INSTALLATION RUNBOOK FOR AudioCodes Mediant VE SBC

Application Type:	**Session Border Controller**
Application Version:	**7.2**
MOS Version:	**9.0**
OpenStack version:	**Mitaka**

Content

Document History

Version	Revision Date	Description
0.1	07-08-2016	Initial Version
0.2	28-08-2016	Updated after Mirantis team review

1 Introduction

This document provides detailed instructions for deploying AudioCodes Mediant VE Session Border Controller (SBC) v7.2 on Mirantis OpenStack v9.0, including reference architecture, installation sequence, pre- and post- installation configuration, limitations and testing procedures.

1.1 Target Audience

The target audience of this document is devops or IT responsible for deploying and administering AudioCodes Mediant VE SBC on Mirantis OpenStackplatform.

2 Product Overview

AudioCodes Mediant VE Session Border Controller (SBC) is a versatile Voice over IP communication platform that interconnects different VoIP networks and equipment.
It is typically deployed at the border between the enterprise and the service provider. In the enterprise environment, SBC forms an effective demarcation point between the business's VoIP network and the service provider's SIP Trunk, performing SIP protocol mediation and media handling (interoperability) and securing the enterprise VoIP network. In the service provider core, SBC provides security and protocol normalization.

Mediant VE is virtual edition of AudioCodes Mediant Session Border Controllers (SBC) family of products that runs as a virtual appliance on top of the commodity server hardware. It is suitable for both Enterprise and Service Provider deployments and provides the following features and benefits:

Benefits

- Meets demands for data center infrastructure harmonization and NFV
- Certified by Miercom for high performance and scalability under security attacks including Denial of Service, malformed SIP messages and rogue RTP packets
- Offers comprehensive interoperability and enhanced voice quality
- Deployable on private and public clouds such as OpenStack and Amazon Web Services (AWS)
- Proven interoperability with 3rd party NFV orchestration solutions
- Rapidly scale session capacity and quickly deploy new instances with AudioCodes' cloud licensing

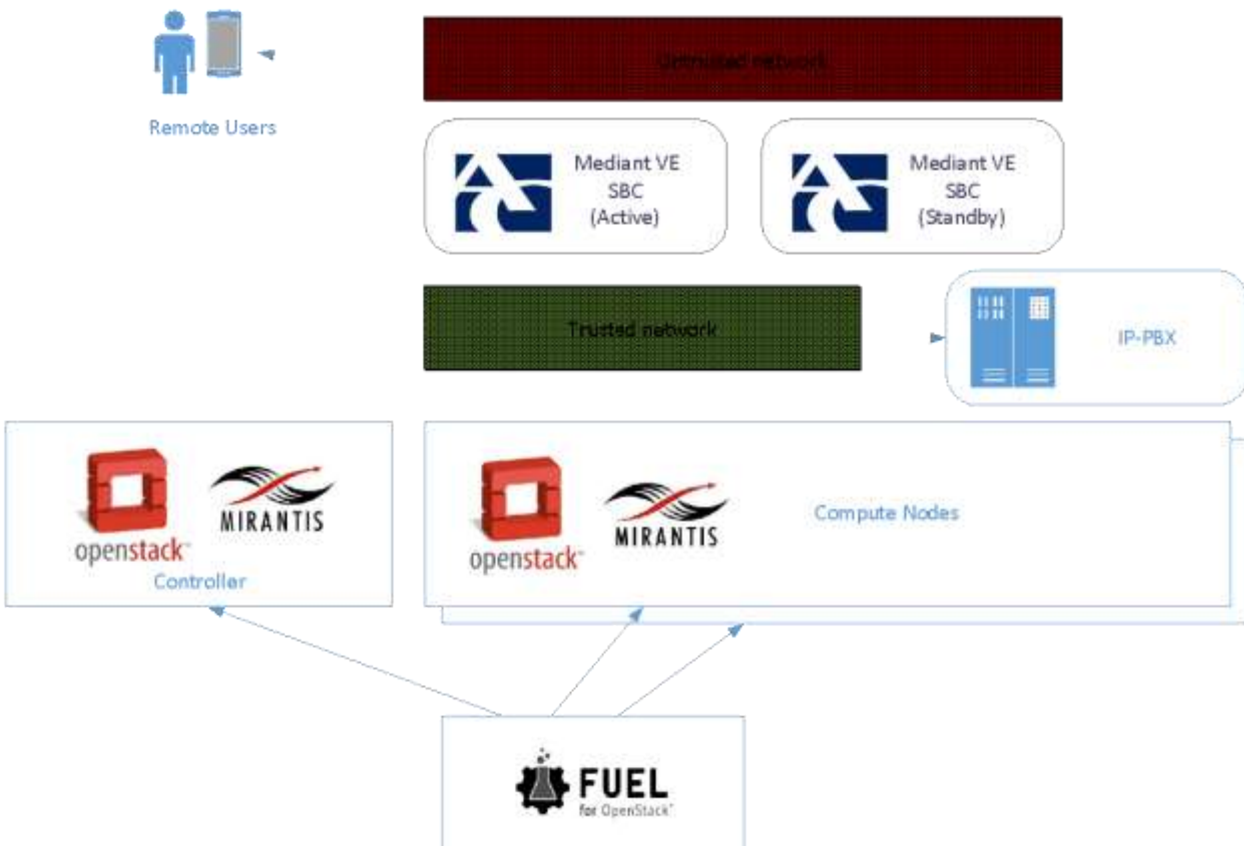
Features

- Same code base as AudioCodes field-proven hardware-based SBCs
- Runs as a VNF in an NFV environment both on a virtual CPE and within service provider datacenters
- Runs on dedicated COTS servers and in virtualized environments
- High packet throughput through optimized network path
- Advanced voice quality monitoring and reporting
- Built-in media transcoding capability in the software
- Qualified for Microsoft Skype for Business/Lync and BroadSoft BroadWorks environments
- Embedded signaling and media encryption hardware
- Media replication for recording through SIPREC
- High-availability 1:1 active-standby configuration ensures business continuity

3 Joint Reference Architecture

Mediant VE may be deployed on top of Mirantis OpenStack platform – as part of NFV environment within service provider data center.

The following diagram shows typical architecture of such deployment.



Mediant VE SBC is typically connected to trusted and untrusted network and mediates the VoIP traffic (both signaling and media) between the two. It is common to deploy two SBC instances – in 1+1 HA configuration – to achieve non-traffic-affecting solution resiliency in case of failure.

Use of Mirantis OpenStack as "NFV infrastructure" provides access to latest technologies that enable optimal performance of VNFs – such as flat provider networks, SR-IOV, NUMA/CPU pinning, guaranteed resource allocation etc. When Mediant VE SBC runs in such environment it achieves significantly better performance, compared to "generic" (non-optimized) private and public cloud environments. Thus providing much more efficient use of hardware resources and reducing overall solution cost.

4 Physical & Logical Network Topology

For the purposes of certification Mirantis OpenStack was installed on 3 physical nodes:

- node 1 – Fuel Master
- node 2 – OpenStack Controller + Cinder
- node 3 – OpenStack Compute

The following physical and logical network topologies were created during default installation:

Figure 1: Physical Topology

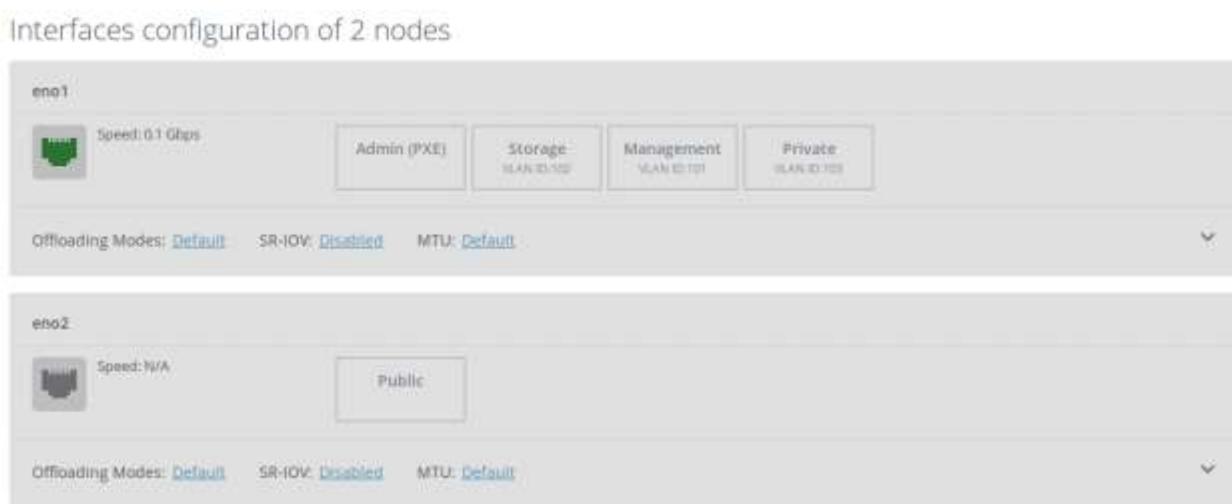
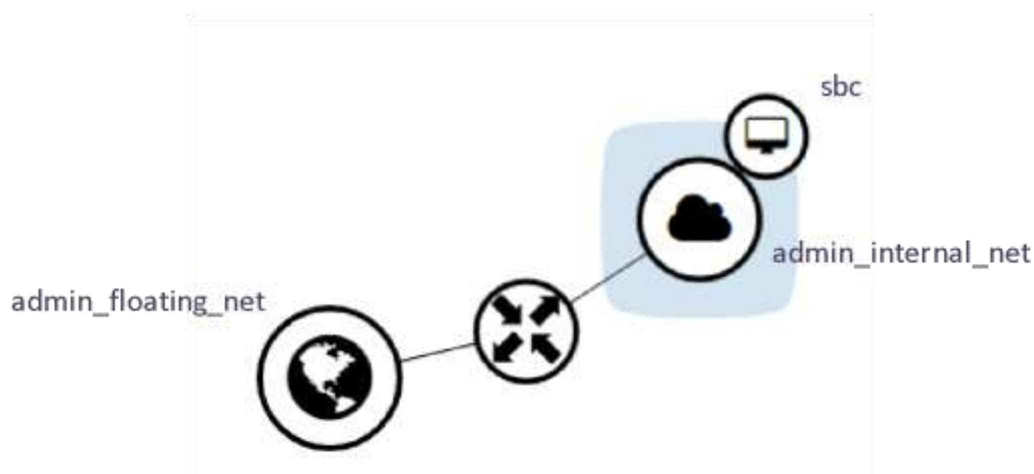


Figure 2: Logical Topology



Default logical topology corresponds to the following Neutron configuration:

```
# neutron net-show admin_internal_net
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| availability_zone_hints   |                                      |
| availability_zones        | nova                                 |
| created_at                | 2016-08-04T10:55:15                  |
| description               |                                      |
| id                        | 6db9469e-5b6c-4665-9b43-a774bae0f363 |
| ipv4_address_scope        |                                      |
| ipv6_address_scope        |                                      |
| mtu                       | 1450                                 |
| name                      | admin_internal_net                   |
| port_security_enabled     | True                                 |
| provider:network_type     | vxlan                                |
| provider:physical_network |                                      |
| provider:segmentation_id  | 2                                    |
| router:external           | False                                |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   | 34b1aa9f-c387-4529-b450-bfe1ff3c60f0 |
| tags                      |                                      |
| tenant_id                 | 7e80494e015b4e48bfd795197d70b929     |
| updated_at                | 2016-08-04T10:55:15                  |
+---------------------------+--------------------------------------+
```

```
# neutron subnet-show admin_internal_net__subnet
+-------------------+------------------------------------------------------+
| Field             | Value                                                |
+-------------------+------------------------------------------------------+
| allocation_pools  | {"start": "192.168.111.2", "end": "192.168.111.254"} |
| cidr              | 192.168.111.0/24                                     |
| created_at        | 2016-08-04T10:55:17                                  |
| description       |                                                      |
| dns_nameservers   | 8.8.4.4                                              |
|                   | 8.8.8.8                                              |
| enable_dhcp       | True                                                 |
| gateway_ip        | 192.168.111.1                                        |
| host_routes       |                                                      |
| id                | 34b1aa9f-c387-4529-b450-bfe1ff3c60f0                 |
| ip_version        | 4                                                    |
| ipv6_address_mode |                                                      |
| ipv6_ra_mode      |                                                      |
| name              | admin_internal_net__subnet                           |
| network_id        | 6db9469e-5b6c-4665-9b43-a774bae0f363                 |
| subnetpool_id     |                                                      |
```

```
| tenant_id          | 7e80494e015b4e48bfd795197d70b929                      |
| updated_at         | 2016-08-04T10:55:17                                   |
+--------------------+-------------------------------------------------------+
```

```
# neutron net-show admin_floating_net
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| availability_zone_hints   |                                      |
| availability_zones        | nova                                 |
| created_at                | 2016-08-04T10:55:13                  |
| description               |                                      |
| id                        | 01495bcb-41b5-446c-8a66-8b8d459e7bc3 |
| ipv4_address_scope        |                                      |
| ipv6_address_scope        |                                      |
| is_default                | False                                |
| mtu                       | 1500                                 |
| name                      | admin_floating_net                   |
| port_security_enabled     | True                                 |
| provider:network_type     | flat                                 |
| provider:physical_network | physnet1                             |
| provider:segmentation_id  |                                      |
| router:external           | True                                 |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   | 40873d8b-9263-47a7-97bd-66583302e87c |
| tags                      |                                      |
| tenant_id                 | 7e80494e015b4e48bfd795197d70b929     |
| updated_at                | 2016-08-04T10:55:13                  |
+---------------------------+--------------------------------------+
```

```
# neutron subnet-show admin_floating_net__subnet
+--------------------+----------------------------------------------------+
| Field              | Value                                              |
+--------------------+----------------------------------------------------+
| allocation_pools   | {"start": "172.16.0.130", "end": "172.16.0.254"} |
| cidr               | 172.16.0.0/24                                      |
| created_at         | 2016-08-04T10:55:19                                |
| description        |                                                    |
| dns_nameservers    |                                                    |
| enable_dhcp        | False                                              |
| gateway_ip         | 172.16.0.1                                         |
| host_routes        |                                                    |
| id                 | 40873d8b-9263-47a7-97bd-66583302e87c               |
| ip_version         | 4                                                  |
| ipv6_address_mode  |                                                    |
| ipv6_ra_mode       |                                                    |
| name               | admin_floating_net__subnet                         |
```

```
| network_id        | 01495bcb-41b5-446c-8a66-8b8d459e7bc3                 |
| subnetpool_id     |                                                      |
| tenant_id         | 7e80494e015b4e48bfd795197d70b929                     |
| updated_at        | 2016-08-04T10:55:19                                  |
+-------------------+------------------------------------------------------+
```

4.1 VXLAN Overlay Network Topology

Default logical topology created during Mirantis OpenStack installation creates tenant-specific *admin_internal_net* network that uses VXLAN tunneling overlay. Instances are deployed in the *admin_internal_net* network and are assigned with IP addresses that are local to the OpenStack domain. In order to expose instances to the "outside world" floating IPs (from the *admin_floating_net*) must be allocated and assigned to the instance.

When Mediant VE SBC is deployed in such topology and needs to communicate with the equipment located outside the OpenStack network (via the floating IP address) it is necessary to configure *NAT Translation* configuration table in the SBC, so that it would be able to properly adjust IP addresses in signaling and media traffic that traverses it.

Figure 3: NAT Translation Table

```
[ NATTranslation ]
FORMAT NATTranslation_Index = NATTranslation_SourceIPInterfaceName,
NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort,
NATTranslation_SourceEndPort, NATTranslation_TargetStartPort,
NATTranslation_TargetEndPort;
NATTranslation 1 = NET1, 193.4.2.11, 6000, 65535, , ;
NATTranslation 2 = NET1, 193.4.2.11, 5060, 5060, , ;
[ NATTranslation ]
```

The 1st line in the above example corresponds to the media (RTP) traffic and the 2nd line – to the control (SIP) traffic. 193.4.2.11 matches to the floating IP address assigned to the SBC instance.

4.2 Flat Provider Network Topology

While VXLAN networks are very easy to manage, especially for large-scale deployments, there is quite significant performance impact of overlay itself and NAT translation that inherently happens in this topology. In order to reduce this impact (and improve performance of the Mediant VE SBC) it is recommended to use *flat provider* networks – that connect Instances/VMs directly to the external network equipment.

Figure 3: Flat Provider Network

Flat provider network topology corresponds to the following Neutron configuration:

```
# neutron net-show admin_flat_net
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| availability_zone_hints   |                                      |
| availability_zones        | nova                                 |
| created_at                | 2016-08-04T17:18:59                  |
| description               |                                      |
| id                        | c70d07a8-35e2-4415-a8bc-249e7d0404a4 |
| ipv4_address_scope        |                                      |
| ipv6_address_scope        |                                      |
| is_default                | False                                |
| mtu                       | 1500                                 |
| name                      | public                               |
| provider:network_type     | flat                                 |
| provider:physical_network | flat1                                |
| provider:segmentation_id  |                                      |
| router:external           | True                                 |
| shared                    | True                                 |
| status                    | ACTIVE                               |
| subnets                   | 12105653-b705-43c1-9d9c-9644a69c930c |
| tags                      |                                      |
| tenant_id                 | 7e80494e015b4e48bfd795197d70b929     |
| updated_at                | 2016-08-04T17:18:59                  |
+---------------------------+--------------------------------------+
```

```
# neutron subnet-show admin_flat_net__subnet
+-------------------+---------------------------------------------------+
| Field             | Value                                             |
+-------------------+---------------------------------------------------+
| allocation_pools  | {"start": "10.4.219.224", "end": "10.4.219.244"} |
| cidr              | 10.4.0.0/16                                       |
| created_at        | 2016-08-04T15:31:27                               |
| description       |                                                   |
| dns_nameservers   | 10.1.1.11                                         |
|                   | 10.1.1.10                                         |
| enable_dhcp       | True                                              |
```

```
| gateway_ip        | 10.4.0.1                                         |
| host_routes       |                                                  |
| id                | 12105653-b705-43c1-9d9c-9644a69c930c             |
| ip_version        | 4                                                |
| ipv6_address_mode |                                                  |
| ipv6_ra_mode      |                                                  |
| name              | admin_flat_net__subnet                           |
| network_id        | c70d07a8-35e2-4415-a8bc-249e7d0404a4             |
| subnetpool_id     |                                                  |
| tenant_id         | 7e80494e015b4e48bfd795197d70b929                 |
| updated_at        | 2016-08-04T15:31:27                              |
+-------------------+--------------------------------------------------+
```

When Mediant VE SBC is deployed “flat” network topology it is assigned with IP addresses that are directly accessible from outside the OpenStack network. Therefore overall performance is significantly better (compared to the VXLAN topology) and there is no need to configure *NATTranslation* configuration table in the SBC.

4.3 SR-IOV Network Topology

In order to further improve performance of the networking layer in general and Mediant VE SBC in particular it is possible to use SR-IOV network topology. Refer to “Mirantis Network Functions Virtualization Guide” for detailed instructions on how to configure such topology.
Mediant VE SBC is fully SR-IOV capable and has all needed drivers integrated into the image – therefore there is no need to perform any additional configuration on it besides enabling SR-IOV at the infrastructure layer.

5 Installation & Configuration

5.1 Environment preparation

Refer to Mirantis OpenStack 9.0 Documentation for environment setup..
Please be sure to review Fuel Documentation at OpenStack website.

MOS 9.0 ISO is available at Mirantis website.

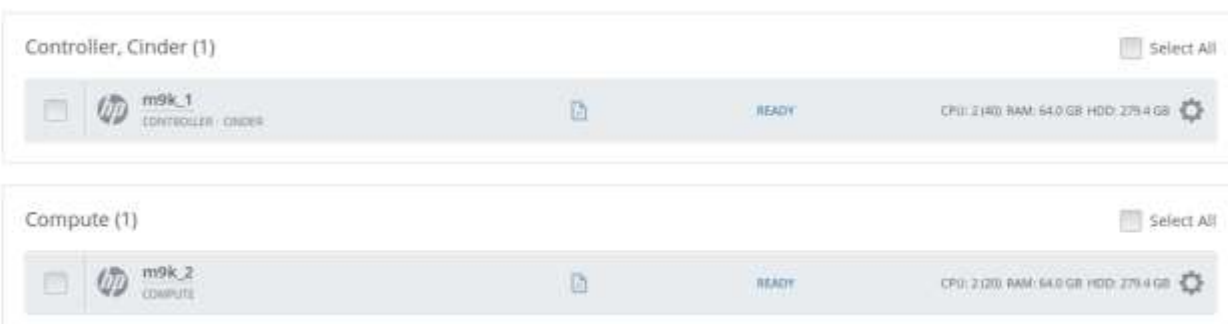
5.2 MOS installation

1. Install Fuel Master node as described in Mirantis OpenStack 9.0 Documentation.

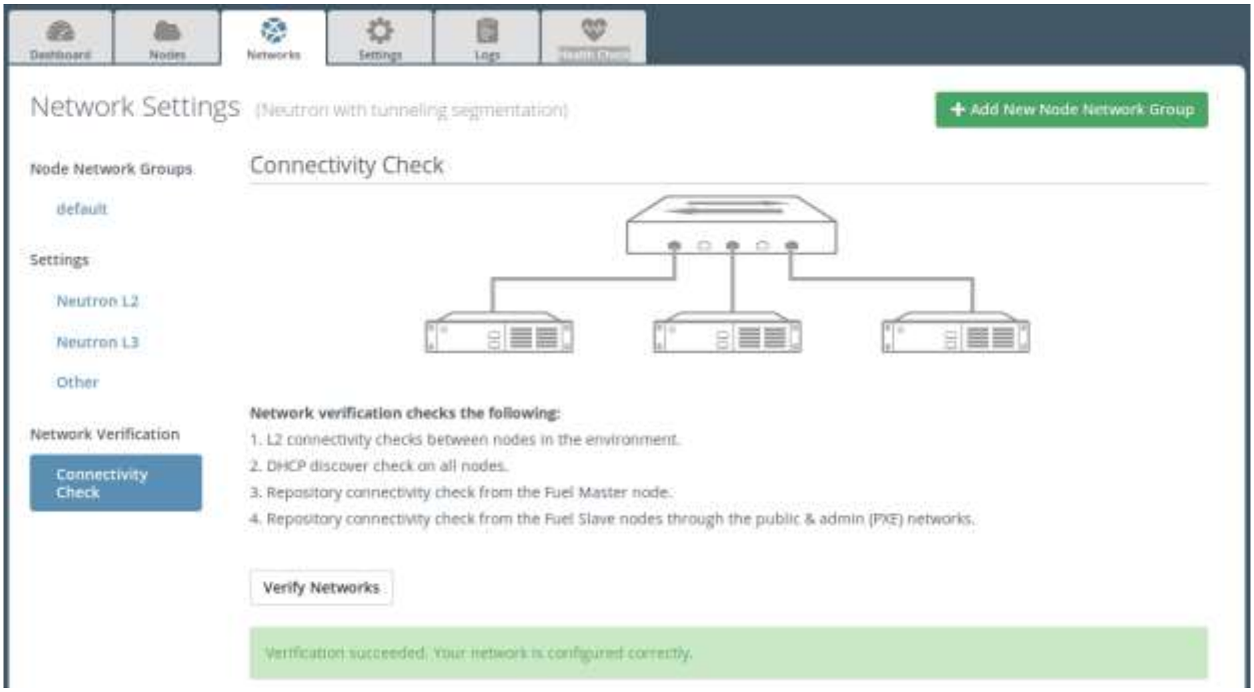
2. Create a new OpenStack Environment with the following configuration:
    - Compute: KVM
    - Network: Neutron with tunneling segmentation

3. Configure Fuel Slave nodes to PXE boot and verify that they are properly detected by Fuel Master node.

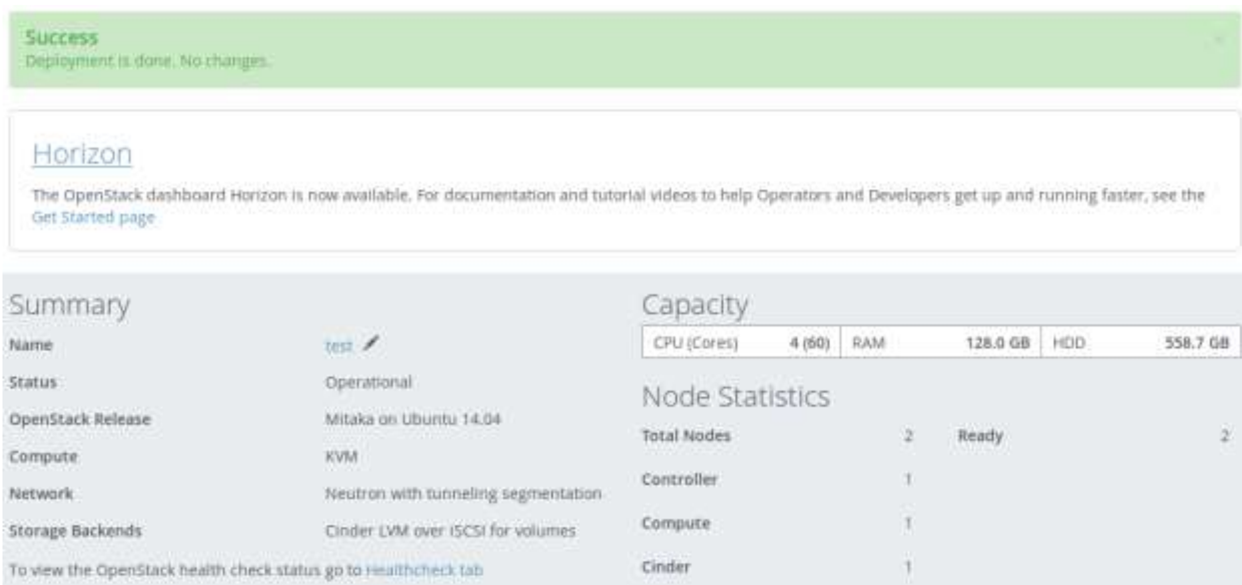
4. Assign roles to Fuel Slave nodes (in the minimal PoC configuration it is possible to have only 2 nodes – as shown in the picture below).



5. Run Connectivity Check to ensure validity of the network setup.

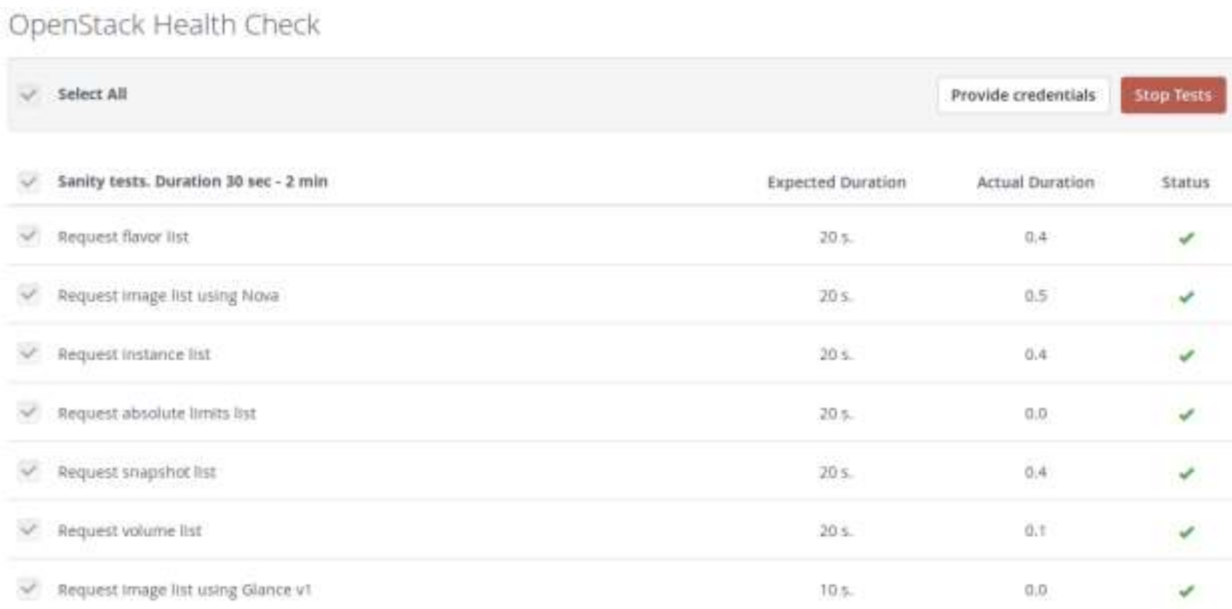


6. Deploy the configured OpenStack Environment.



5.2.1 Health Check Results

The deployment should pass for basic sanity test, functional test and HA test if controllers are configured in HA mode.



OpenStack Health Check

Select All | Provide credentials | Stop Tests

Sanity tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
Request flavor list	20 s.	0.4	✔
Request image list using Nova	20 s.	0.5	✔
Request instance list	20 s.	0.4	✔
Request absolute limits list	20 s.	0.0	✔
Request snapshot list	20 s.	0.4	✔
Request volume list	20 s.	0.1	✔
Request image list using Glance v1	10 s.	0.0	✔

5.3 MOS Preparation for Mediant VE SBC Installation

Login to the OpenStack dashboard (Horizon) on the Controller Node.

In Admin > System > Flavors create the following new flavors for Mediant VE SBC:

Flavors

Filter | + Create Flavor | Delete Flavors

FLAVOR NAME	VCPUS	RAM	ROOT DISK	EPHEMERAL DISK	SWAP DISK	RX/TX FACTOR	ID	PUBLIC	METADATA	ACTIONS
sbc.large	4	16GB	10GB	0GB	0MB	1.0	95aee8f9-8cdd-4575-968c-d4f3d2526252	Yes	No	Edit Flavor
sbc.small	1	2GB	10GB	0GB	0MB	1.0	e6bac708-de96-4ccb-97a9-7d92bca9f495	Yes	No	Edit Flavor

In Project > Compute > Access & Security > Security Groups add the following rules to the **default** security group:

- allow inbound SSH, HTTP and HTTPS traffic
- allow inbound media traffic (UDP ports 6000-65535)

- allow inbound SIP control traffic (UDP port 5060)

DIRECTION	ETHER TYPE	IP PROTOCOL	PORT RANGE	REMOTE IP PREFIX	REMOTE SECURITY GROUP	ACTIONS
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	Any	Any	-	default	Delete Rule
Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	443 (HTTPS)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	UDP	6000 - 65535	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	UDP	5060	0.0.0.0/0	-	Delete Rule

Contact AudioCodes and receive Mediant VE SBC QCOW2 image. Upload the image to the Mirantis OpenStack environment.

Images

Project (1) | Shared with Me (0) | Public (2) | + Create Image | Delete Images

IMAGE NAME	TYPE	STATUS	PUBLIC	PROTECTED	FORMAT	SIZE	ACTIONS
sbc_7.20A.001	Image	Active	Yes	No	QCOW2	1.9 GB	Launch

Login to the CLI interface on the Controller Node.
Create **admin_flat_net** network and corresponding subnet.for flat provider network topology.

```
<on controller node>

# . keystonerc
# neutron net-create admin_flat_net --shared --provider:network_type flat \
          --provider:physical_network flat1
# neutron subnet-create --name admin_flat_net___subnet --gateway 10.4.0.1 \
          --dns-nameserver 10.1.1.11 --allocation-pool \
          start=10.4.219.224,end=10.4.219.244 admin_flat_net 10.4.0.0/16
```

Login to the CLI interface on the Compute Node(s).
Configure OVS bridge and create proper mapping for flat provider network.

```
<on compute node>
# vi /etc/network/interfaces.d/ifcfg-br-ex1
auto br-ex1
allow-ovs br-ex1
```

```
iface br-ex1 inet manual
  ovs_type OVSBridge
  ovs_ports ens1f0

allow-br-phys ens1f0
iface ens1f0 inet manual
  ovs_bridge br-ex1
  ovs_type OVSPort

# ifup br-ex1
# ifup ens1f0

# vi /etc/neutron/plugins/ml2/openvswitch_agent.ini
  bridge_mappings = flat1:br-ex1

# restart neutron-openvswitch-agent
```

5.4 Mediant VE SBC Installation Steps

Login to the OpenStack dashboard (Horizon) on the Controller Node.

In Project > Compute > Instances launch a new instance.
Choose Mediant VE SBC image – **sbc_7.20A.001** – created in the previous step.
Choose one of the Mediant VE SBC specific flavors – **sbc.small** or **sbc.large** – created in the previous step.

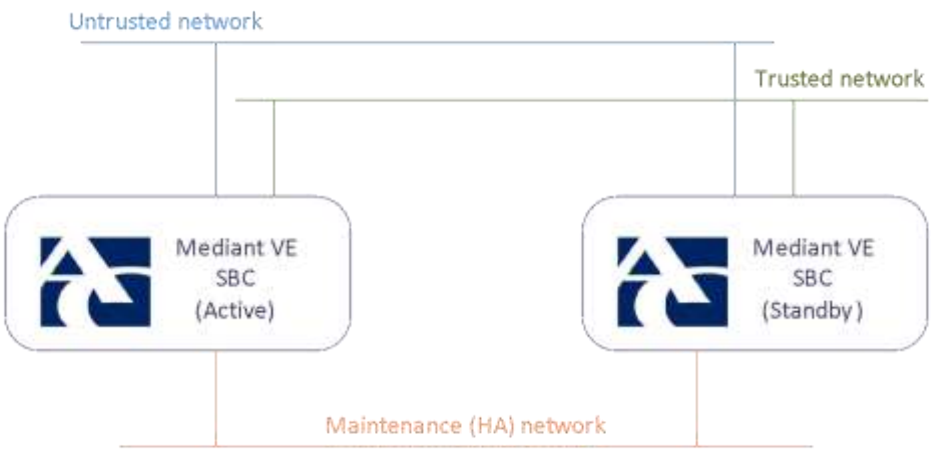
Connect Mediant VE SBC to proper network(s).

If you are using VXLAN tunneling network topology, assign Floating IP address to the Mediant VE SBC instance and configure NATTranslation table as described in chapter 3 above.

If DHCP and metadata services are properly configured in your setup, Mediant VE SBC will automatically configure IP addresses on all available interfaces. Otherwise you will need to connect to its CLI interface – via the virtual console – and configure IP addresses manually. Refer to “LTRT-10407 Mediant Virtual Edition SBC Installation Manual Ver. 7.2” for detailed instructions of network configuration.

5.5 Mediant VE SBC HA Configuration

Mediant VE SBC supports 1+1 Active/Standby HA configuration for achieving carrier-grade solution resiliency. In such configuration a pair of SBC instances is deployed and configured as shown in the picture below.



Maintenance HA network enables keep-alive and state synchronization between Active and Standby SBC units. The only traffic carried via it is traffic between the SBC instances – therefore there is no need to enable connectivity from this network to the “outside world”.

In order to deploy Mediant VE SBC HA configuration you need to create the following entities:

- **mtc** network and corresponding subnet for internal communication between two SBC instances
- **sbc_mtc** security group that will be used on **mtc** network

Login to the CLI interface on the Controller Node.
Create **mtc** network and corresponding subnet.

```
<on controller node>

# . keystonerc
# neutron net-create mtc --shared
# neutron subnet-create --name mtc_subnet --allocation-pool \
          start=192.168.100.224,end=192.168.100.244 mtc 192.168.100.0/24
```

Login to the OpenStack dashboard (Horizon) on the Controller Node.
In Project > Compute > Access & Security > Security Groups create new **sbc_mtc** security group that will be used for internal communication between two SBC instances. Add the following rules to it:

- allow inbound SSH, HTTP and HTTPS traffic
- allow inbound HA traffic between active and redundant SBC units (TCP port 2427).

☐	DIRECTION	ETHER TYPE	IP PROTOCOL	PORT RANGE	REMOTE IP PREFIX	REMOTE SECURITY GROUP	ACTIONS
☐	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	Any	Any	-	default	Delete Rule
☐	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	TCP	443 (HTTPS)	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	TCP	2427	0.0.0.0/0	-	Delete Rule

The simplest way to deploy Mediant VE SBC in 1+1 HA configuration is by using HEAT orchestration template. Create the following template file on your PC:

```
heat_template_version: 2013-05-23

description: >
  HEAT template for 1+1 HA configuration of AudioCodes Mediant VE SBC.

parameter_groups:
- label: instance
  description: SBC instance properties
  parameters:
  - image
  - flavor
  - key_name
- label: network
  description: Networks and subnets into which SBC is deployed
  parameters:
```

```yaml
  - target_net
  - target_subnet
  - mtc_net
  - mtc_subnet
  - mtc_security_group

parameters:
  image:
    type: string
    description: Name of the Mediant VE SBC image
    default: sbc_7.20A.001
    constraints:
      - custom_constraint: glance.image
  flavor:
    type: string
    description: Flavor of the SBC instances
    default: sbc.small
    constraints:
      - custom_constraint: nova.flavor
  key_name:
    type: string
    description: Name of the key pair
    default: admin
    constraints:
      - custom_constraint: nova.keypair
  target_net:
    type: string
    description: Network into which SBC is deployed
    default: admin_flat_net
    constraints:
      - custom_constraint: neutron.network
  target_subnet:
    type: string
    description: Subnet into which SBC is deployed
    default: admin_flat_net__subnet
    constraints:
      - custom_constraint: neutron.subnet
  mtc_net:
    type: string
    description: Maintenance network for internal communication between SBC
instances
```

```
    default: mtc
    constraints:
      - custom_constraint: neutron.network
  mtc_subnet:
    type: string
    description: Maintenance subnet for internal communication between SBC
instances
    default: mtc_subnet
    constraints:
      - custom_constraint: neutron.subnet
  mtc_security_group:
    type: string
    description: Security group for internal communication between SBC
instances
    default: sbc_mtc
    constraints:
      - custom_constraint: neutron.security_group

resources:
  server1:
    type: OS::Nova::Server
    properties:
      name: sbc1
      image: { get_param: image }
      flavor: { get_param: flavor }
      key_name: { get_param: key_name }
      networks:
        - port: { get_resource: server1_port1 }
        - port: { get_resource: server1_port2 }
      config_drive: true
      user_data:
        str_replace:
          template: |
            #ini-file
            HARemoteAddress = $ip
            HAPriority = 10
            HAUnitIdName = sbc1
            #cloud-end

          params:
            $ip: { get_attr: [server2_port2, fixed_ips, 0, ip_address] }
```

```
server1_port1:
  type: OS::Neutron::Port
  properties:
    network: { get_param: target_net }
    fixed_ips:
      - subnet: { get_param: target_subnet }

server1_port2:
  type: OS::Neutron::Port
  properties:
    network: { get_param: mtc_net }
    fixed_ips:
      - subnet: { get_param: mtc_subnet }
    security_groups:
      - { get_param: mtc_security_group }

server2:
  type: OS::Nova::Server
  properties:
    name: sbc2
    image: { get_param: image }
    flavor: { get_param: flavor }
    key_name: { get_param: key_name }
    networks:
      - port: { get_resource: server2_port1 }
      - port: { get_resource: server2_port2 }
    config_drive: true
    user_data:
      str_replace:
        template: |
          #ini-file
          HARemoteAddress = $ip
          HAPriority = 5
          HAUnitIdName = sbc2
          #cloud-end

        params:
          $ip: { get_attr: [server1_port2, fixed_ips, 0, ip_address] }
  depends_on: server1
```

```
  server2_port1:
    type: OS::Neutron::Port
    properties:
      network: { get_param: target_net }
      fixed_ips:
        - subnet: { get_param: target_subnet }
      allowed_address_pairs:
        - ip_address: { get_attr: [server1_port1, fixed_ips, 0, ip_address] }

  server2_port2:
    type: OS::Neutron::Port
    properties:
      network: { get_param: mtc_net }
      fixed_ips:
        - subnet: { get_param: mtc_subnet }
      security_groups:
        - { get_param: mtc_security_group }

outputs:
  public_ip:
    description: The public IP address of the deployed SBC "HA pair"
    value: { get_attr: [server1_port1, fixed_ips, 0, ip_address] }
```

In Project > Orchestration > Stacks launch a new stack.
Choose HEAT template file created in the previous step. Adjust resource names (image, flavor, networks, subnets etc.) to match the actual names used in your setup

5.6 Testing

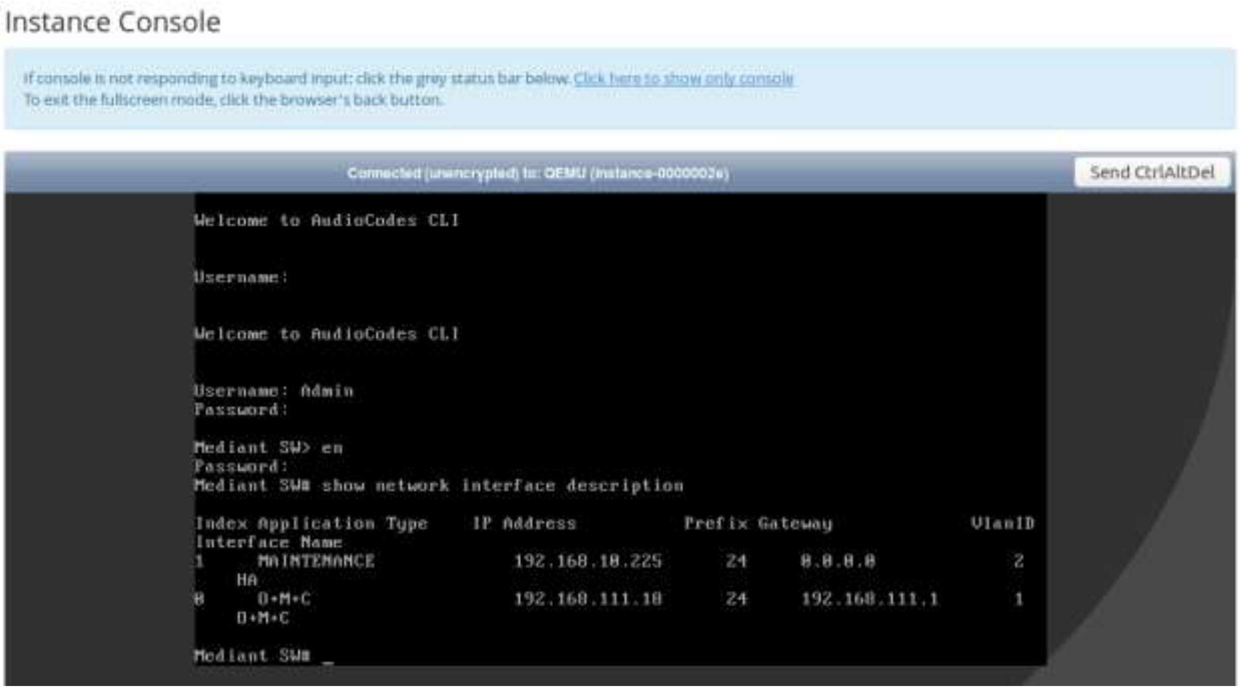
5.6.1 Test Cases

The following test cases were performed in the test environment to ensure correct operation of Mediant VE SBC deployed on Mirantis OpenStack platform.

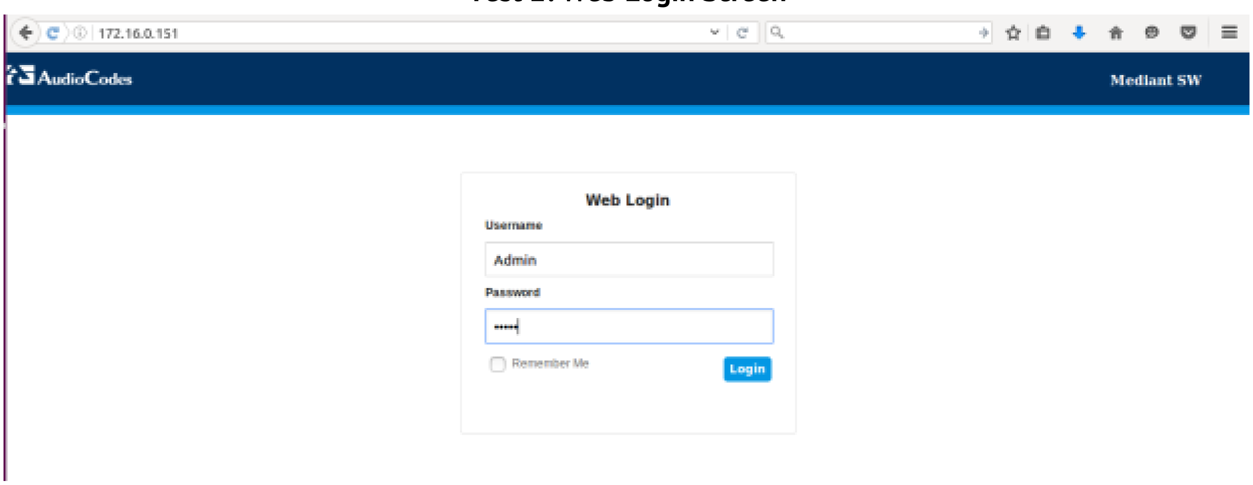
#	Test Description	Status
1	Verify that Mediant VE SBC instance correctly comes up and acquires IP address configuration on all network interfaces	Passed
2	Verify that Mediant VE SBC can be accessed via Web management interface (both HTTP and HTTPS) and basic functionality of this interface	Passed
3	Verify that Mediant VE SBC can be accessed via CLI interface (both SSH and virtual console) and basic functionality of the interface	Passed
4	Configure typical SIP trunking application using SBC Configuration Wizard.	Passed
5	Generate sample traffic using two SIPp generator instances (one – emulating IP-PBX, and another one – SIP Trunk). Verify successful establishment and completion of calls using SIPp call statistics and Mediant VE SBC performance measurements.	Passed
6	Deploy 1+1 Active/Standby HA pair using HEAT template	Passed
7	Emulate failure of Active SBC unit by shutting down the corresponding instance. Verify that standby unit assumes Active role and SBC service continues uninterrupted.	Passed

5.6.2 Test Results

Test 1: Automatic IP configuration



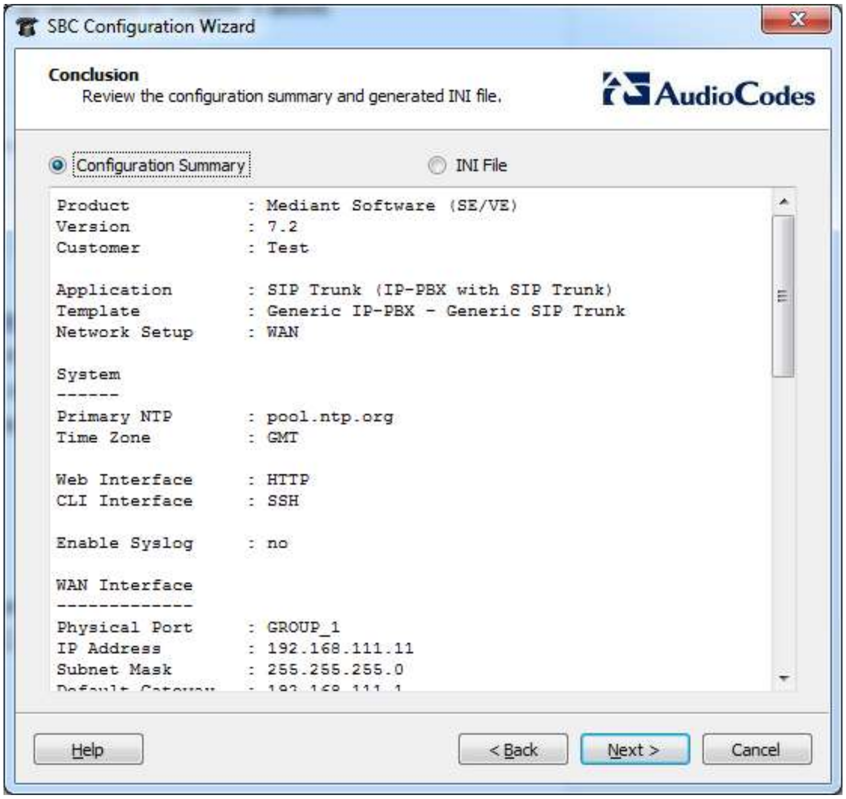
Test 2: Web Login Screen



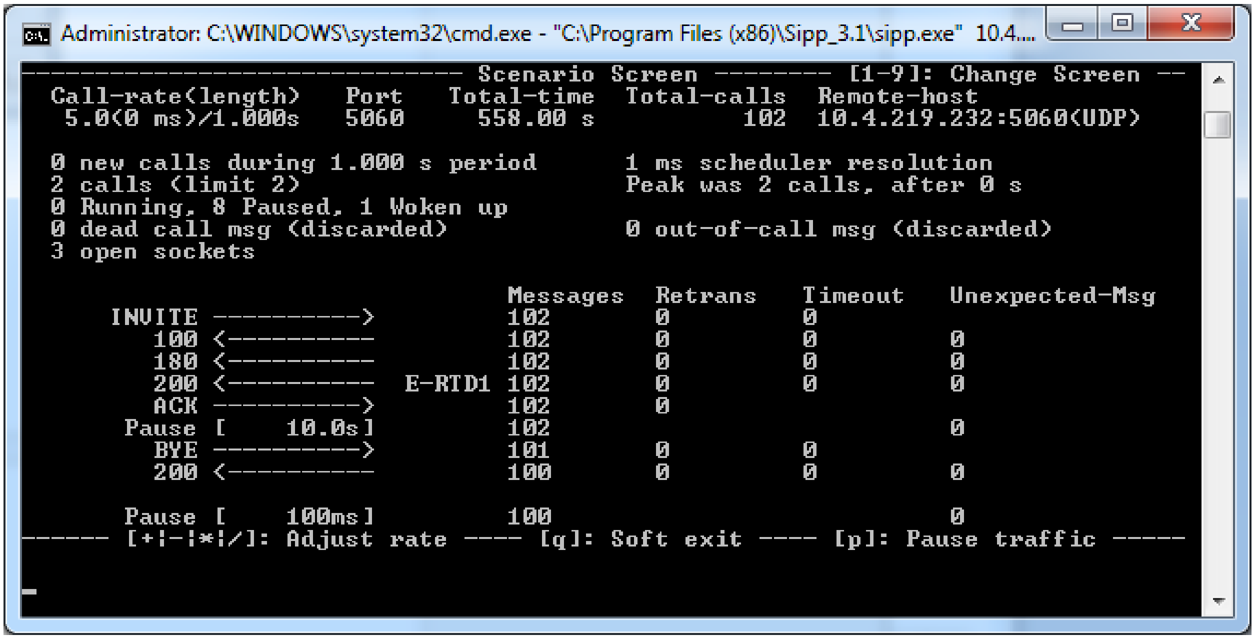
Test 3: CLI Management interface

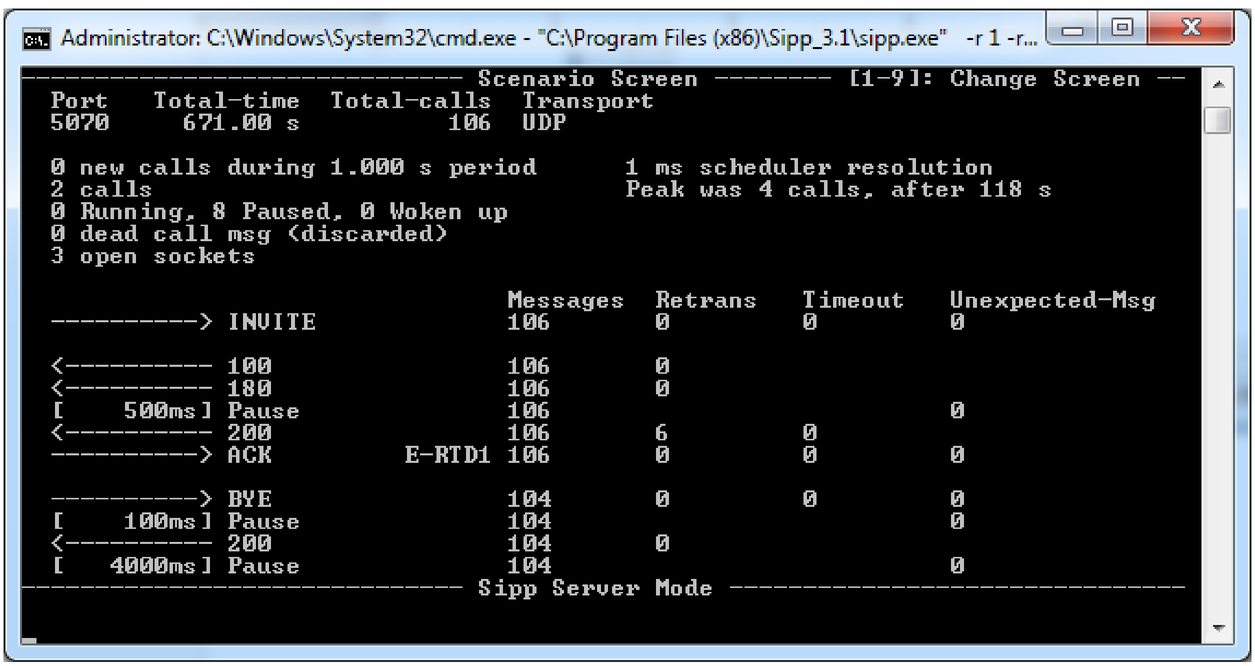
```
alex@alex-ThinkPad-X230: ~/sipp
alex@alex-ThinkPad-X230:~/sipp$ ssh Admin@172.16.0.151
Welcome to AudioCodes CLI
Admin@172.16.0.151's password:
Last login: Tue Aug 09 2016 at 08:59:07

Mediant SW> show version

Version info:
--------------
;Board: Mediant SW
;Board Type: 73
;Serial Number: 11150488428090
;Product Key:
;Slot Number: 1
;Software Version: 7.20A.001
;DSP Software Version: SOFTDSP => 720.23
;Board IP Address: 192.168.111.18
;Board Subnet Mask: 255.255.255.0
;Board Default Gateway: 192.168.111.1
;Ram size: 1876M   Flash size: 0M
;Num of DSP Cores: 0  Num DSP Channels: 0
;Profile: NONE
;;;Key features:;Board Type: Mediant SW ;Max SW Ver: 9.80;Channel Type: DspCh=0 ;HA ;Security: IPSEC MediaEncryption
StrongEncryption EncryptControlProtocol ;DSP Voice features: ;DATA features: ;Coders: G723 G729 G728 NETCODER GSM-FR
GSM-EFR AMR EVRC-QCELP G727 ILBC EVRC-B AMR-WB G722 EG711 MS_RTA_NB MS_RTA_WB SILK_NB SILK_WB SPEEX_NB SPEEX_WB OPUS_
NB OPUS_WB ;Control Protocols: MGCP SIP SBC=3 MSFT ;Default features:;Coders: G711 G726;


;MAC Addresses in use:
;---------------------------
;GROUP_1 - fa:16:3e:97:06:c0
;GROUP_2 - fa:16:3e:a0:56:74
;----------------------------------------------


Mediant SW>
```

Test 4: Configuring SIP Trunking Application Using SBC Wizard

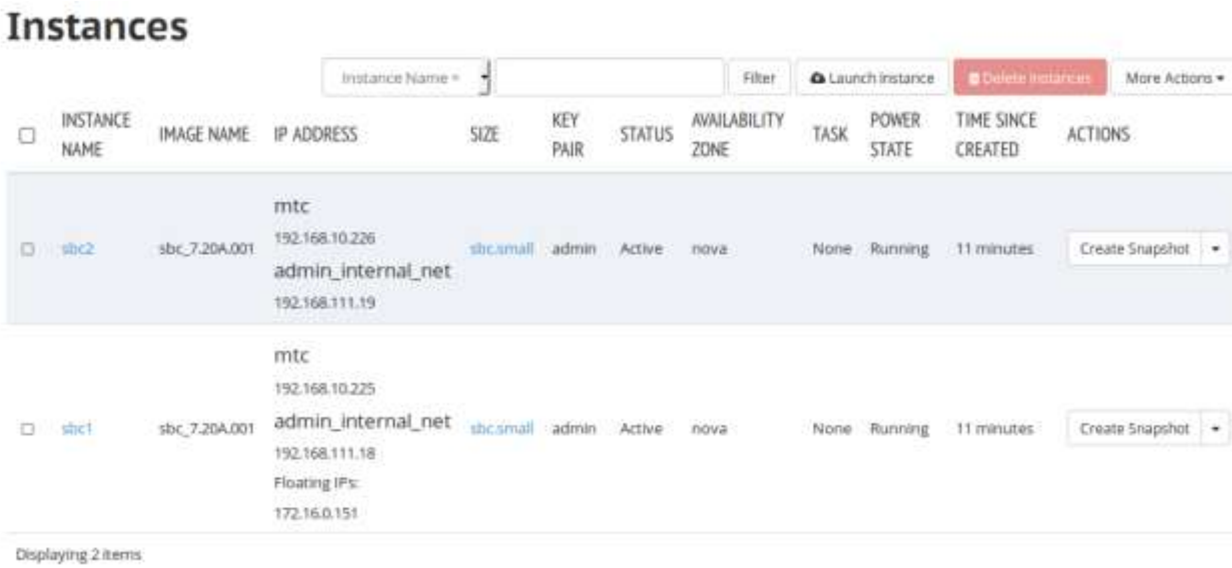


Test 5: Sample Traffic Simulation





Tests 6-7: Mediant VE SBC in HA Configuration



Instances

Instance Name | Filter | Launch Instance | Delete Instances | More Actions

	INSTANCE NAME	IMAGE NAME	IP ADDRESS	SIZE	KEY PAIR	STATUS	AVAILABILITY ZONE	TASK	POWER STATE	TIME SINCE CREATED	ACTIONS
☐	sbc2	sbc_7.20A.001	mtc 192.168.10.226 admin_internal_net 192.168.111.19	sbc.small	admin	Active	nova	None	Running	11 minutes	Create Snapshot
☐	sbc1	sbc_7.20A.001	mtc 192.168.10.225 admin_internal_net 192.168.111.18 Floating IPs: 172.16.0.151	sbc.small	admin	Active	nova	None	Running	11 minutes	Create Snapshot

Displaying 2 items