        |
| volume_type     | AWS                       | Defines the volume type for EBS disks.<br>Valid values include:<br>gp2<br>gp3<br>io1<br>io2<br>sc1<br>sc2<br>standard<br>Example:<br>volume_type = gp3  | Rebuild       |

### 3.8.11.3 Advanced Configuration for VoiceAl Connect

The following table describes advanced parameters available for VoiceAI Connect.

| Table 3-7: Advanced | Parameters | Description |
|---------------------|------------|-------------|
|---------------------|------------|-------------|

| Parameter            | Applicable<br>Environment | Description   | Apply<br>Mode |
|----------------------|---------------------------|---|---------------|
| auto_start_time      | All                       | <ul> <li>Defines the time of day when stack automatically starts.</li> <li>Supported syntax: <ul> <li>08:00: Time of day (24h).</li> <li>1/08:00: Weekday (0=Sunday, 1=Monday,, 6=Saturday) and time.</li> <li>0,1,2/08:00: Multiple weekdays and time.</li> <li>0-5/08:00: Range of weekdays and time.</li> <li>0,1/08:00]2-4/09:00: Multiple statements.</li> </ul> </li> <li>Example: <ul> <li>auto_start_time = 08:00</li> </ul> </li> </ul>  | Instant       |
| auto_stop_time       | All                       | Defines the time of day when stack<br>automatically stops.<br>Syntax is identical to the<br><b>auto_start_time</b> parameter.<br>Example:<br>auto_stop_time = 22:00   | Instant       |
| automatic_update_url | All                       | <pre>Defines the automatic update URL that is configured on the "worker" SBC instances and front-end SBC. If configured: • On the front-end SBC, the IncrementalIniFileURL parameter is provisioned with the value: \${url}/global/fe- incremental.ini • On the "worker" SBCs, the IncrementalIniFileURL parameter is provisioned with the value: \${url}/global/sbc- incremental.ini Example: automatic_update_url = https://my.company.com/file s During stack creation (via Web interface), this parameter is the Create dialog.</pre> | Update        |
| bot_dialplan         | All                       | Defines the bot Dial Plan on te<br>"worker" SBCs. If enabled, the <code>bots</code>   | Update        |

| Parameter                         | Applicable<br>Environment | Description   | Apply<br>Mode       |
|-----------------------------------|---------------------------|---|---------------------|
|                                   |                           | <pre>Dial Plan is created and the<br/>DialPlanCSVFileUrl<br/>configuration parameter is<br/>provisioned with the value:<br/>\${url}/global/bot-<br/>dialplan.csv<br/>Supported values:<br/>disable (Default)<br/>enable<br/>Example:<br/>bot_dialplan = enable<br/>During stack creation (via Web<br/>interface), this parameter is<br/>configured via the Use Bot Dialplan<br/>parameter in the Create dialog.</pre> |                     |
| center_base_url<br>center_token   | All                       | Defines the base URL and API<br>credentials of the pre-created<br>configuration manager (center)<br>instance.<br>If you use these parameters, you are<br>also expected to use the<br>sm_env_vars advanced config<br>parameter to specify the<br>MONGODB_HOST parameter for<br>session managers.<br>Example:<br>center_base_url =<br>https://fqdn<br>center_token = 123456   | Rebuild<br>(center) |
| center_disk_size,<br>sm_disk_size | AWS,<br>Azure             | Defines the disk size (in GB) for<br>configuration manager / data center<br>and session manager instances.<br>Example:<br>center_disk_size = 64   | Rebuild             |
| center_env_vars,<br>sm_env_vars   | All                       | Defines additional environment<br>variables for configuration manager /<br>data center and session manager<br>instances.<br>Syntax: a single line with \n as line<br>delimiter, e.g.,:<br>sm_env_vars = VAR1=val1<br>center_env_vars =<br>VAR2=val2\nVAR3=val3  | Update              |
| center_iam_role,<br>sm_iam_role   | AWS                       | Defines the IAM role assigned to the<br>configuration manager / data center<br>or session manager virtual machine<br>instances.<br>Example:   | Rebuild             |

| Parameter  | Applicable<br>Environment | Description  | Apply<br>Mode |
|--|---------------------------|--|---------------|
|  |                           | <pre>center_iam_role = CENTER-<br/>ROLE</pre>  |               |
| center_image_id,<br>sbc_image_id,<br>sm_image_id,<br>vaic_image_id     | All                       | <ul> <li>Defines the local image for<br/>corresponding stack components.</li> <li>Use the vaic_image_id parameter<br/>to specify the local image for both<br/>configuration manager / data center<br/>and session manager components.</li> <li>Syntax:</li> <li>AWS: AMI ID, e.g.,:<br/>vaic_image_id = ami-<br/>8b41cff2</li> <li>Azure: Resource Group name /<br/>image name, e.g.,:<br/>vaic_image_id =<br/>rg1/image1</li> </ul> | Rebuild       |
| center_image_url,<br>sbc_image_url,<br>sm_image_url,<br>vaic_image_url | All                       | <pre>Defines a URL that contains a text<br/>file with the value of the<br/>corresponding *_image_id<br/>parameter.<br/>Syntax:<br/>• (http https)://<path><br/>• (http https)://<username>:<passw<br>ord&gt;@<path><br/>• file://<filepath><br/>Example:<br/>vaic_image_url =<br/>https://sbc123.blob.core.<br/>windows.net/pub/vaic_imag<br/>e.txt</filepath></path></passw<br></username></path></pre>                             | Rebuild       |
| center_instance_type,<br>sbc_instance_type,<br>sm_instance_type        | All                       | Defines the instance type / VM size<br>for the corresponding components<br>("configuration manager / data<br>center" / SBC / "session manager").<br>Example:<br>sm_instance_type =<br>Standard_D4ds_v4   | Rebuild       |
| center_nsg_id,<br>sbc_nsg_id,<br>sbc_public_nsg_id,<br>sm_nsg_id       | AWS                       | <ul> <li>Defines the name of the existing<br/>Network Security Group (NSG) for<br/>the corresponding component's<br/>network interface, instead of creating<br/>a new one.</li> <li>Syntax:</li> <li>AWS: Security Group ID, e.g.,:<br/>center_nsg_id = sg-123</li> <li>Azure: Resource Group name /<br/>NSG name, e.g.,:<br/>sm nsg_id = rg1/sm-nsg</li> </ul>  | Rebuild       |

| Parameter   | Applicable<br>Environment | Description  | Apply<br>Mode |
|---|---------------------------|--|---------------|
| center_public_ips,<br>sbc_public_ips,<br>sm_public_ips                                  | All                       | <ul> <li>Defines network interface names on the corresponding components ("configuration manager / data center" / SBC / "session manager") that should be assigned with public IP addresses.</li> <li>Syntax: comma-separated list of interface names:</li> <li>main: Assign a public IP address to the "main" interface (eth0).</li> <li>public: Assign a public IP address to the "public" interface (eth1) – applicable to SBC components only.</li> <li>Examples:</li> <li>sm_public_ips = main sbc_public_ips = main, public</li> </ul>   | Rebuild       |
| center_startup_script_<br>url,<br>sm_startup_script_url,<br>vaic_startup_script_ur<br>l | All                       | <pre>Defines a URL that contains a<br/>custom script that is run on the<br/>"center" or "session manager" virtual<br/>machine instances during their<br/>creation. This, for example, can be<br/>used to install additional software on<br/>"center" or "session manager" virtual<br/>machines.<br/>Sample script:<br/>#!/bin/bash<br/>echo "text" &gt;<br/>/opt/test.txt<br/>Use the vaic_startup_script_url<br/>parameter if you want to run the<br/>same script on both "center" and<br/>"session manager" virtual machines.<br/>Syntax:<br/>(http https)://<path><br/>(http https)://<username>:<passw<br>ord&gt;@<path><br/>file://<filepath><br/>Example:<br/>vaic_startup_script_url<br/>https://sbc123.blob.core.<br/>windows.net/pub/startup.s<br/>h</filepath></path></passw<br></username></path></pre> | Rebuild       |
| center_tags,<br>sm_tags,<br>sbc_tags  | All                       | Defines tags assigned to the following signaling components' resources:  | Rebuild       |

| Parameter                   | Applicable<br>Environment | Description  | Apply<br>Mode            |
|-----------------------------|---------------------------|--|--------------------------|
|                             |                           | <ul> <li>AWS: Tags are assigned to EC2<br/>instance, volume, network<br/>interfaces and Elastic IPs.</li> <li>Azure and Google Cloud: Tags<br/>are assigned to VM instances.</li> <li>Syntax:</li> <li>AWS or Azure: comma-<br/>separated list of name=value<br/>pairs, e.g.,:<br/>sm_tags =<br/>type=vaic, role=sm</li> <li>Google: comma-separated list of<br/>tags<br/>sm_tags = vaic, sm</li> <li>See also the tags and<br/>common_tags parameters.</li> </ul> |                          |
| common_tags                 | AWS                       | Defines tags that are assigned to<br>created network security groups.<br>Syntax: comma-separated list of<br>name=value pairs.<br>Example:<br>common_tags = type=vaic<br>See also the center_tags,<br>sbc_tags and sm_tags<br>parameters.   | Rebuild                  |
| deployment_mode             | All                       | Defines the deployment mode for<br>VoiceAI Connect components.<br>Supported values:<br>docker (Default)<br>podman<br>"Podman" mode is supported only on<br>RHEL / CentOS / Rocky Linux hosts.  | Rebuild                  |
| frontend_proxyset_na<br>me  | All                       | Defines the name of the Proxy Set on<br>the front-end SBC, which is updated<br>with the VoiceAI Connect worker<br>SBC IP addresses upon initial<br>configuration, scale-out/scale-in<br>and "reconfigure" operation. If not<br>specified, the 'vaic' Proxy Set is used.<br>Example:<br>frontend_proxyset_name =<br>my-ps-1   | Reconfig<br>ure<br>(fe)  |
| frontend_transport_ty<br>pe | All                       | Defines the transport type for<br>communication between the front-<br>end SBC and VoiceAl Connect<br>worker SBCs.<br>Supported values:<br><b>udp</b> (Default)<br><b>tcp</b>   | Reconfig<br>ure<br>(sbc) |

| Parameter        | Applicable<br>Environment | Description  | Apply<br>Mode |
|------------------|---------------------------|--|---------------|
|                  |                           | <pre>• tls Example:     frontend_transport_type = tls</pre>  |               |
| imds             | AWS                       | <ul> <li>Defines the version of the AWS meta-data instance for the deployed EC2 instances.</li> <li>Supported values:</li> <li>any: Allow both IMDSv1 and IMDSv2.</li> <li>v2: (Default) Enforce IMDSv2.</li> <li>Example:</li> <li>imds = v2</li> </ul>   | Update        |
| kms_key_id       | AWS                       | <pre>Identifier of the AWS KMS key for<br/>Amazon EBS disk encryption. You<br/>can specify the key via one of the<br/>following:<br/>Key ID<br/>Key alias<br/>Key ARN<br/>Alias ARN<br/>Example:<br/>kms-key-id =<br/>arn:aws:kms:us-east-1:<br/>012345678910:1234abcd-<br/>12ab6ef-1234567890ab</pre>                         | Rebuild       |
| manage_via_https | All                       | <ul> <li>Defines the protocol used by Stack<br/>Manager when connecting to the<br/>deployed stack's management<br/>interface.</li> <li>Supported values: <ul> <li>enable: (Default) Use HTTPS<br/>protocol.</li> </ul> </li> <li>disable: Use HTTP protocol.</li> <li>Example:<br/>manage_via_https =<br/>disable</li> </ul>   | Instant       |
| oam_ip           | All                       | <ul> <li>Defines the IP addresses used by<br/>Stack Manager to manage the<br/>VoiceAI Connect stack.</li> <li>Supported values: <ul> <li>internal: (Default) Use the internal<br/>IP addresses.</li> </ul> </li> <li>public: Use the public IP<br/>addresses.</li> </ul> <li>Example: <ul> <li>oam_ip = public</li> </ul></li> | Rebuild       |



| Parameter   | Applicable<br>Environment | Description   | Apply<br>Mode                                      |
|---|---------------------------|---|--|
|   |                           | During stack creation (via Web<br>interface), this parameter is<br>configured via the <b>Manage via</b><br><b>Public IPs</b> parameter in the <b>Create</b><br>dialog.  |  |
| private_ip_center_eth*<br>,<br>private_ip_sbc-X_eth*,<br>private_ip_sm-X_eth* | All                       | Defines predefined private IP<br>addresses. See Section 3.25.3 for<br>more information.<br>See also the public_ip_*<br>parameters.  | Rebuild  |
| public_ip_center_eth*,<br>public_ip_sbc-X_eth*,<br>public_ip_sm-X_eth*        | All                       | Defines predefined public IP<br>addresses. See Section 3.25.3 for<br>more information.<br>See also the private_ip_*<br>parameters.  | Rebuild  |
| resource_group  | Azure                     | Defines the name of the existing<br>Resource Group.<br>If not empty, stack resources are<br>deployed in this Resource Group<br>instead of creating a new one. The<br>Resource Group must be empty prior<br>to stack creation.<br>Example:<br>resource_group = MyGroup1  | Can't be<br>modified<br>after<br>stack<br>creation |
| sbc_ini_config  | All                       | Defines additional configuration<br>parameters (in INI file format) for<br>SBC instances during stack creation /<br>rebuild.<br>Syntax: a single line with \n as line<br>delimiter, e.g.,:<br>ini_incremental =<br>EnableSyslog =<br>1\nSyslogServerIP =<br>10.1.2.3<br>Specified configuration is applied via<br>REST API and therefore, has no size<br>limit. Therefore. this parameter is<br>preferred over the sbc_ini_params<br>parameter that is applied via the<br>cloud-init mechanism and is<br>therefore, limited by instance user-<br>data size. | Update   |
| sbc_ini_params  | All                       | Defines additional configuration<br>parameters (in INI file format) for<br>SBC instances during stack creation /<br>rebuild.<br>Syntax: a single line with \n as line<br>delimiter, e.g.,:<br>Example:  | Rebuild  |

| Parameter                     | Applicable<br>Environment | Description  | Apply<br>Mode                                      |
|-------------------------------|---------------------------|--|--|
|                               |                           | <pre>sc_ini_params =<br/>EnableSyslog =<br/>1\nSyslogServerIP =<br/>10.1.2.3 Specified configuration is applied via<br/>the cloud-init mechanism and is<br/>therefore limited by instance user-<br/>data size (for example, in AWS it's<br/>limited by 16 KB). For long<br/>configurations, use the<br/>sbc_ini_config parameter<br/>instead</pre>   |  |
| sbc_user_data                 | All                       | Defines additional cloud-init<br>configuration parameters for SBC<br>instances.<br>Syntax: a single line with \n as line<br>delimiter.<br>Example:<br>sc_user_data = #customer-<br>id\n123456\n#license-<br>key\nokRTr5top   | Rebuild  |
| ssh_key_user,<br>ssh_key_file | AWS                       | <pre>By default, Stack Manager uses<br/>credentials provisioned during initial<br/>stack creation to establish connection<br/>with "configuration manager"<br/>instances, for example during<br/>Upgrade operation.<br/>If you want Stack Manager to<br/>authenticate with an SSH public key<br/>instead, configure the following<br/>parameters:<br/>ssh_key_user: Username.<br/>ssh_key_file: Full path to the<br/>private key file in PEM format<br/>stored on Stack Manager VM and<br/>readable by the stack_mgr user.<br/>Example:<br/>ssh_key_file =<br/>/var/stack_mgr/ssh_key_1.pe<br/>m</pre> | Rebuild  |
| spot_instances                | Azure                     | <ul> <li>Enables Azure Spot instances for testing environments. Note that Spot instances might be abruptly stopped and therefore, should never be used in production environment.</li> <li>Supported values:</li> <li>enable: Use Spot instances.</li> </ul>   | Can't be<br>modified<br>after<br>stack<br>creation |

| Parameter            | Applicable<br>Environment | Description  | Apply<br>Mode                                      |
|----------------------|---------------------------|--|--|
|                      |                           | <ul> <li>disable: (Default) Use regular<br/>instances.</li> <li>Example:<br/>spot_instances = enable</li> </ul>  |  |
| storage_account_type | Azure                     | Defines the storage account type for<br>managed disks.<br>Valid values include:<br>• Standard_LRS<br>• Premium_LRS<br>• StandardSSD_LRS<br>Example:<br>storage_account_type =<br>Premium_LRS   | Rebuild  |
| tags                 | Azure                     | <ul> <li>Defines tags that are assigned to all created stack resources.</li> <li>Syntax: comma-separated list of name=value pairs</li> <li>Example:         <ul> <li>tags =</li> <li>type=vaic, region=us</li> </ul> </li> <li>You can also use the center_tags, sbc_tags, and sm_tags         <ul> <li>parameters to define additional tags that are assigned to corresponding VMs only.</li> </ul> </li> </ul> | Azure:<br>Update<br>Other:<br>Rebuild              |
| volume_type          | AWS                       | Defines the volume type for EBS<br>disks.<br>Valid values include:<br>gp2<br>gp3<br>io1<br>io2<br>sc1<br>sc2<br>standard<br>Example:<br>volume_type = gp3  |  |
| use_placement_group  | AWS                       | <ul> <li>Defines if components are deployed<br/>in the placement group.</li> <li>Supported values:</li> <li>enable: (Default) Use placement<br/>group.</li> <li>disable: Don't use placement<br/>group.</li> <li>Example:</li> <li>use_placement_group =<br/>disable</li> </ul>  | Can't be<br>modified<br>after<br>stack<br>creation |

| Parameter                         | Applicable<br>Environment | Description   | Apply<br>Mode                                      |
|-----------------------------------|---------------------------|---|--|
| use_proximity_<br>placement_group | Azure                     | <ul> <li>Defines if components are deployed<br/>in the proximity placement group.</li> <li>Supported values: <ul> <li>enable: (Default) Use proximity<br/>placement group.</li> </ul> </li> <li>disable: Don't use proximity<br/>placement group.</li> </ul> <li>Example: <ul> <li>use_proximity_placement_g<br/>roup = disable</li> </ul></li> | Can't be<br>modified<br>after<br>stack<br>creation |

#### **Checking Stack State and Configuration** 3.9

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: Sta   | ck Information Page  |
|--|--|
| CC stack_mgr Stacks Configuration Logs About   | Logout   |
| Start Scale Out Scale In Scale To  | 🕈 Modify 💭 Update 🛛 🚥 More 🖌 📋 Delete  |
|  | cel  |
| Namece1TypeMediant CEEnvironmentAWSStaterunningIP Address3.124.221.82Manage via HTTPSenableCreated OnNov 21, 2019 11:41:12Started OnNov 21, 2019 11:41:12Minimum number of media components2   | Automatic scaling       disable         Media utilization scale in threshold       > 250% free         Media utilization scale out threshold       < 100% free         DSP utilization scale out 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 |
| Maximum number of media components 5 Update needed false Modified parameters   | Media Components Instance type r4.large Image ID Profile forwarding  |
| Signaling Components       Instance Type     r4.2xlarge       Image ID     Disk Size     10 GB       ID     IP Address     Status     Type       sc-1     172.31.237.113     active     r4.2xlarge       sc-2     standby     r4.2xlarge | 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       2       2       2         Free media resources       200%       200%       2  |

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 recreate the stack while preserving its current network 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    |   |
| dvanced conf<br>public_ip_sbc_th<br>public_ip_sbc-1_et<br>private_ip_sbc-1_e<br>private_ip_sbc-2_et<br>public_ip_sbc-2_et | ig<br>I = Eitan-HA-VE/<br>th1 = Eitan-HA-V<br>th0 = 10.9.0.5<br>th1 = 10.9.1.4,10<br>th1 = Eitan-HA-V<br>th0 = 10.9.0.4 | 'Eitan-HA-VE-sbc-eth1-ip<br>E/Eitan-HA-VE-sbc-1-eth1-ij<br>39.1.5<br>E/Eitan-HA-VE-sbc-2-eth1-ij | p                 | ĺ |

Figure 3-26: Show IP Addresses Page

# 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).

| Subnet: subnet-bc46b8cb (main)   |
|--|
| Name: main   |
| Availability zone: eu-west-la  |
| CIDR: 1/2.31.32.0/20   |
| Route table: rtb-09bd915c613bdc464   |
|  |
| Route tables   |
|  |
| Route table: rtb-09bd915c613bdc464   |
| Routes:  |
| Destination: 172.31.0.0/16. Target: local                                    |
|  |
| *****  |
| SUMMARY  |
| *****  |
| The following problems were detected:  |
| 1. Route table rtb-09bd915c613bdc464 lacks default route to Internet Gateway |
| v  |
|  |
|  |
| Copy to clipboard  |
|  |
|  |
| Class  |
| Close  |
|  |

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.

| STACK_Mgr Stack   | s Configuration Log  | js About   |   |  |  | Admin  | Logout                               |
|---|--|--|---|--|--|--|--------------------------------------|
| Start Stop 💝 Heal   | ▲ Scale Out  | icale In 💽 Scale To  | 🖍 Modify  | C Update   | ••• More 🗸                               | 💼 Delete                                     |                                      |
| Checking connectivity   |  |  |   |  |  |  | 8                                    |
| Initializing Azure client done<br>Checking connectivity with SBC                                      | Coluster failed  |  |   |  |  |  |                                      |
| ERROR: Stack Manager cannot<br>This means that some network   | establish connection with<br>configuration / security<br>ato a different Vinet / Sub | h the SBC cluster via H<br>equipment is blocking                                 | TTPS protocol - re<br>connection betwee<br>main interface (at | equest timed c<br>een Stack Man                    | out.<br>ager and SBC clu                 | ister.                                       | noctivity                            |
| Also verify configuration of Ne<br>allow traffic from Stack Manag<br>check that Firewall configuratio | twork Security Groups fo<br>er via HTTPS protocol (tc<br>on is not blocking HTTPS    | r SBC cluster's main int<br>p/443). Finally log into<br>traffic from the Stack N | terface (eth1) - at<br>the SBC cluster's<br>Manager to OAM    | both Interface<br>Web UI, navig<br>interface (eth1 | and Subnet leve<br>ate to IP NETWO<br>). | el - and make sure the<br>RK > SECURITY > Fi | nectivity.<br>nat they<br>rewall and |
| -   | -  |  | -   |  |  |  |                                      |

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 Co | onnectivity 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   |
| 20.99.156.30  |
| Username  |
| StackMgr  |
| Password  |
| Update Cancel   |



**Note:** 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.

| ack_mgr          | Stacks   | Configuration   | Logs  | About   |  |   |  |   | Logout   |
|------------------|--|---|---|---|--|---|--|---|--|
| Create new stack |  |   |   |   |  |   |  |   |  |
|                  |  |   |   |   | Stacks   |   |  |   |  |
| lame             | Ту   | pe  |   |   | Environment  | State   | Alarms   | IP Address  |  |
| lex-ce-test-1    | М  | ediant CE   |   |   | AWS  | running   | mc-2-down  | 18.158.47.8   |  |
| lex-ce-test-2    | М  | ediant CE   |   |   | Azure  | stopped   |  | 40.64.82.179  |  |
|                  | a ck_mgr<br>Create new stack<br>lame<br>lex-ce-test-1<br>lex-ce-test-2 | ack_mgr Stacks Create new stack Iame Ty lex-ce-test-1 M lex-ce-test-2 M | ack_mgr     Stacks     Configuration       • Create new stack         Jame     Type       lex-ce-test-1     Mediant CE       lex-ce-test-2     Mediant CE | ack_mgr     Stacks     Configuration     Logs       Create new stack     Type     Image: Stacks     Image: Stacks       Iame     Type     Image: Stacks       Iame     Mediant CE     Image: Stacks | ack_mgr     Stacks     Configuration     Logs     About       • Create new stack     Image: Stacks     Image: Stacks     Image: Stacks     Image: Stacks     Image: Stacks       Iame     Type     Image: Stacks     Image: Stacks     Image: Stacks     Image: Stacks       Iame     Type     Image: Stacks     Image: Stacks     Image: Stacks     Image: Stacks       Iame     Type     Image: Stacks     Image: Stacks     Image: Stacks     Image: Stacks       Iame     Mediant CE     Image: Stacks     Image: Stacks     Image: Stacks | ack_mgr     Stacks     Configuration     Logs     About       • Create new stack       Stacks       Jame     Type     Environment       lex-ce-test-1     Mediant CE     AWS       lex-ce-test-2     Mediant CE     Azure | Type       Environment       State         lex-ce-test-1       Mediant CE       AWS       running         lex-ce-test-2       Mediant CE       Azure       stopped | Ack_mgr       Stacks       Configuration       Logs       About         • Create new stack  lame       Type | Ack_mgr       Stacks       Configuration       Logs       About         • Create new stack |

#### Figure 3-30: Active Alarms in Stacks Summary Screen

|             | 0              |                |            |            |                           |                   | •                     |        |
|-------------|----------------|----------------|------------|------------|---------------------------|-------------------|-----------------------|--------|
| 🕻 stack_mgr | Stacks Co      | nfiguration Lo | gs About   |            |                           |                   |                       | Logout |
|             |                |                |            |            |                           |                   |                       |        |
| Start Stor  | p 😵 Heal       | ▲ Scale Out    | ✓ Scale In | • Scale To | 🗹 Modify 😂 Update         | ••• More 🛩        | 🛅 Delete              |        |
|             |                |                |            |            |                           |                   |                       |        |
|             |                |                |            | alex-ce-   | test-1                    |                   |                       |        |
|             |                | General        |            |            |                           | Active Alarms     | 5                     |        |
| Name        | alex-ce-test-1 |                |            |            | Description               |                   | Raised on             |        |
| Туре        | Mediant CE     |                |            |            | Media component 'mc-2'    | is 'disconnected' | Jul 02, 2020 09:47:56 |        |
| Environment | AWS            |                |            |            |                           |                   |                       |        |
| State       | running        |                |            |            | A                         | Automatic Scali   | na                    |        |
| IP Address  | 18.158.47.8    |                |            |            |                           |                   |                       |        |
| OS Version  | 6              |                |            |            | Automatic scaling         | c                 | lisable               |        |
|             |                |                |            |            | Media utilization scale i | n threshold >     | 250% free             |        |

Figure 3-31: Active Alarms in Stack Information Page

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



Note: 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").

| Scale Out  |
|--|
| How many Media Components do you want to activate? |
| e.g. mc-3,mc-4                                     |
| Scale Out Cancel                                   |

#### Figure 3-32: Scale Out Operation

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: S | cale In Operation |
|----------------|-------------------|
|                |                   |

| Scale In   |
|--|
| How many Media Components do you want to deactivate? |
| e.g. mc-3,mc-4                                       |
| Scale In 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 |
|--------|-------|-------|------|-----------|
|--------|-------|-------|------|-----------|

| Scale To  |
|---|
| How many Media Components do you want to be active? |
| Scale To 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

Note: 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:

+----+ | 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 %



**Note:** 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.

| Modify stack  |           |          |
|---|-----------|----------|
| General   |           | <b>^</b> |
| Minimum number of media components                      | 2         |          |
| Maximum number<br>of media<br>components <sup>(1)</sup> | 3         |          |
| Automatic Scaling                                       |           |          |
| Automatic scaling                                       | disable 🗸 |          |
| Media utilization scale in threshold                    | 250       |          |
| Media utilization scale out threshold                   | 100       | Ŧ        |
| Modify Cancel   |           |          |

Figure 3-35: Modifying Stack Configuration

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.

| Advanced Config <sup>(3</sup>           |  |  |
|---|--|--|
| Comment                                 |  |  |
| Comment                                 | 5  |  |
|   |  |  |
| <sup>1)</sup> change of parameter requi | res "Update" command                                 |  |
| change of parameter requi               | res "Update" command and causes service interruption |  |

Figure 3-36: Modify Screen Footnote





### 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 |
|--|--------|-------|----------|---------|---------|-------|
|--|--------|-------|----------|---------|---------|-------|

| stack_mgr          | Stacks Configuration Logs About    |   |
|--------------------|------------------------------------|---|
| Start Stop         | 🏶 Heal 🔷 Scale Out 🔽 Scale In 💿 Sc | cale To 🔀 Modify 🔀 Update 🔤 🚥 More 🎽 菌 Delete     |
|                    |                                    | ce1   |
|                    | General                            | Automatic Scaling                                 |
| Name               | ce1                                | Automatic scaling disable                         |
| Туре               | 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          |
|                    |                                    | Automatic scaling scale-in step 1                 |
| Created On Nov     | 21, 2019 11:41:12                  | Automatic scaling scale-out step 1                |
| Started On Nov     | 21, 2019 11:41:12                  |   |
|                    |                                    |   |
| Minimum number     | of media components 2              |   |
| Maximum number     | of media components 6              | Media Components                                  |
| Undate needed      | true                               | Instance type r4 large                            |
| Modified parameter |                                    | Image ID  |
| woumen paramete    | ers max_mc_num                     | Profile forwarding                                |

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

Figure 3-39: Update Screen

| Update   |
|--|
| How do you want to perform the update?<br>(Hitless update minimizes service interruption, but may be significantly slower)<br>Regular update 💙 |
| Update 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.

| C | CC stack_mgr Stacks Configuration Logs About  |  |        |                                    |                  | Logout |
|---|---|--|--------|------------------------------------|------------------|--------|
|   | Start Stop  | Heal Scale Out Scale In O Scale To                         | Modify | 🗸 🔤 🏎 More 👻 💼                     | Delete           |        |
|   | Updating stack<br>Initializing AWS client.<br>Checking that configu<br>Updating signaling co<br>Updating media comp | done<br>ration is allowed done<br>mponents done<br>vonents |        |                                    |                  |        |
|   |   |  | ce1    |                                    |                  |        |
|   |   | General  |        | Au                                 | tomatic Scaling  |        |
|   | Name  | ce1  |        | Automatic scaling                  | disable          |        |
|   | Туре  | Mediant CE   |        | Media utilization scale in thresh  | old > 250% free  |        |
|   | Environment   | AWS  |        | Media utilization scale out thres  | hold < 100% free |        |
|   | State   | updating   |        | DSP utilization scale in threshold | disabled         |        |
|   | IP Address  | 3.124.221.82   |        | DSP utilization scale out thresho  | ld 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<br>Environment | Requires<br>Update | Service<br>Affecting |
|------------|---------------------------------------|---------------------------|--------------------|----------------------|
| General    | Minimum number of media components    | All                       | No                 | No                   |
|            | Maximum number of media components    | All                       | Yes                | No                   |
| Automatic  | Automatic scaling                     | All                       | No                 | No                   |
| Scaling    | 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                   |

| Group Name              | Parameter                        | Applicable<br>Environment | Requires<br>Update | Service<br>Affecting |
|-------------------------|----------------------------------|---------------------------|--------------------|----------------------|
|                         | Automatic scaling scale-out step | All                       | No                 | No                   |
| Automatic<br>Healing    | Automatic healing                | AWS, Azure,<br>Google     | No                 | No                   |
|                         | Automatic healing interval       | AWS, Azure,<br>Google     | No                 | No                   |
| Signaling<br>Components | Number of network interfaces     | AWS, Azure,<br>Google     | Yes                | Yes <sup>(1)</sup>   |
|                         | Interfaces with public IP        | AWS, Azure,<br>Google     | Yes                | Yes <sup>(1)</sup>   |
|                         | Interfaces with additional IP    | AWS, Azure,<br>Google     | Yes                | Yes <sup>(1)</sup>   |
|                         | Management ports                 | AWS, Azure,<br>Google     | Yes                | No                   |
|                         | Signaling ports                  | AWS, Azure,<br>Google     | Yes                | No                   |
|                         | Media ports                      | AWS, Azure,<br>Google     | Yes                | No                   |
|                         | Use main subnet for              | AWS, Azure,<br>Google     | Yes                | No                   |
|                         | Instance type                    | AWS, Azure,<br>Google     | Yes                | Yes <sup>(1)</sup>   |
|                         | Image ID                         | AWS, Azure,<br>Google     | Yes                | Yes <sup>(1)</sup>   |
| Media<br>Components     | Number of network interfaces     | AWS, Azure,<br>Google     | Yes                | Yes (1)              |
|                         | Interfaces with public IP        | AWS, Azure,<br>Google     | Yes                | Yes (1)              |
|                         | Interfaces with additional IP    | AWS, Azure,<br>Google     | Yes                | Yes (1)              |
|                         | Media ports                      | AWS, Azure,<br>Google     | Yes                | No                   |
|                         | Profile                          | AWS, Azure,<br>Google     | Yes                | Yes (1)              |
|                         | Instance type                    | AWS, Azure,<br>Google     | Yes                | Yes (1)              |
|                         | Image ID                         | AWS, Azure,<br>Google     | Yes                | Yes <sup>(1)</sup>   |
| Network<br>Subnets      | Additional 1 subnet              | AWS, Azure,<br>Google     | No <sup>(2)</sup>  | No                   |
|                         | Additional 2 subnet              | AWS, Azure,<br>Google     | No <sup>(2)</sup>  | No                   |
| Advanced                | OS version                       | AWS, Azure                | Yes                | Yes (3)              |



| Group Name Parameter |                 | Applicable<br>Environment | Requires<br>Update | Service<br>Affecting |
|----------------------|-----------------|---------------------------|--------------------|----------------------|
|                      | Advanced config | All                       | Yes (4)            | Yes (1)              |
|                      | Comments        | All                       | No                 | No                   |

<sup>(1)</sup> In AWS and Azure environments, service interruption can be minimized by choosing the "Hitless update" mode in the **Update** operation.

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

<sup>(3)</sup> 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.

<sup>(4)</sup> 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<br>Environment | Requires<br>Update | Service<br>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 <sup>(2)</sup>  | No                   |
|            | Additional 2 subnet           | All                       | No <sup>(2)</sup>  | No                   |
| Automatic  | Automatic healing             | All                       | No                 | No                   |
| Healing    | 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                   |

<sup>(1)</sup> In AWS and Azure environments, service interruption can be minimized by choosing the "Hitless update" mode in the **Update** operation.

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

<sup>(3)</sup> 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.

<sup>(4)</sup> 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

| Stop                                   |
|--|
| Do you really want to stop this stack? |
| Stop Cancel                            |

# 3.16 Rebooting Stack Components

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

Figure 3-42: Rebooting Stack

| Reboot   |
|--|
| Which components do you want to reboot?                              |
| e.g. mc-1,mc-2 or sc,mc  |
| Syntax: comma-separated list of 'sc', 'mc', 'sc-1', 'sc-2' or 'mc-X' |
| Reboot Cancel  |

# 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.

| α | stack_mgr   | Stacks Configuration Logs About |  | Logout |  |  |
|---|---|---------------------------------|--|--------|--|--|
|   | Start       Stop       Heal       Scale Out       Scale To       Image: Modify       Update       Image: More       Image: Delete         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 'up' done         Checking all media components that should be 'up' done       Stopping components 'mc-5, mc-4, mc-3'' |                                 |  |        |  |  |
|   |   |                                 | ce1  |        |  |  |
|   |   | General                         | Automatic Scaling                                |        |  |  |
|   | Name  | ce1                             | Automatic scaling disable                        |        |  |  |
|   | Туре  | Mediant CE                      | Media utilization scale in threshold > 250% free |        |  |  |

Figure 3-43: Healing Stack

**Automatic Healing** 

3.124.221.82

AWS

runnina

Environment

IP Address

Manage via HTTPS enable

State

3.17.1

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

Media utilization scale out threshold < 100% free

disabled

disabled

900 sec

DSP utilization scale in threshold

DSP utilization scale out threshold

Automatic scaling cool down time

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 | а | Stack |  |
|--------|-------|----------|---|-------|--|
|        |       |          |   |       |  |

| Delete                                   |
|--|
| Do you really want to delete this stack? |
| Delete Cancel                            |

# 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

| Rebuild  |
|--|
| Which components do you want to rebuild?   |
| e.g. mc-1,mc-2 or sc,mc  |
| Notes:   |
| * if you rebuild signaling components, their local license will be lost                |
| * if you rebuild both SCs, TLS contexts configuration and auxiliary files will be lost |
| Rebuild Cancel   |

# 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.



**Note:** 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

| Cstack_mgr Stacks Users Config           | guration Logs | Files About           |          |        | aud     | c Logout |
|--|---------------|-----------------------|----------|--------|---------|----------|
| + Add file                               |               |                       |          |        | Search  |          |
|  |               | Files                 |          |        |         |          |
| Name                                     | Туре          | 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



**Note:** 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

| Upgrade  |
|--|
| Software (CMP) URL   |
| https://sbcwestus2.blob.core.windows.net/pub/HostedTP_TEST.cmp                       |
| none 💙   |
| Which components do you want to upgrade?   |
| sc,mc  |
| Syntax: 'sc', 'mc' or 'sc,mc'<br>Graceful timeout for media components upgrade (sec) |
| 60   |
|  |
| Upgrade Cancel   |



**Note:** 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.

**Note:** 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.
- 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



**Note:** 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

| Shelve  |
|---|
| Do you really want minimize stack footprint by deleting most of its components? |
| Shelve Cancel   |

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.

| Reset Password         |  |
|------------------------|--|
| Username<br>e.g. Admin |  |
|                        |  |
| Reset Password Cancel  |  |

Figure 3-50: Resetting Stack Password

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.

| Send INI File   |
|---|
| Send incremental INI file to the SBC Choose File No file chosen |
| Send Cancel   |

Figure 3-51: Sending INI File

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.

|                        | AWS Services - Resource G                  | oups 🗸 🛠                     |                 | 🗘 Alex Agranov 👻 Frankfurt 👻 Support 👻 |  |  |
|------------------------|--|------------------------------|-----------------|--|--|--|
| 0                      | CloudFormation      Stacks                 |                              |                 |  |  |  |
| Cr                     | Create Stack   Actions  Design template  C |                              |                 |  |  |  |
| Filter: Active  stack1 |  |                              |                 |  |  |  |
|                        | 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        |  |  |

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

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.



**Note:** 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 = Ce1ResourceGroup/ScPublicIP
    public_ip_mc-1_eth1 = Ce1ResourceGroup/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 predefined 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_mc-1_eth0 = 172.31.128.101
private_ip_mc-1_eth1 = 172.31.68.101
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 <u>Mediant VE SBC for</u> <u>AWS Installation Manual</u> or <u>Mediant CE SBC for AWS Installation Manual</u> 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.



**Note:** 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.



This page is intentionally left blank.

# 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

\$ stack mgr --help

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:

```
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.

| \$ <b>stack_mgr listhelp</b><br>usage: stack_mgr list [-h] [no-status]  |                              |                  |                                |                                    |  |
|---|------------------------------|------------------|--------------------------------|------------------------------------|--|
| optional arguments:<br>-h,help show this help message and exit<br>no-status do not show real-time status<br>\$ stack_mgr list |                              |                  |                                |                                    |  |
| name  | type                         | vim              | state                          | ip                                 |  |
| stack1  <br>  stack2  <br>+   | sbc-cluster<br>  sbc-cluster | azure<br>  azure | e   running<br>e   scaling-out | 51.143.59.195  <br>  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.



**Note:** 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:

## **C**audiocodes

```
+---+
| # | 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
              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:  $\boldsymbol{1}$ 

```
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
                         1
| 2 | sbc-ssh-key
                         1
| 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. +---+ | name | cidr range | avail zone | | # | id | 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. | name | cidr range | avail zone | | # | id +---+ | 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



**Note:** 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:



**Note:** 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
                         - assign two additional private IP
#
   "main:2"
                           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
```

## **C** audiocodes

```
# 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

## **C** audiocodes

```
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=sbc,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,
sc1, 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
```



**Note:** 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 <a href="https://www.audiocodes.com/library/technical-documents.">https://www.audiocodes.com/library/technical-documents.</a>

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 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.

```
-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 |
      | additional1 |
| eth2
                         | 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 | | additional2 | eth3 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 |
                             | - | m5.large |
| - | m5.large |
                       | -
| mc-4 | 172.31.75.108 | down
                       | -
| mc-5 | 172.31.67.179 | down
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 | ethl           |                        | 20.115.200.133    |
| sbc-1<br> | eth1<br>eth1:1 | 10.9.1.4<br>  10.9.1.5 | 20.99.165.170     |
| sbc-2<br> | eth1<br>eth1:1 | 10.9.1.6<br>  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).
```
<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>



**Note:** 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.

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:

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
```

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:
                        show this help message and exit
  -h, --help
  --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.



**Note:** The *Update* operation might be service affecting cause. It's therefore recommended to run it during periods of maintenance.

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
```

# **C**audiocodes

Updating media components... done Updating SBC cluster configuration... done Wait for new configuration to be applied..... done

.... 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.

The stop command can also be used to stop specific components by using the --ids argument. This option is primarily used for debugging.

Stopping stack components.....

### 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.

```
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.

```
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 dele- --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.

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' from SBC configuration... done
Removing media components 'mc-2' to 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.



**Note:** 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
              file name
  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.

- 1. Copy CMP file to Stack Manager using the SCP/SFTP protocol.
- 2. Add the copied file to the Files Repository using the **stack\_mgr file-add** command.
- 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



**Note:** 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...

```
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:

# 4.20 Sending INI File

You can update configuration of the deployed Mediant VE / CE stack via the **send-ini** command:

# 4.21 Multiple Operations

'scaling-in')

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
```

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    | /api/v1/stacks                                      | list stacks                 |
| GET    | /api/v1/stacks/ <stack_name></stack_name>           | show stack                  |
| POST   | /api/v1/stacks/ <stack_name></stack_name>           | create stack                |
| DELETE | /api/v1/stacks/ <stack_name></stack_name>           | delete stack                |
| PUT    | /api/v1/stacks/ <stack_name></stack_name>           | modify stack                |
| PURGE  | /api/v1/stacks/ <stack_name></stack_name>           | purge stack                 |
| POST   | /api/v1/stacks/ <stack_name>/heal</stack_name>      | heal stack                  |
| POST   | /api/v1/stacks/ <stack_name>/scale-in</stack_name>  | scale-in stack              |
| POST   | /api/v1/stacks/ <stack_name>/scale-out</stack_name> | scale-out stack             |
| POST   | /api/v1/stacks/ <stack_name>/scale</stack_name>     | scale stack                 |
| POST   | /api/v1/stacks/ <stack_name>/update</stack_name>    | update stack                |
| GET    | /api/v1/config                                      | get global configuration    |
| PUT    | /api/v1/config                                      | update global configuration |

# 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.

```
200 OK
Content-Type: application/json
{
    "status": "in_progress",
    "output": "Removing SBC media components... "
}
```

```
GET /api/v1/tasks/1
```

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                 |
|             |                                     |



Note: 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 <u>https://portal.azure.com/</u>.
- 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.
- 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..."
}
```

 $\succ$ To access Stack Manager's REST API using access token:

Include the access token in the Authorization header when accessing Stack Manager's 1. **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/v   | 1   |
| 200 07       |   |
| 200 OK       |   |
| Content-Ty   | pe: 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"                  |
| }            |   |
| ,            |   |
| L            |   |

# 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_expiration_text": "never",
            "password_updated_at": 1722117527,
            "status": "active",
            "type": "sec-admin",
            "updated_at": 0
        }
    ]
}
```

#### 5.5.2 Adding User

| Met  | hod:          | POST  |
|------|---------------|---|
| Path | 1:            | /api/v1/users   |
| Arg  | uments:       | none  |
| Con  | tent:         |   |
|      | name          | username  |
|      | password      | password  |
|      | type          | "sec-admin" (default)   "admin"   "operator"   "monitor"                          |
|      | status        | "active" (active)   "change-password"   "locked"                                  |
|      | password_expi | <i>ration</i> number of days for password expiration or 0 (unlimited); default: 0 |
| Con  | tent type:    | application/json  |
| Des  | cription:     | 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

| Met                | hod:          | PUT   |
|--------------------|---------------|---|
| Path               | 1:            | /api/v1/users   |
| Arg                | uments:       | none  |
| Con                | tent:         |   |
|                    | name          | username  |
|                    | new_name      | new user name; default: unchanged   |
|                    | password      | password; default: unchanged  |
|                    | type          | "sec-admin" (default)   "admin"   "operator"   "monitor"                  |
|                    | status        | "active" (default)   "change-password"   "locked"                         |
|                    | password_expi | ation number of days for password expiration or 0 (unlimited); default: 0 |
| Content type: appl |               | application/json  |
| Description:       |               | Modifies existing user.   |

```
PUT /api/vl/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:                                  |                  |  |
| name                                      | username         |  |
| Content type:                             | application/json |  |
| Description:                              | Deletes user.    |  |
|   |                  |  |
| DELETE /api/v1/files                      |                  |  |
| <pre>Content-Type: application/json</pre> |                  |  |
| {   |                  |  |

```
"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/con | fig  |

```
200 OK
Content-Type: application/json
{
    "aws_access_key": "ABCDEDFGHIJKLMN",
    "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": "ABCDEDFGHIJKLMN",
    "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 % $ \label{eq:result} \end{tabular} \left( \begin{array}{c} \end{tabular} \end{tabular} \right) = \left( \end{tabular} \end{tabular} \end{tabular} \right) = \left( \end{tabular} \end{tabular} \right) = \left( \end{tabular} \end{tabular} \end{tabular} \right) = \left( \end{tabular} tabular$ |

```
GET /api/v1/stacks
```

```
200 OK
Content-Type: application/json
{
    "stacks": [
        {
            "created_at": "Mar 14, 2021 16:59:15",
            "deleted_at": "",
            "id": "stack1",
            "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></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.<br>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: |  |  |
|            | <ul> <li>type – stack type; supported values:</li> </ul> |  |
|            | o sbc – Mediant VE                                       |  |
|            | <ul> <li>sbc-cluster – Mediant CE</li> </ul>             |  |
|            | <ul> <li>vaic – Voice.Al Connect</li> </ul>              |  |
|            | <ul> <li>win-server – Windows Server</li> </ul>          |  |
|            |  |  |

• vim - virtual environment; supported values:

- o aws Amazon Web Services
- o azure Microsoft Azure
- o google Google Cloud Platform
- o 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/ <stac< td=""><td>k_name&gt;</td></stac<>   | k_name>   |
| Arguments:   | ?no_status=True  | Don't include real-time status information<br>(connection status and media/dsp utilization)<br>in the response. |
| Description: | Returns detailed information of the specific stack.<br>Unless <i>?no-status=True</i> argument is provided, the command queries<br>the active Signaling Component for real-time connection status and<br>media/dsp utilization per Media Component. This might result in a<br>delay in response (up to 30 seconds), for example, if connection with<br>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</stack_name> |
| 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</stack_name>                         |
| 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-
0872da39clb6112d8, sg-038adbdf823381bf8, sg-0a3b1e2d2f039eec9\n\n-
-----\n..."
}
```

### 5.9.3 Checking Connectivity

| Method:      | POST   |
|--------------|--|
| Path:        | /api/v1/stacks/ <stack_name>/check-connectivity</stack_name> |
| 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</stack_name> |  |
| Description:  | Updates stack's connectivity parameters                       |  |
| Arguments:    | none  |  |
| Content:      |   |  |
| ip            | IP address  |  |
| username      | username  |  |
| password      | 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"
}
```



**Note:** 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</stack_name>    |
| 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</stack_name>          |  |
| Arguments:     |   |  |
| ?num=2         | Defines the number of Media Components to remove            |  |
| ?ids=mc-1,mc-2 | 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</stack_name>              |  |
| 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></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</stack_name>             |  |
| 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/st | acks/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</stack_name>      |  |
| Arguments:     |   |  |
| ?ids=mc-1,mc-3 | 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</stack_name>     |
| Arguments:     |   |
| ?ids=mc-1,mc-3 | 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</stack_name>       |
| Arguments:     |  |
| ?ids=mc-1,mc-3 | 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></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></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</stack_name>         |  |
| Arguments:  | none   |  |
| Description:  | Heals the stack  |  |
| Response:   | URL of asynchronous task (as described in Section 5.2) |  |
| POST /api/v1/stacks/stack1/heal                                     |  |  |
| <pre>202 Accepted Content-Type: application/json {</pre>            |  |  |
| <pre>"description": "task accepted", "url": "/api/v1/tasks/1"</pre> |  |  |

# 5.16 Rebuilding Stack

| Method:        | POST   |  |
|----------------|--|--|
| Path:          | /api/v1/stacks/ <stack_name>/rebuild</stack_name>      |  |
| 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
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

| Meti          | hod:   | POST     |   |  |
|---------------|--|----------|---|--|
| Path: /api/v1 |  | /api/v1/ | /1/stacks/ <stack_name>/upgrade</stack_name>                |  |
| Argı          | uments:  |          |   |  |
|               | ?ids=sc.mc   | Co       | mma-separated list of component types to be rebuilt (sc/mc) |  |
|               | &cmp_url= <ur< th=""><th>L&gt; Pu</th><th>blicly accessible HTTP/HTTPS URL with software load (CMP)</th></ur<> | L> Pu    | blicly accessible HTTP/HTTPS URL with software load (CMP)   |  |
|               | &graceful_timeout=60<br>&mode=(hitless\reset)  |          | Graceful timeout for media components upgrade (in seconds)  |  |
|               |  |          | Mode for Signaling Components upgrade                       |  |
|               |  |          |   |  |

**Response:** URL of asynchronous task (as described in Section 5.2)

```
POST
```

```
/api/v1/stacks/stack1/upgrade?ids=sc,mc&cmp_url=<URL>&graceful_tim
eout=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.
- 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
             file name
  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



**Note:** 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</stack_name>       |
| 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</stack_name>     |
| 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</stack_name>   |
| Arguments:    | none   |
| Content:      |  |
| username      | username   |
| password      | 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"
}
```



**Note:** 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</stack_name>     |
| 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

| stack       | <b>_mgr</b> s    | tacks C   | Configuration Lo    | ogs About Alex Agrano   | v Logout |
|-------------|------------------|-----------|---------------------|---|----------|
| Activity    | Application      | Auto job  | HTTP server         | HTTP access API server  |          |
| 2 Refresh   |                  |           |                     | <b>Lines</b> 100  |          |
| [2022-01-16 | 06:36:35,154679] | [MAJOR]   | Raise 'rest-api' a  | larm on stack 'orenu-ngm' - Cannot connect to SBC via REST API  |          |
| [2022-01-16 | 06:37:40,578501] | [CLEARED] | Clear 'rest-api' a  | larm on stack 'orenu-nqm' - Successfully connected to SBC via REST API                                  |          |
| [2022-01-16 | 06:51:24,914897] | [MAJOR]   | Raise 'rest-api' a  | larm on stack 'orenu-nqm' - Cannot connect to SBC via REST API  |          |
| [2022-01-16 | 06:51:57,999404] | [CLEARED] | Clear 'rest-api' a  | larm 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-a | aws-ve-1' - auto-job, internal  |          |
| [2022-01-17 | 17:35:30,639569] | [DONE]    | Heal stack 'garyd-a | aws-ve-1' - done  |          |
| [2022-01-17 | 17:39:51,538544] | [INFO]    | Heal stack 'garyd-a | aws-ve-1' - Alex Agranov, https, 172.18.110.11  |          |
| [2022-01-17 | 17:40:03,264757] | [DONE]    | Heal stack 'garyd-a | aws-ve-1' - done  |          |
| [2022-01-17 | 17:40:19,843708] | [INFO]    | Stop stack 'garyd-a | aws-ve-1' - Alex Agranov, https, 172.18.110.11  |          |
| [2022-01-17 | 17:42:06,548630] | [DONE]    | Stop stack 'garyd-a | aws-ve-1' - done  |          |
| [2022-01-19 | 12:04:55,710839] | [INFO]    | Create stack 'dmit  | ryh-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 'arie  | :l-test-sbc' - type: Mediant VE - Ariel Mannes, https, 172.18.110.18                                    |          |
| [2022-01-19 | 12:17:13,767672] | [FAILED]  | Create stack 'arie  | :l-test-sbc' - failed   |          |
| [2022-01-19 | 12:17:54,898331] | [INFO]    | Delete stack 'arie  | :l-test-sbc' - Ariel Mannes, https, 172.18.110.18   |          |
| [2022-01-19 | 12:18:37,281247] | [DONE]    | Delete stack 'arie  | :l-test-sbc' - done   |          |
| [2022-01-19 | 12:20:25,795081] | [DONE]    | Create stack 'dmite | ryh-ce-test-1' - done   |          |
| [2022-01-19 | 12:21:00,240188] | [INFO]    | Create stack 'arie  | l-test-sbc' - type: Mediant VE - Ariel Mannes, https, 172.18.110.18                                     |          |
| [2022-01-19 | 12:25:40,557028] | [DONE]    | Create stack 'arie  | 'l-test-sbc' - done   |          |

To view logs through CLI, use the following command:

```
$ stack mgr log --name activity log --lines 10
[2020-12-14 17:58:03] [INFO]
                              Delete stack 'test-ce-2'
                               Delete stack 'test-ce-2' - done
[2020-12-14 18:00:17] [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 creates 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 <code>customEvents</code> 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": "2022-01-26T06:57:48.0000002",
    "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



This page is intentionally left blank.

# 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 unattendedupgrades
- 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.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  ${\tt add\_header}$  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:

- Install the "http-headers" nginx module using the following command: sudo apt install libnginx-mod-http-headers-more-filter
- Add the following after the "server\_tokens off" line in the nginx configuration file:

```
more_clear_headers Server;
```

Refer to Nginx documentation at <a href="http://nginx.org/en/docs">http://nginx.org/en/docs</a> for more information. Restart the Web server via the following command to apply the changes: <a href="systemctl">systemctl</a> 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 <u>https://www.ssh.com/academy/ssh/sshd\_config</u> for more information.

Restart the SSH server via the following command to apply the changes:  ${\tt systemctl}$  restart  ${\tt sshd}$ 

#### **International Headquarters**

6 Ofra Haza Street Naimi Park 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: <u>https://www.audiocodes.com/corporate/offices-worldwide</u> Website: <u>https://www.audiocodes.com</u>

©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-28968