



Configuration Guide

InGate SIParator/Firewall in Amazon Web Services (AWS)

A how-to Guide

For the Ingate SIParator®/Firewalls using software release 6.2.1 or later
Available thru AWS Marketplace

November 2022

1	INTRODUCTION	3
2	WHY/WHEN AN SBC IS A NEED IN THE CLOUD?	3
3	USE CASES.....	4
3.1	ASSUMPTIONS.....	5
3.2	INITIAL REQUIREMENTS FOR CONFIGURING SIPARATOR.....	5
3.3	CONFIGURING SIPARATOR	6
3.3.1	Management access and connectivity configuration	6
3.3.2	Ingate SIParator/Firewall Type and Mode	7
3.3.3	Networks and Computers	9
3.3.4	Review and configure network interfaces	9
3.3.5	Enable NATing.....	13
3.3.6	Configuring SIP Services.....	13
3.3.7	Configuring Remote Connectivity.....	14
3.3.8	SIP Traffic Configuration	16
3.3.9	SIP Trunks Configurations	17
3.3.10	Intrusion Detection and Prevention	20
3.3.11	Dial Plan Configuration	20
3.4	CONFIGURING PBX AND ENDPOINTS (SOME EXAMPLES).....	23
3.4.1	SIP Trunking considerations.....	23
3.4.2	Inbound Routes	24
3.4.3	Outbound Routes	24
3.4.4	NATing Considerations	24
3.4.5	Phones and endpoints configuration.....	25
4	FIREWALL CONFIGURATION TO SUPPORT PBX NON SIP FEATURES	26
5	ENABLING TLS/SRTP	28
6	INGATE SIPARATOR USING AWS VPN SERVICE.....	33
6.1	AWS VPN SCENARIOS.....	33
6.2	SINGLE OFFICE CONNECTION:.....	33
6.3	MULTISITE WITH HOSTED IP-PBX/UC.....	34
6.4	CONFIGURING TWO VPN TUNNELS FOR YOUR VPN CONNECTION	34
6.4.1	AWS VPN Setup	35
6.4.2	Ingate SIParator VPN Setup	36
6.4.2.1	IPsec Peers	36
6.4.2.2	IPsec Tunnels.....	38
6.4.2.3	IPsec Advanced	39
6.4.2.4	Networks and Computers	39
6.4.2.5	Save/Load Configuration.....	40
7	USING INGATE STARTUP TOOL TG	40
7.1	INTRODUCTION.....	40
7.2	SETTING UP CONNECTIVITY.....	41
8	ADDITIONAL HELP OR SUPPORT	44

Version: InGate SIParator/Firewall version 6.2.1

1 Introduction

Cloud Service Providers are becoming a popular alternative to Capex intensive solution, where building appropriated infrastructure, investing in Hardware and Software, as well as being exposed to a hardware obsolescence, represent some of the traditional entry barriers for technology adoption.

This is already a proven concept and trend for traditional IT infrastructure, and more recently starting to be adopted in Real Time Applications (Voice, Video, PBX, Unified Communications, IM, SMS, etc...)

Cloud infrastructure is allowing Enterprise and Small Business to easily adopt modern IT technologies originally only affordable to Large and Fortune Corporations.

Key Players are driving Cloud infrastructure offerings in the market, counting among others Amazon Web Services (AWS), Google Cloud Platform, Microsoft Azure, IBM Cloud, etc.

Ingate has for many years had a software version of Ingate SIParator® RTS-FW our versatile, powerful and cost-effective SBC (Session Border Controller), firewall and router for virtual machines (in addition to its range of appliances). However, loading and using the software SIParator in a cloud is different offering and the delivery method is usually different that traditional ISO installs in a VM Hypervisor.

This document explains basic use case configuration once you have SIParator Instance already installed.

For details on how to launch an Ingate SIParator instance from scratch, we suggest reviewing our:

[***“Orientation and how to install Ingate SBC and E-SBC on AWS”***](#)

2 Why/when an SBC is a need in the cloud?

There are big differences when deploying your RTC (i.e. Telephony) infrastructure in the cloud (AWS in our case).

- 1) Now any Service hosted in the Cloud, including for instance IPPBX, are published following a DMZ model and imposing a NAT 1-1 for any service when making the service publicly available.
- 2) Even endpoints from now on will be considered remote or at least not neighbour to the Server associated, there are connectivity issues similar to the ones when IPPBX was on premise, to be able to reach the service and don't break media paths, but now this situation becomes wider and not associated just to SIP Trunks and a few remote extensions.
- 3) Some of the issues can be solved by extending the VPC reach to customer offices using Cloud Provided VPN or even direct connection, but that also makes the final solution expensive
- 4) There are no more local users to the PBX anymore.
 - a. Any user is then remote to the PBX
 - b. NATing challenges are imposed also in the far end (User side)

- 5) As RTP is negotiated at the signalling (SIP) level, you will need a way to take full control on what's the best and shortest path for media, without breaking the session.
- 6) AWS don't offer any Firewall SIP aware functionalities, not even any application aware capabilities on their VPC NAT Gateways or Internet Gateways
- 7) As any cloud-based service, now not only you pay for what you use, but also for all traffic flows that happen between end points of a session with transit thru AWS infrastructure. This makes it even more critical if we have bandwidth hungry media applications such as video (WebRTC)

Among several other and traditional reasons to add an SBC in the IPPBX deployment equation, in the Cloud adds the capabilities to maintain cost low when media is an important cost factor (in AWS you will pay for all traffic going in and out of you VPC borders).

Making sure we keep media in the shortest and quality efficient path is one of the big contributions of putting SIParator as your edge device to intermediate external connectivity to users and ITSP's.

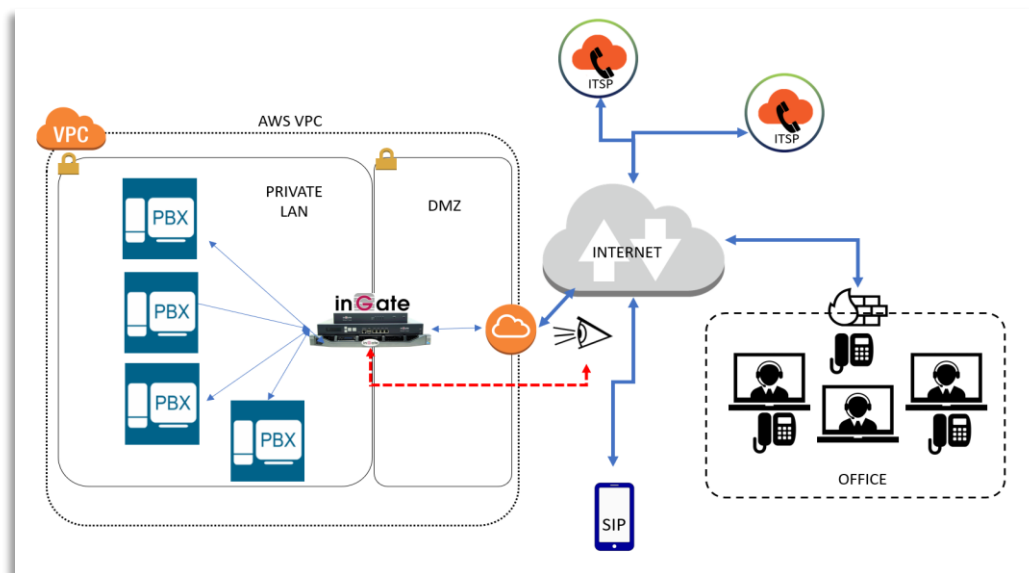


Figure 1

3 Use Cases

In the following sections you will learn all you need to configure your SIParator for the most common uses.

The most typical applications are:

- 1) SIP Trunking, where the SIParator will be responsible to enable and make the connectivity between your SIP service providers and your IPPBX platform behind the SIParator. It will solve any interop issues and solve any issues introduced by the fact that you don't have a public IP address directly assigned to your external

interface (AWS only allow NAT 1-1 for any Service that needs to be published in the Web.

- 2) Users Access. As we know, any user (or endpoint) associated to your IPPBX, because the Service is hosted in the cloud, will consider them remotely located. By such, typical challenges will happen such as FENT (far end NAT traversal), Signalling and media exposure and more.

Typically, this will be useful when AWS has been selected to host the core Telephony/UC infrastructure in several scenarios:

- With connectivity for Corporate Office and Branch Offices via a Public Network (i.e. Internet)
- With Connectivity for Road Warriors (remote users non-statically located)
- Connectivity with one or more ITSPs (either with carrier IP addressing or via Internet with registration)
- Service Continuity and Survivability, to be a secondary registrar (IPPBX Failover)
- Additional needs for endpoints such as Provisioning and management, SMS, phone handset features, WebRTC, Desktop Sharing, Collaboration, etc.
- Secure connections using several options, such as TLS/SRTP, DTLS

3.1 Assumptions

In order to start doing the configuration you need to have provisioned and activated your SIParator Instance in a VPC. To learn how to do so, we suggest you review the [“Orientation and how to install Ingate SBC and E-SBC on AWS”](#) .

3.2 Initial requirements for configuring SIParator.

In this case we assume you have already setup your SIParator with the following attributes:

- 1) VPC already configured (similar to this)
 - VPC allocated in one Region
 - Two Subnets:
 - One named “Public” which will be used to enable Inbound and outbound external access. (10.0.0.0/24)
 - Second one that will be used for instances without direct access (Inbound/Outbound from the outside. It is named “Private”. (10.0.1.0/24)
 - IP-PBX (using an Open Source PBX in our example), located in the Private Subnet with IP address 10.0.1.149.

- Ingate SIParator/Firewall for AWS 6.2.1 installed with 2 Interfaces. Main Interface (eth0) will be designated from the Public Subnet. Second Interface (eth1) will be designated from Private subnet, which is the same subnet where the IP-PBX is located.
- Ingate SIParator will be used as the NAT Instance gateway for the Private Subnet. This will facilitate proper Sip and Media routing Between the external world (i.e. Internet), the PBX and other endpoints related to call flows (such as ITSPs)

3.3 Configuring SIParator

At this point we should be ready to start a deployment with a specific operational support. In our case we will deploy the scenario shown:

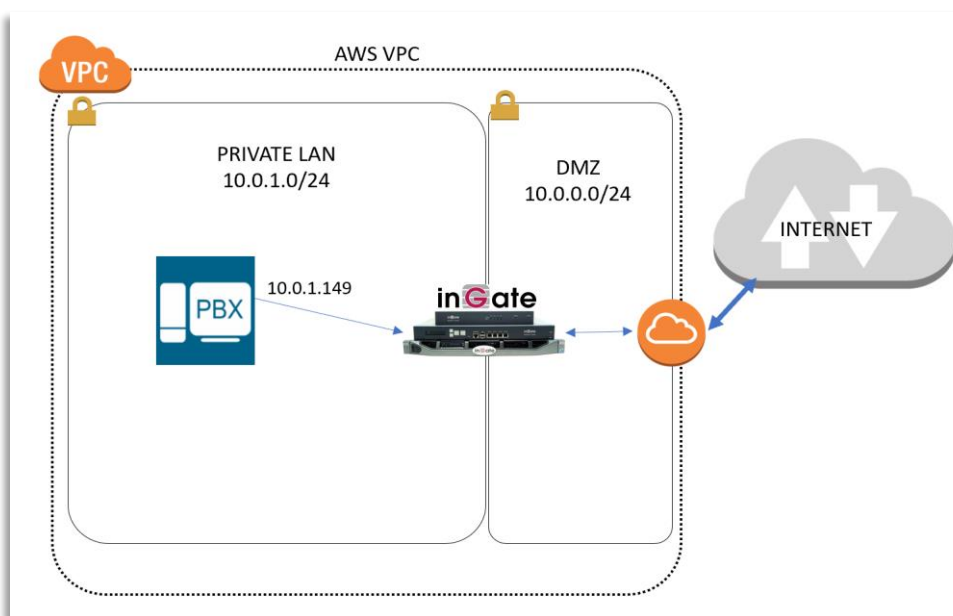


Figure 2

3.3.1 Management access and connectivity configuration

The first steps are related to defining all networking attributes as well as define access control rules regarding GUI and CLI access.

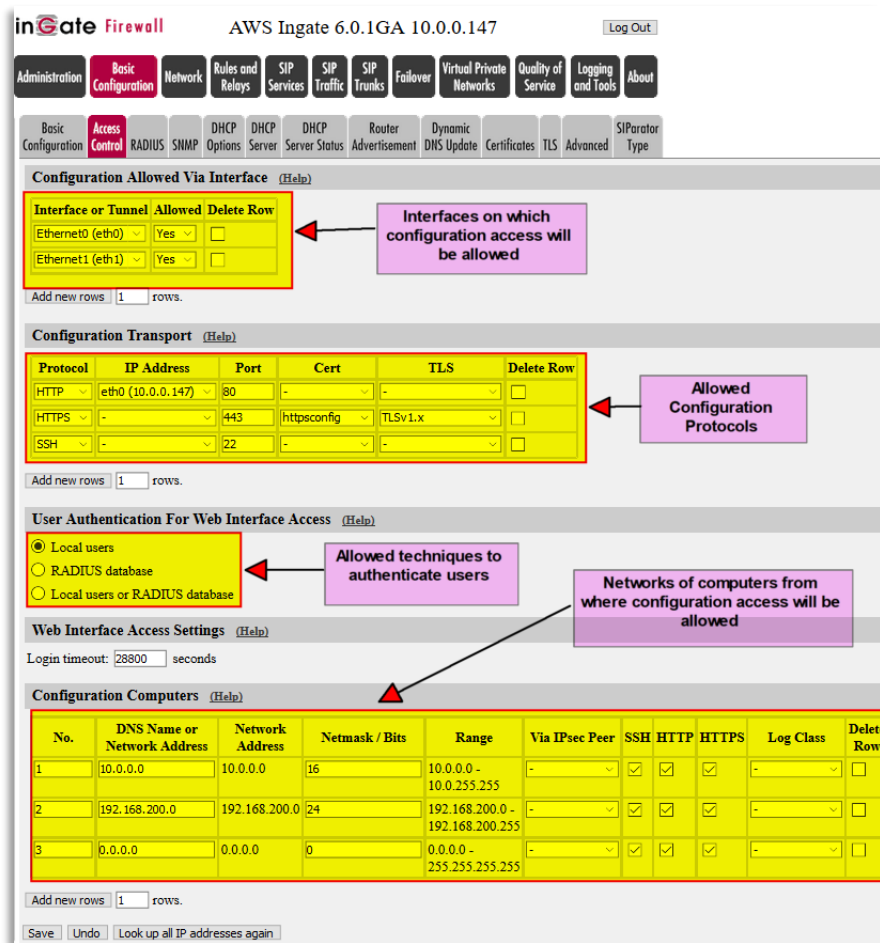


Figure 3

- You can control which eth interfaces will accept access for configuration
- Also, you can define which protocols can be used for configuration access (http, https, ssh)
- User authentication can be done with local database (users are defined in another page), using an external RADIUS server, or both.
- You must define subnets or specific IP addresses from where you can initiate a configuration access.

3.3.2 Ingate SIParator/Firewall Type and Mode

Here we will make sure SIParator is enabled, Topology or also known as Type is selected as DMZ/LAN and finally Firewall mode is enabled.

SIParator type refers to the topology role the Instance will play. Here the options offered:

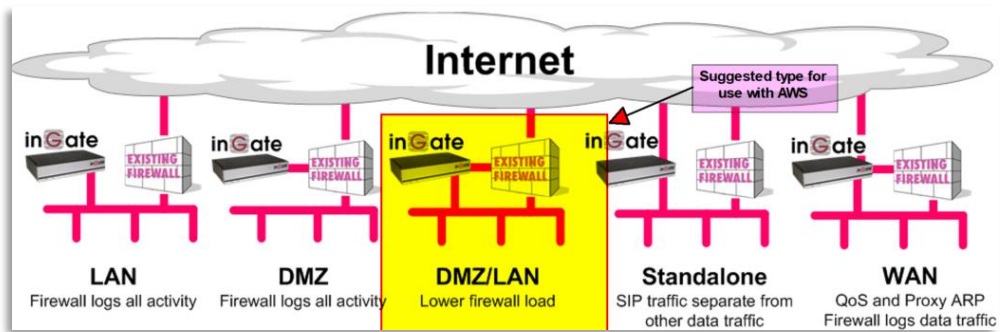


Figure 4

In order to setup both, “Type” and “Mode”:

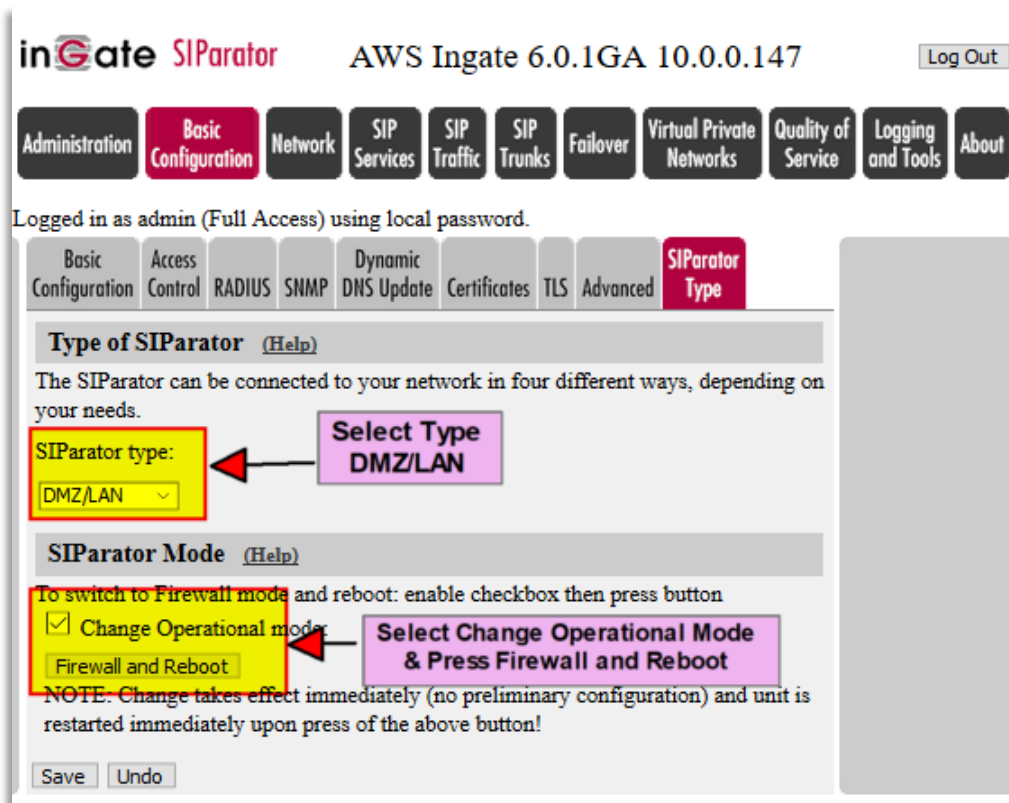


Figure 5

Error! Reference source not found.

DMZ/LAN is suggested as the best fit to VPC with 2 subnets in AWS. Remember VPC, when using Public IP addresses, they are mapped in a NAT 1-1 to a Network Interface with a private IP address. Also, our Instance has 2 eth interfaces and the eth1 is connected to the Private subnet (LAN)

Enabling Firewall Mode will complete all what is needed to implement the NAT Instance functionality.

3.3.3 Networks and Computers

Here we will define by name and IP's networks and computers we will refer to during configuration in other sections.

It is a powerful way for easy maintenance in case we need to change IP addressing of any network resource used in many places inside the Instance configuration.

Name	Subgroup	Lower Limit		Upper Limit (for IP ranges)		Interface/VLAN	Delete Row
		DNS Name or IP Address	IP Address	DNS Name or IP Address	IP Address		
IPPBX	-	10.0.1.149	10.0.1.149	10.0.1.149	10.0.1.149	-	<input type="checkbox"/>
Internet	-	0.0.0.0	0.0.0.0	255.255.255.255	255.255.255.255	Ethernet0 (eth0 untagged)	<input type="checkbox"/>
Office	-	192.168.200.0	192.168.200.0	192.168.200.255	192.168.200.255	-	<input type="checkbox"/>
PrivateLan	-	10.0.1.0	10.0.1.0	10.0.1.255	10.0.1.255	-	<input type="checkbox"/>
PublicLan	-	10.0.0.0	10.0.0.0	10.0.0.255	10.0.0.255	-	<input type="checkbox"/>
Sip Trunk	-					Ethernet0 (eth0 untagged)	<input type="checkbox"/>
	-					Ethernet0 (eth0 untagged)	<input type="checkbox"/>
access	Internet					-	<input type="checkbox"/>
	Office					-	<input type="checkbox"/>
aws_vpc	PrivateLan					-	<input type="checkbox"/>
	PublicLan					-	<input type="checkbox"/>

Figure 6

In this example:

- IP-PBX. Is the IP-PBX IP address in the private network.
- Internet. Any address on the Public side (eth0) of the Instance
- Office. IP Network address of a remote Office (Note they are private IP addresses). This will allow us to refer to such network later.
- PrivateLan. IP address of our VPC Private Subnet.
- PublicLan. IP address of our VPC Public Subnet (DMZ)
- SIP Trunk. IP addresses of 2 SIP trunk destinations belonging to same ITSP. We will be able to refer to both Trunks with a single name.
- access. A single name to group Internet and Office as networks to which we will provide access for remote endpoints
- aws_vpc. A single name to group VPC Private and Public Subnets under a single name

3.3.4 Review and configure network interfaces

Now we will configure eth0 (Public Interface) and eth1 (Private Interface). Most of the configuration is already done automatically, but let's confirm.

Interface eth0:

inGate Firewall AWS Ingate 6.0.1GA 10.0.0.147 Log Out

Administration Basic Configuration **Network** Rules and Relays SIP Services SIP Traffic SIP Trunks Failover Virtual Private Networks Quality of Service Logging and Tools About

Networks and Computers Default Gateways All Interfaces NAT VLAN **Eth0** Eth1 Interface Status PPPoE Tunnels Topology

General

Physical device: eth0

This interface is: Active Inactive

Interface name:

Directly Connected Networks (Help)

Name	Address Type	DNS Name or IP Address	IP Address	Netmask / Bits	Network Address	Broadcast Address	VLAN Id	VLAN Name	Delete Row
eth0	Static	10.0.0.147	10.0.0.147	24	10.0.0.0	10.0.0.255		-	<input type="checkbox"/>

Add new rows rows.

Alias (Help)

Below are the ranges from which you can select aliases.

Name	DNS Name or IP Address	IP Address	Delete Row
------	------------------------	------------	------------

Add new rows rows.

Proxy ARP (Help)

Get Network From	Proxy ARPed Network			VLAN Id	VLAN Name	Delete Row
	DNS Name or Network Address	Network Address	Netmask / Bits			

Add new rows rows.

Static Routing (Help)

Routed Network			Router		Delete Row	
DNS Name or Network Address	Network Address	Netmask / Bits	Dynamic	DNS Name or IP Address		
0.0.0.0	0.0.0.0	0	-	10.0.0.1	10.0.0.1	<input type="checkbox"/>

Add new rows rows.

Save Undo Look up all IP addresses again

Figure 7

- Change the interface name to Outside to facilitate identification in other places inside the configuration. Outside meaning the one that will be used to face Internet.
- AWS VPC automatically assigns IP address via DHCP (in this case 10.0.0.147), so let's change to static keeping the same IP address. This will add some additional flexibility later in the Firewall capabilities. In any case you can select Dynamic and AWS will assign the IP for you.
- Let's make sure we have the static route to send all default traffic to the pre-established VPC subnet default gateway 10.0.0.1 (This can't be changed as per AWS VPC requirements). If you selected dynamic in the IP address, it can be selected dynamic too associated to the same interface.

We will do something similar with eth1, with the exception of not having a static route, as there is only one default gateway.

inGate Firewall AWS Ingate 6.0.1GA 10.0.0.147 Log Out

Administration Basic Configuration **Network** Rules and Relays SIP Services SIP Traffic SIP Trunks Failover Virtual Private Networks Quality of Service Logging and Tools About

Networks and Computers Default Gateways All Interfaces NAT VLAN Eth0 **Eth1** Interface Status PPPoE Tunnels Topology

General

Physical device: eth1

This interface is: Active Inactive

Interface name:

Directly Connected Networks [\(Help\)](#)

Name	Address Type	DNS Name or IP Address	IP Address	Netmask / Bits	Network Address	Broadcast Address	VLAN Id	VLAN Name	Delete Row
eth1	Static	10.0.1.147	10.0.1.147	24	10.0.1.0	10.0.1.255		-	<input type="checkbox"/>

Add new rows rows.

Alias [\(Help\)](#)

Below are the ranges from which you can select aliases.

Name	DNS Name or IP Address	IP Address	Delete Row
------	------------------------	------------	------------

Add new rows rows.

Proxy ARP [\(Help\)](#)

Get Network From	Proxy ARPed Network			VLAN Id	VLAN Name	Delete Row
	DNS Name or Network Address	Network Address	Netmask / Bits			

Add new rows rows.

Static Routing [\(Help\)](#)

Routed Network			Router		Delete Row
DNS Name or Network Address	Network Address	Netmask / Bits	Dynamic	DNS Name or IP Address	

Add new rows rows.

Figure 8

A summary of the configuration done can be verified or even modified in the All Interfaces Tag:

inGate Firewall AWS Ingate 6.0.1GA 10.0.0.147 [Log Out](#)

Administration Basic Configuration **Network** Rules and Relays SIP Services SIP Traffic SIP Trunks Failover Virtual Private Networks Quality of Service Logging and Tools About

Networks and Computers Default Gateways **All Interfaces** NAT VLAN Eth0 Eth1 Interface Status PPPoE Tunnels Topology

Interface Overview

General

Physical Device	Interface Name	Active
eth0	Outside	Yes
eth1	Inside	Yes

Directly Connected Networks [\(Help\)](#)

Name	Address Type	DNS Name or IP Address	IP Address	Netmask / Bits	Network Address	Broadcast Address	Interface or Tunnel	VLAN Id	VLAN Name	Delete Row
eth0	Static	10.0.0.147	10.0.0.147	24	10.0.0.0	10.0.0.255	Outside (eth0)		-	<input type="checkbox"/>
eth1	Static	10.0.1.147	10.0.1.147	24	10.0.1.0	10.0.1.255	Inside (eth1)		-	<input type="checkbox"/>

Add new rows rows.

Alias [\(Help\)](#)

Name	DNS Name or IP Address	IP Address	Interface	Delete Row
------	------------------------	------------	-----------	------------

Add new rows rows.

Proxy ARP [\(Help\)](#)

Get Network From	Proxy ARPed Network			Interface	VLAN Id	VLAN Name	Delete Row
	DNS Name or Network Address	Network Address	Netmask / Bits				

Add new rows rows.

Static Routing [\(Help\)](#)

Routed Network			Router				Delete Row
DNS Name or Network Address	Network Address	Netmask / Bits	Dynamic	DNS Name or IP Address	IP Address	Interface or Tunnel	
0.0.0.0	0.0.0.0	0	-	10.0.0.1	10.0.0.1	Outside (eth0)	<input type="checkbox"/>

Add new rows rows.

Unreachable [\(Help\)](#)

Unreachable Network			Delete Row
DNS Name or Network Address	Network Address	Netmask / Bits	

Add new rows rows.

[Save](#) [Undo](#) [Look up all IP addresses again](#)

Figure 9

3.3.5 Enable NATing

In our case SIParator/Firewall will be NATing from the Inside to the Outside. In order to configure NATing we will do it in the NAT tag under Network:

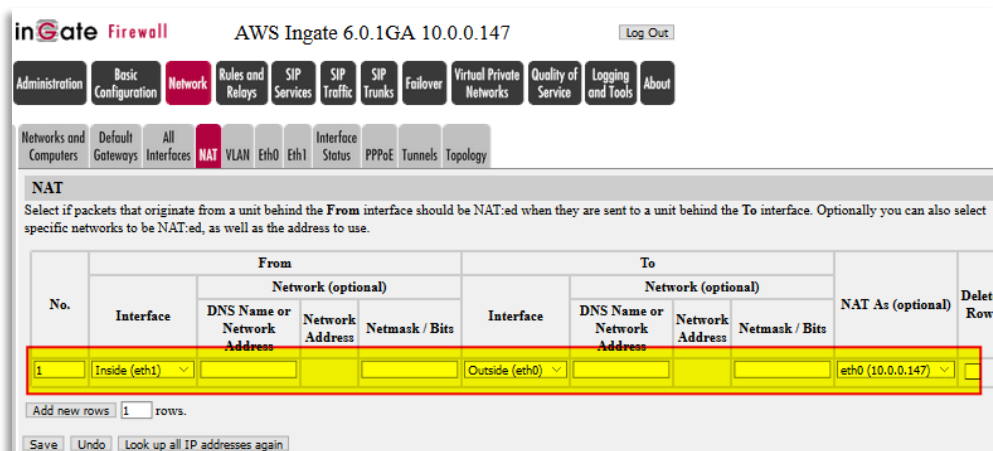


Figure 10

3.3.6 Configuring SIP Services

First let's define the basic SIP elements:

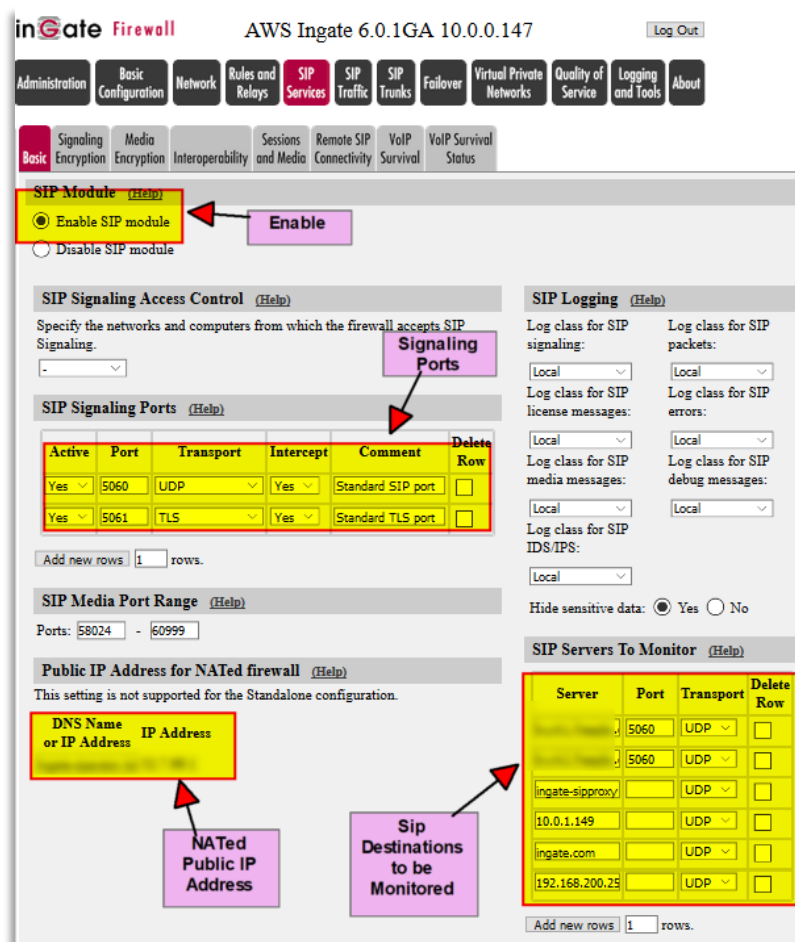


Figure 11

Error! Reference source not found.

- Make sure SIP module is enabled.
- Let's enable SIP on UDP and TLS. In this case we are using standard ports, but you can change them according to your needs.
- As the SIParator/Firewall is type DMZ/LAN and the outside interface is a private IP, you should complete the Public NATed IP address. You can use an IP Address or FQDN which will be resolved accordingly. In case you need it, the actual Public IP is always shown in the About page in the GUI.

Cloud Service Information (AWS)	
ami-id:	ami-f3
mac:	12:d8:ce:a2:f2:12
public-ipv4:	
instance-type:	t2.small
instance-id:	i-0
placement/availability-zone:	us-east-1a

Figure 12

- The Public IP address can be obtained and associated to eth0 interface using AWS Elastic IP option in EC2 Service.
- Any SIP destination can be monitored. This function is done via SIP Options pings. You can add any destination here and you will be able to see their status in the SIP Status TAG.

3.3.7 Configuring Remote Connectivity

Remote Connectivity section will allow remote users/endpoints to be properly managed and integrated via Ingate SIParator/Firewall to the IP-PBX in the Private Subnet.

This section enables features needed to manage and solve Far End Nat Traversal challenges.

In our case we will just enable the more typical attributes for FENT but will not use STUN.

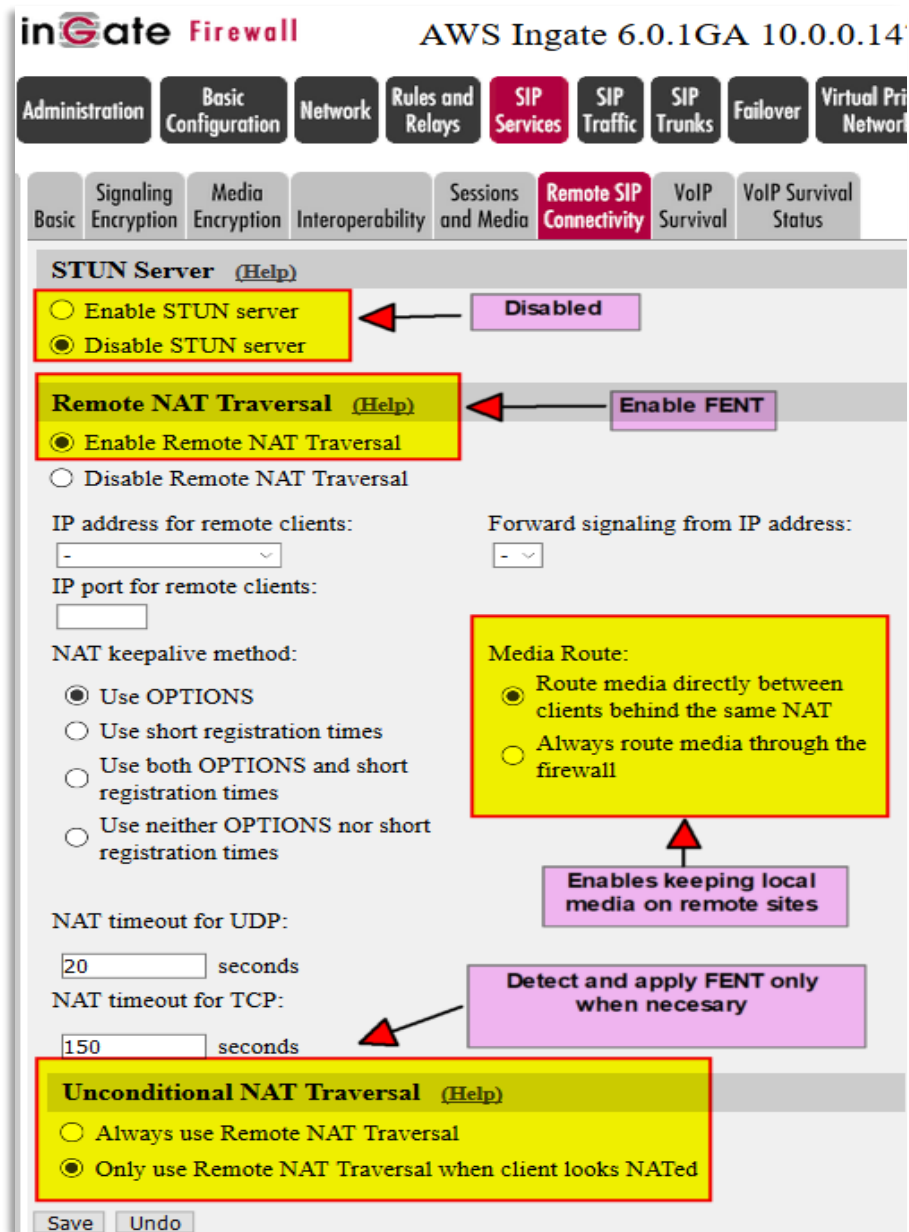


Figure 13

- If necessary, you can enable STUN and provide configuration for STUN Server, etc. In our case we will not use STUN. Most cases it will be enough.
- An interesting feature we recommend enabling is to try keeping media local between endpoint behind the same NAT. This will work unless the IP-PBX forces Media Relay.
- Enable Remote NAT Traversal. Most endpoints in Cloud hosted environments are always remotely located and most likely behind a NAT.
- However, we will enable detecting when NAT Traversal is necessary. You can also select Always use Remote NAT transversal and manually add exceptions.

3.3.8 SIP Traffic Configuration

In this section we will configure everything related to SIP traffic flows, including dial plan, routings, etc.

Some tags will be left with factory default values, but you can later try them and test (In most cases it will depend on specific interop needs with specific Vendors).

The first tag we will configure is Routing, as this one defines the core logics behind call flow

Let's assume we have an FQDN that resolves on the external public IP Address for the Ingate SIParator/Firewall. We will use this FQDN as the domain for our exercise.

One of the key features in Ingate is the DNS Override, which allows the SIParator/Firewall to modify and replace the Domain to a desired IP address. This is very similar to having a local DNS re-resolving the domain name. This way SIParator/Firewall will be able to forward SIP requests arriving to the Outside interface to the IP-PBX behind the Inside interface.

It will also enable protocol conversion for example to convert incoming SIP requests from UDP to TCP or TLS, or vice versa.

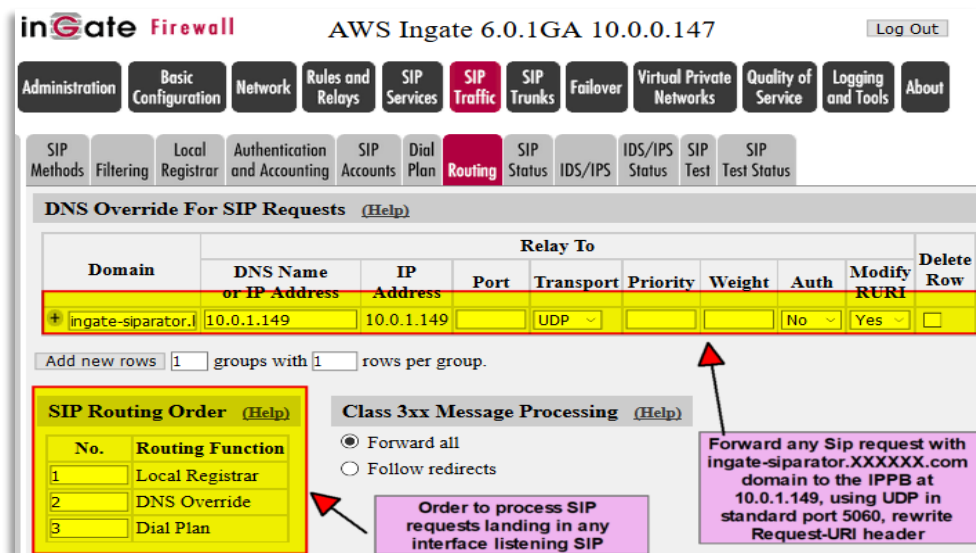


Figure 14

- We will use DNS Override to forward SIP requests coming from remote endpoints to the IP-PBX. It is important to understand that requests arriving at this point has been already cleaned and filtered from any threat, by SIP Methods, Malformed Packets, IDS/IPS and Filters.
- Another level of control here is the priority and order on applying Local registrar, DNS Override and Dial plan. In our example, Routing will check first if SIP request is sent to a local registered UA, secondly will match domain to decide if it needs to be forwarded (as defined in DNS Override) and third look for matching in the Dial Plan. The first Match of the 3 is the one that will be used.

3.3.9 SIP Trunks Configurations

In our exercise we will have 3 ITSPs, and one of them has two destinations for failover.

We will use one of the most powerful and simplified features in Ingate SIParator/Firewall SIP Trunk pages.

A SIP Trunk Page defines a path that connects an ITSP with an IP-PBX with specific configuration needs.

A single IP-PBX could be destination of several ITSP Trunks, and same ITSP Trunk can be used in more than one destination IP-PBX (i.e. DID's define which IP-PBX should receive the call).

Here we will show only one of the SIP Trunks in our exercise:

inGate Firewall AWS Ingate 6.0.1GA 10.0.0.147 [Log Out](#)

Administration Basic Configuration Network Rules and Relays SIP Services SIP Traffic SIP Trunks Failover Virtual Private Networks Quality of Service Logging and Tools About

View trunk: **SIP Trunk 1: Sigestation1;IPPBX** [Goto SIP Trunk page](#)

SIP Trunk 4 [\(Help\)](#)

Enable SIP Trunk
 Disable SIP Trunk

SIP Trunking Service [\(Help\)](#)

Use parameters from other SIP trunk
 Define SIP trunk parameters

Service name: *(Unique descriptive name)*

Service Provider Domain: *(FQDN or IP address)*

Restrict to calls from: *('.' = No restriction)*

Outbound Proxy: *(FQDN or IP address)*

Use alias IP address: *(Forces this source address from our side)*

Outbound Gateway: *('.' = Use Default Gateway)*

Signaling Transport: *('.' = Automatic)*

Port number:

From header domain:

Host name in Request-URI of incoming calls: *(Trunk ID - Domain name)*

Remote Trunk Group Parameters (RFC 4904):
Used as: *('.' = Don't use TGP)*

Local Trunk Group Parameters (RFC 4904):
Used as: *('.' = Don't use TGP)*

Preserve Max-Forwards:

Relay media:

Exactly one Via header:

'gin' registration (RFC 6140):

Hide Record-Route:

Show only one To tag:

SIP 3xx redirection to provider domain:

SIP 3xx redirection to caller domain:

Route incoming based on:

Service Provider domain is trusted: *(For P-Asserted-Identity)*

Use P-Preferred-Identity: *(Instead of P-Asserted-Identity)*

Forward outgoing REFER:

Max simultaneous calls: *(Call Admission Control)*

Max simultaneous calls per Trunk Line:

Figure 15

Previous figure corresponds only to the ITSP side of the Trunk Page.

- This Trunk Page associate carrier trunk named “Sotel” with the IP-PBX in the Private Subnet. Use help link to get a full explanation for each parameter

- You should adjust parameters and interop attributes based on your ITSP requirements.
- You can control for example maximum simultaneous calls in the SIP trunk or limit by Trunk Line (A trunk Line in this case could be a DID)

Outgoing Calls to the trunk are sent to a specific SIP Trunking page via Forward To in the Dial Plan. The from header in an outgoing call is searched for a match in the From-columns.

Incoming Calls from the trunk are first scanned through the Incoming Trunk Match columns and only sent to the Dial Plan if no match is found.

Use “Help” links to obtain detailed information.

Main Trunk Line (Help)									
No.	Reg	Display Name	User Name	Identity	User ID	Password	Incoming Trunk Match	Forward to	
1	Yes		0291	Authentication attributes	0291	Change Password			

PBX Lines (Help)										
No.	Reg	From PBX Number/User	Display Name	User Name	Identity	User ID	Password	Incoming Trunk Match	Forward to PBX Account	Delete Row
6	No				Inbound DID routing. Destination in the PBX		Change Password	0291	0291	<input type="checkbox"/>
7	No						Change Password	0292	0292	<input type="checkbox"/>

Figure 16

- If the SIP Trunk requires implicit registration you need to enable it here
- You can load Authentication credentials that will be used for registration and call authentication challenges
- Incoming DID’s can be routed to specific UA inside the IP-PBX

Figure 17

- Here you can associate a new PBX to this Trunkgroup or just refer to another already created.

- Make sure you configure the IP address. In our case it will be 10.0.1.149 located in the Private Subnet
- Complete the remaining parameters associated to the IP-PBX.

3.3.10 Intrusion Detection and Prevention

Here we will enable the default options predefined in SIParator/Firewall to detect and prevent Intrusion attacks.

The way it works is based on:

- Definition on how to match a potential threat (Packet Match Definition)
- Packet Rate Thresholds to control potential brute force and DDOS attacks.
- Rules, to define how to apply and what to do based on the matches.
- Maximum System Load to avoid system stops for overload.

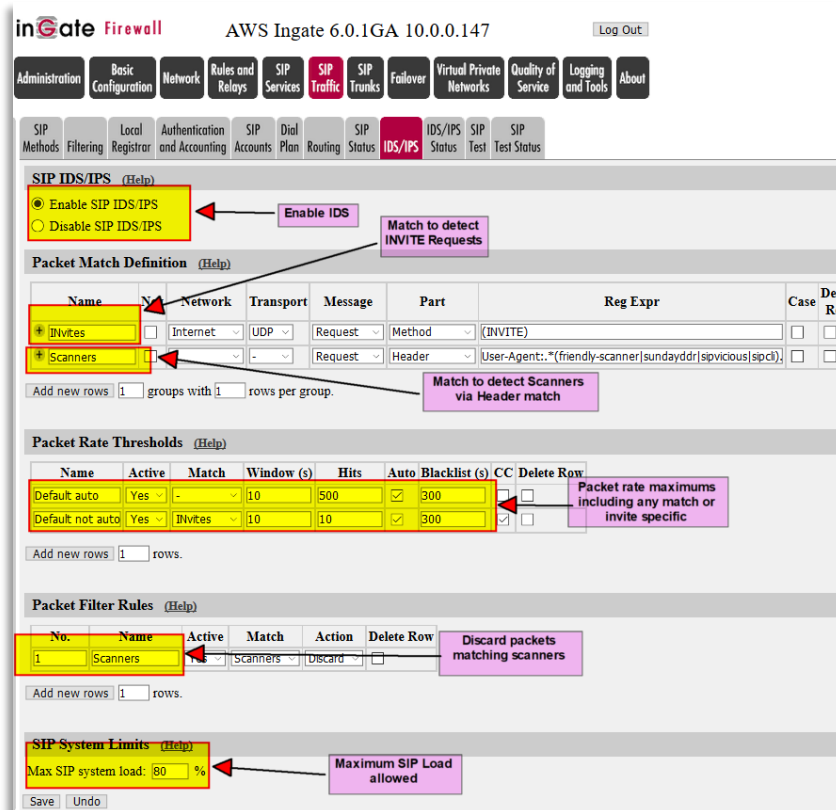


Figure 18

3.3.11 Dial Plan Configuration

Dial Plan, as one of the Routing options could be applied depending on how the order was defined in the Routing tag and how the request is matched.

The Dial plan is based on 4 lists of elements:

- 2 Matching elements lists
 - From Header.
 - Request-Uri

- Forward to destinations list
- Dial Plan actual Rules

Any request arriving to the Dial Plan, can be matched looking in 2 components of the request, “From Header” and “Request-URI”

In our case we will process via Dial Plan call coming from the IP-PBX, in which case we will match From Header domain = 10.0.1.149 and will match and strip call sent to the SIParator/Firewall Inside interface (10.0.1.147) with prefixes 91, 92 and 93.

In our exercise we will have 3 potential destinations to forward calls to, depending on the prefix matched in the Request-URI.

The Dial Plan Rules based on the combined matching of IP-PBX and prefix used will route the calls to one of 3 ITSPs.

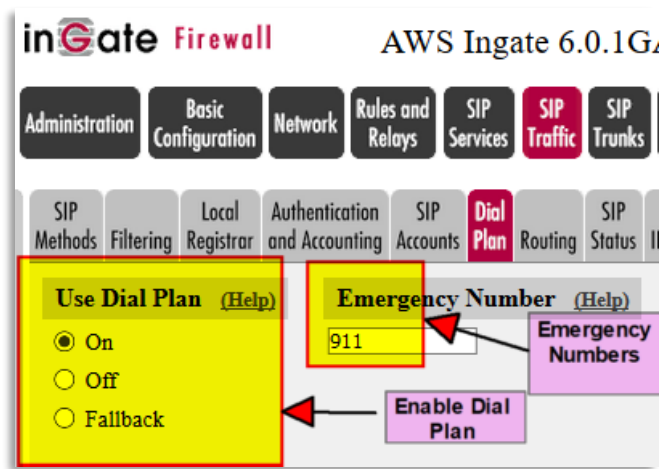


Figure 19

- The Dial Plan is an advanced routing tool for SIP signaling. For each line in the Dial Plan, you can match an incoming SIP message on the SIP From header and the Request-URI. Based on this, you will be able to define how the SIP message should be forwarded. The Dial Plan can be turned On, Off or used in fallback mode. In fallback mode, the Dial Plan is inactive unless a SIP server to be routed to, is out of order. As a backup, the Dial Plan then becomes active.
- Enter the emergency phone number for your country (like 112 or 911). Calls to this number will be allowed even if all Concurrent Calls SIP Trunk Sessions are used up. If you have multiple emergency numbers, you can add each additional number separated by a space character. You cannot enter more than 5 numbers.

Matching From Header (Help)

Name	Use This Or This	Transport	Network	Delete Row
Username	Domain	Reg Expr			
IPPBX	*	10.0.1.149	UDP	IPPBX	<input type="checkbox"/>

Add new rows | 1 | rows.

Matching Request-URI (Help)

Name	Use This Or This	Delete Row			
Prefix	Head	Tail	Min. Tail	Domain	Reg Expr	
Outbound_Ingate	90		0..9, +, -, #, *		10.0.1.147	<input type="checkbox"/>
Outbound_Sipst	91		0..9, +, -, #, *		10.0.1.147	<input type="checkbox"/>
Outbound_Sotel	92		0..9, +, -, #, *		10.0.1.147	<input type="checkbox"/>

Add new rows | 1 | rows.

Forward To (Help)

Name	No.	Use This Or This	... Or This	... Or This	Use Alias IP	Delete Row
Account	Replacement Domain	Port	Transport	Reg Expr	Trunk		
# Sipstation	1	-	-	-	SIP Trunk 1: Sipstation1;IPPBX	<input type="checkbox"/>	<input type="checkbox"/>
# ingate	1	-	-	-	SIP Trunk 2: Sipstation2;IPPBX	<input type="checkbox"/>	<input type="checkbox"/>
# sotel	1	-	-	-	SIP Trunk 3: ingate;IPPBX	<input type="checkbox"/>	<input type="checkbox"/>
					SIP Trunk 4: Sotel;IPPBX	<input type="checkbox"/>	<input type="checkbox"/>

Add new rows | 1 | groups with 1 | rows per group.

Figure 20

- Match any call coming from the IP-PBX (10.0.1.149)
- Match any calls sent to SIParator (10.0.1.147) with specific Prefixes that will be stripped out
- “Forward to” destinations using 3 different ITSPs and associated to their SIP trunks

Dial Plan (Help)

No.	From Header	Request-URI	Action	Forward To	Add Prefix	ENUM Root	Time Class	Comment	Delete Row
					Forward	ENUM			
1	IPPBX	Outbound_Sotel	Forward	sotel			-		<input type="checkbox"/>
2	IPPBX	Outbound_Sipstation	Forward	Sipstation			-		<input type="checkbox"/>
3	IPPBX	Outbound_Ingate	Forward	ingate			-		<input type="checkbox"/>

Add new rows | 1 | rows.

Figure 21

- This is the actual dial plan.
- Here we are deciding which destination will be applied based on the match combination of “From” and “Request-URI”

At this point, everything is ready in the SIParator/Firewall to start mediating remote endpoints with the IP-PBX, as well as SIP trunks between the IP-PBX and ITSPs. Now next sections will show us what needs to be done in IP-PBX as well as endpoints

3.4 Configuring PBX and Endpoints (Some examples)

For this documentation, we are using an Asterisk Open Source IP-PBX and most of the specific configuration suggestions can be extrapolated to other platforms.

3.4.1 SIP Trunking considerations

As SIParator/Firewall is taking care of mediation with various ITSPs we will have only one SIP trunk defined in the PBX to send all outbound calls to SIParator, regardless of who is the carrier to be used.

IP-PBX SIP Trunk definition:

Outgoing

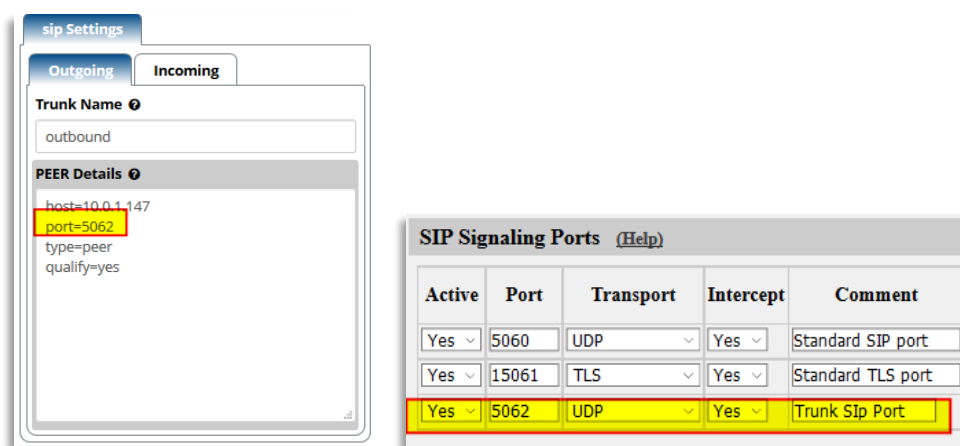


Figure 22

- Notice we are assigning a separated port to listen SIP signalling in the SIParator/Firewall. Make sure you make the adjustment in the SIP Service/Basic tag → signalling ports
- Any outbound call from IP-PBX to PSTN will be sent to SIParator/Firewall Inside Interface (10.0.1.147) port 5062

Incoming

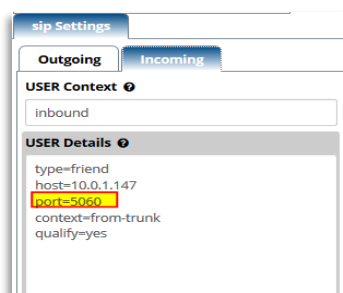


Figure 23

- Incoming calls from PSTN will come via SIParator/Firewall (10.0.1.147) and port is by default 5060

3.4.2 Inbound Routes

Inbound routs can be defined based on DID, as any call coming from any ITSP will be routed to the IP-PBX

DID	CID	Description	Destination
Any	Any		Terminate Call: Play no service message
70291	Any	Sotel 1	Extensions: 3001 Joe Doe 3001
70292	Any	Sotel 2	Extensions: 3001 Joe Doe 3001
76800	Any	Support Inbound & Fax	Extensions: 3001 Joe Doe 3001
45780	Any	ErnestoIngate	Extensions: 3001 Joe Doe 3001
51400	Any	Conference Bridge	Conferences: 83999 Office Bridge

Figure 24

3.4.3 Outbound Routes

All outbound calls will be sent from IP-PBX via Trunk to SIParator/Firewall

Figure 25

3.4.4 NATing Considerations

SIParator/Firewall is taking care of all connectivity with the Outside/PSTN, all NAT aware parameters in the PBX should be disabled

Figure 26

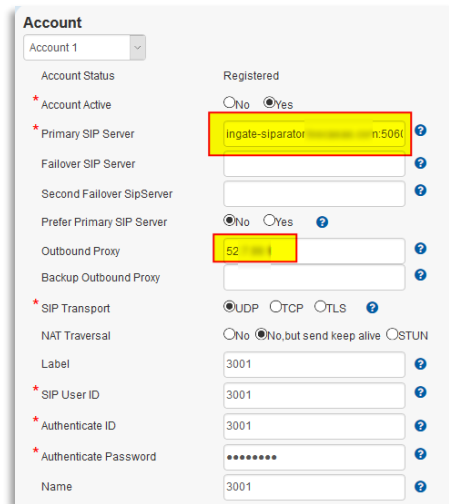
Error! Reference source not found.

3.4.5 Phones and endpoints configuration

Now any endpoint, to be registered and located from the Outside, will use SIParator/Firewall as his Outbound proxy and will use the domain defined in DNS Override as shown in Figure 14

Some examples here:

S500 Phone:

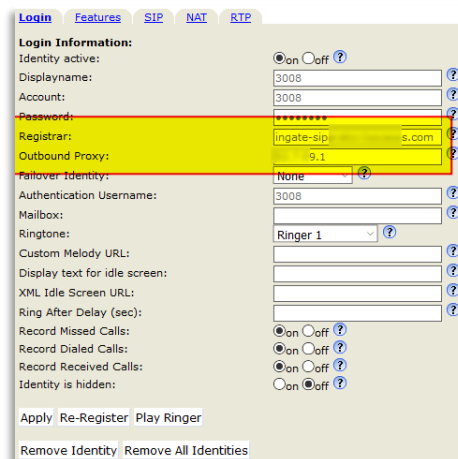


The screenshot shows the 'Account' configuration page for an S500 phone. The 'Account' dropdown is set to 'Account 1'. The 'Account Status' is 'Registered'. The 'Account Active' checkbox is checked. The 'Primary SIP Server' is 'ingate-siparator.siparator.com'. The 'Outbound Proxy' is '52.101.100.100'. The 'SIP Transport' is 'UDP'. The 'NAT Traversal' is 'No, but send keep alive'. The 'Label' is '3001'. The 'SIP User ID' is '3001'. The 'Authenticate ID' is '3001'. The 'Authenticate Password' is masked with dots. The 'Name' is '3001'.

Figure 27

- Note the Primary SIP Server is defined using the Domain Name
- Outbound proxy should be pointing to SIParator Public IP address.
- If Domain name FQDN resolves same SIParator Public IP address, then you can just leave Domain Name in Sip Proxy and leave blank the Outbound Proxy,

Snom 870



The screenshot shows the 'Login Information' configuration page for a Snom 870 phone. The 'Identity active' checkbox is checked. The 'Displayname' is '3008'. The 'Account' is '3008'. The 'Password' is masked with dots. The 'Registrar' is 'ingate-siparator.siparator.com'. The 'Outbound Proxy' is '52.101.100.100'. The 'Failover Identity' is 'None'. The 'Authentication Username' is '3008'. The 'Mailbox' is blank. The 'Ringtone' is 'Ringer 1'. The 'Custom Melody URL' is blank. The 'Display text for idle screen' is blank. The 'XML Idle Screen URL' is blank. The 'Ring After Delay (sec)' is blank. The 'Record Missed Calls' checkbox is checked. The 'Record Dialed Calls' checkbox is checked. The 'Record Received Calls' checkbox is checked. The 'Identity is hidden' checkbox is checked. The 'Apply', 'Re-Register', and 'Play Ringer' buttons are visible. The 'Remove Identity' and 'Remove All Identities' buttons are also visible.

Figure 28

- Note the Registrar Server is defined using the Domain Name
- Outbound proxy should be pointing to SIParator Public IP address.
- If Domain name FQDN resolves same SIParator Public IP address, then you can just leave Domain Name in registrar and leave blank the Outbound Proxy,

Grandstream GXV3240

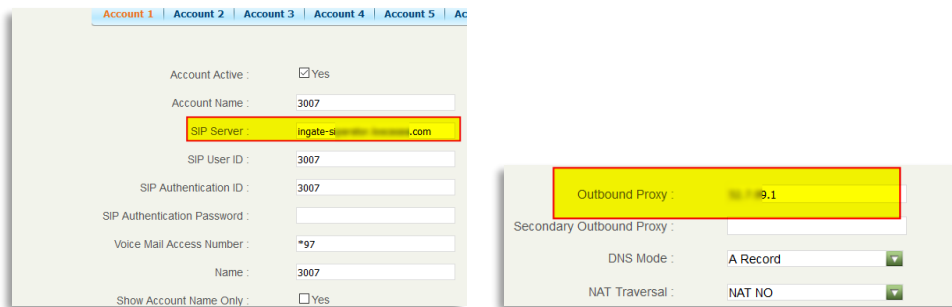


Figure 29

- Note the SIP Server is defined using the Domain Name
- Outbound proxy should be pointing to SIParator Public IP address.
- If Domain name FQDN resolves same SIParator Public IP address, then you can just leave Domain Name in Sip Server and leave blank the Outbound Proxy,

4 Firewall configuration to support PBX non SIP features

In typical deployments, besides to enable SIP signalling thru SIParator/Firewall, it is necessary to enable additional capabilities to be managed thru the Ingate and be properly forwarded. Such is the case of Phone Provisioning for instance. Provisioning might need for example to properly forward TFTP from the outside to the IPPBXIP-PBX in the Inside.

We are going to use as an example an Asterisk Distribution based PBX and will identify In our example the following ports are needed for additional features

Type of use	TCP	FROM
Web Management	80	8080
Web Management (Secure)	443	4343
User Web Access	81, 4443, 8001, 8003	same
WebRTC	8088, 8089	same
SoftClient	8002	same
Operator Panel	58080, 55050	same
HTTP Provisioning	83	same
HTTPS Provisioning	1443	same
OpenVPN Server	1194	same
REST Apps (HTTP)	84	same
REST Apps (HTTPS)	3443	same
XMPP	5222	same
FTP	21	same
TFTP	69	same

Figure 30

Mapping will be done in Rules and relays tag, under Relays:

The screenshot shows the inGate Firewall web interface for AWS Ingate 6.0.1GA 10.0.0.147. The 'Rules and Relays' tab is selected. The 'Relays' configuration page is displayed, showing a table with columns: Listen To (IP Address, Port), Relay To (DNS Name or IP Address, IP Address, Port), Relay Type, and Allow Access (Network). The table lists various port forwarding rules, including those for ports 21, 25, 69, 81, 83, 84, 1443, 2001, 3443, 4343, 4443, 5006, 5007, 5222, 8001-8003, 8080, 8088-8089, and 55050. The 'Listen To' and 'Relay To' columns are highlighted with red boxes, indicating the mapping of ports and IP addresses.

Listen To ...		Relay To ...			Relay Type	Allow Access
IP Address	Port	DNS Name or IP Address	IP Address	Port		
eth0 (10.0.0.147)	21	10.0.1.149	10.0.1.149	21	TCP port forwarding	access
eth0 (10.0.0.147)	25	10.0.1.149	10.0.1.149	25	TCP port forwarding	access
eth0 (10.0.0.147)	69	10.0.1.149	10.0.1.149	69	UDP port forwarding	access
eth0 (10.0.0.147)	81	10.0.1.149	10.0.1.149	81	TCP port forwarding	access
eth0 (10.0.0.147)	83	10.0.1.149	10.0.1.149	83	TCP port forwarding	access
eth0 (10.0.0.147)	84	10.0.1.149	10.0.1.149	84	TCP port forwarding	access
eth0 (10.0.0.147)	1443	10.0.1.149	10.0.1.149	1443	TCP port forwarding	access
eth0 (10.0.0.147)	2001	10.0.1.149	10.0.1.149	2001	TCP port forwarding	access
eth0 (10.0.0.147)	3443	10.0.1.149	10.0.1.149	3443	TCP port forwarding	access
eth0 (10.0.0.147)	4343	10.0.1.149	10.0.1.149	4343	TCP port forwarding	access
eth0 (10.0.0.147)	4443	10.0.1.149	10.0.1.149	4443	TCP port forwarding	access
eth0 (10.0.0.147)	5006	10.0.1.149	10.0.1.149	5006	TCP port forwarding	access
eth0 (10.0.0.147)	5007	10.0.1.149	10.0.1.149	5007	TCP port forwarding	access
eth0 (10.0.0.147)	5222	10.0.1.149	10.0.1.149	5222	TCP port forwarding	access
eth0 (10.0.0.147)	8001-8003	10.0.1.149	10.0.1.149		TCP port forwarding	access
eth0 (10.0.0.147)	8080	10.0.1.149	10.0.1.149	80	TCP port forwarding	access
eth0 (10.0.0.147)	8088-8089	10.0.1.149	10.0.1.149		TCP port forwarding	access
eth0 (10.0.0.147)	55050	10.0.1.149	10.0.1.149	55050	TCP port forwarding	access

Figure 31

- Ports are mapped as per the table, and port overlapping with SIPArator reserved ports such as http and https, they are remapped to alternate ports (8080 and 4343).
- Network from where port forwarding is allowed is the one named "access". (See Figure 6).

In addition, enabling flow traffic policies will be needed to complete Firewall configuration

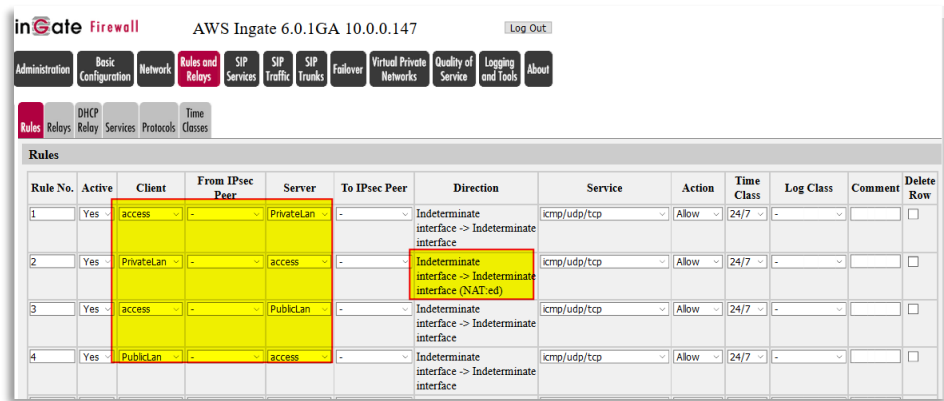


Figure 32

- Note we are using network names we created before (see Figure 6)
- Note that automatically the Policy for PrivateLan to Access is being NATed.
- These policies are the minimum needed to allow traffic flow between remote users and IP-PBX for anything different to SIP signaling and RTP media.

5 Enabling TLS/SRTP

Has the last step to close the full loop, we will enable TLSP/SRTP between Remote extensions and SIParator/Firewall but will leave standard SIP/UDP downstream to the IP-PBX. This way we are delegating anything related to TLS and SRTP to SIParator, unloading that responsibility from the IP-PBX.

Conversion between SIP/UDP \leftrightarrow SIP/TLS as well as media RTP \leftrightarrow SRTP will happen inside SIParator/Firewall.

For simplicity and understanding this case is for teaching purposes we will create a self-signed certificate for TLS

inGate SIParator DE11-8869-1C1A-1614-15D3-E702 Log Out

Administration Basic Configuration Network SIP Services SIP Traffic SIP Trunks Failover Virtual Private Networks Quality of Service Logging and Tools About

Changes have been made to the preliminary configuration, but have not been applied.

Current Certificate

No current certificate.

Create Certificate or Certificate Request

Fill in the certificate data for "tlsvoice" below, then create either a certificate or a certificate request. After generating a certificate request, and having it signed by a signing authority, the certificate must be

Expire in (days): * 365 Country code (C): US Organization (O): Ingate
 Common Name (CN): * ator.loscasas.cc State/province (ST): FL Organizational Unit (OU): Support
 Email address: hesto@ingate.cc Locality/town (L):

SubjectAltName Extension

Enter the alternative names that you want to add to a certificate or a certificate request. Multiple values can be added by using comma separation.

Email:
 URI:
 DNS:
 IP:

Key Length and Signature Algorithm

Select the key length and the signature algorithm that you want to use when creating a certificate or a certificate request.

Key length (bits): 2048
 Signature algorithm: SHA-256

If you generate several certificates with identical data you should make sure they have different serial numbers.

Serial number: * 2

Fields marked with "*" are mandatory.

Create a self-signed X.509 certificate Create an X.509 certificate request Abort

Figure 33

- Make sure you use the Domain FQDN as the Common Name (CN)
- Use the "Create self-signed X.509 certificate"

You will get this:

Changes have been made to the preliminary configuration, but have not been applied.

- Self signed certificate created:
 - Subject: /C=US/emailAddress=ernesto@ingate.com/ST=FL/O=Ingate/OU=Support/CN=ingate-siparator.loscasas.com
 - Issuer: /C=US/emailAddress=ernesto@ingate.com/ST=FL/O=Ingate/OU=Support/CN=ingate-siparator.loscasas.com
 - Serial Number: 2
 - MD5 Fingerprint: 50:10:1F:4C:33:23:96:30:D0:01:CF:D2:40:B0:48:29
 - SHA1 Fingerprint: 802A C9B7 2086 57AA 3A2E 2FD4 B2A0 D17A A8FE D830
 - Valid from 2017-08-03 17:48:09 to 2018-08-03 17:48:09 GMT.
 - Subject Key ID: 73:11:CF:58:8F:79:80:B1:31:1B:2F:37:12:A6:E6:0D:B9:D1:23:49
 - Authority Key ID: 73:11:CF:58:8F:79:80:B1:31:1B:2F:37:12:A6:E6:0D:B9:D1:23:49

Private Certificates [\(Help\)](#)

Name	Certificate			Information	Delete Row
httpsconfig	Create New	Import	View/Download	Subject: /CN=DE11-8869-1C1A-1614-15D3-E702 Issuer: /CN=DE11-8869-1C1A-1614-15D3-E702 MD5 Fingerprint: F0:79:D4:9D:9C:27:D9:E7:93:13:11:47:D9:D4:ED:75 SHA1 Fingerprint: 09AC 2A1C 2EEE 838C EB82 533E 5C5B 1192 0167 429A Valid from: 2017-08-01 21:52:28 Valid to: 2018-08-01 21:52:28 Subject Key ID: A8:12:5F:67:46:07:D5:CB:0D:D6:9A:14:26:1E:48:A8:91:BE:5B:3D Authority Key ID: A8:12:5F:67:46:07:D5:CB:0D:D6:9A:14:26:1E:48:A8:91:BE:5B:3D	<input type="checkbox"/>
tlsvoice	Create New	Import	View/Download	Subject: /C=US/emailAddress=ernesto@ingate.com/ST=FL/O=Ingate/OU=Support/CN=ingate-siparator.loscasas.com Issuer: /C=US/emailAddress=ernesto@ingate.com/ST=FL/O=Ingate/OU=Support/CN=ingate-siparator.loscasas.com MD5 Fingerprint: 50:10:1F:4C:33:23:96:30:D0:01:CF:D2:40:B0:48:29 SHA1 Fingerprint: 802A C9B7 2086 57AA 3A2E 2FD4 B2A0 D17A A8FE D830 Valid from: 2017-08-03 17:48:09 Valid to: 2018-08-03 17:48:09 Subject Key ID: 73:11:CF:58:8F:79:80:B1:31:1B:2F:37:12:A6:E6:0D:B9:D1:23:49 Authority Key ID: 73:11:CF:58:8F:79:80:B1:31:1B:2F:37:12:A6:E6:0D:B9:D1:23:49	<input type="checkbox"/>

Figure 34

Enable TLS on SIP Services

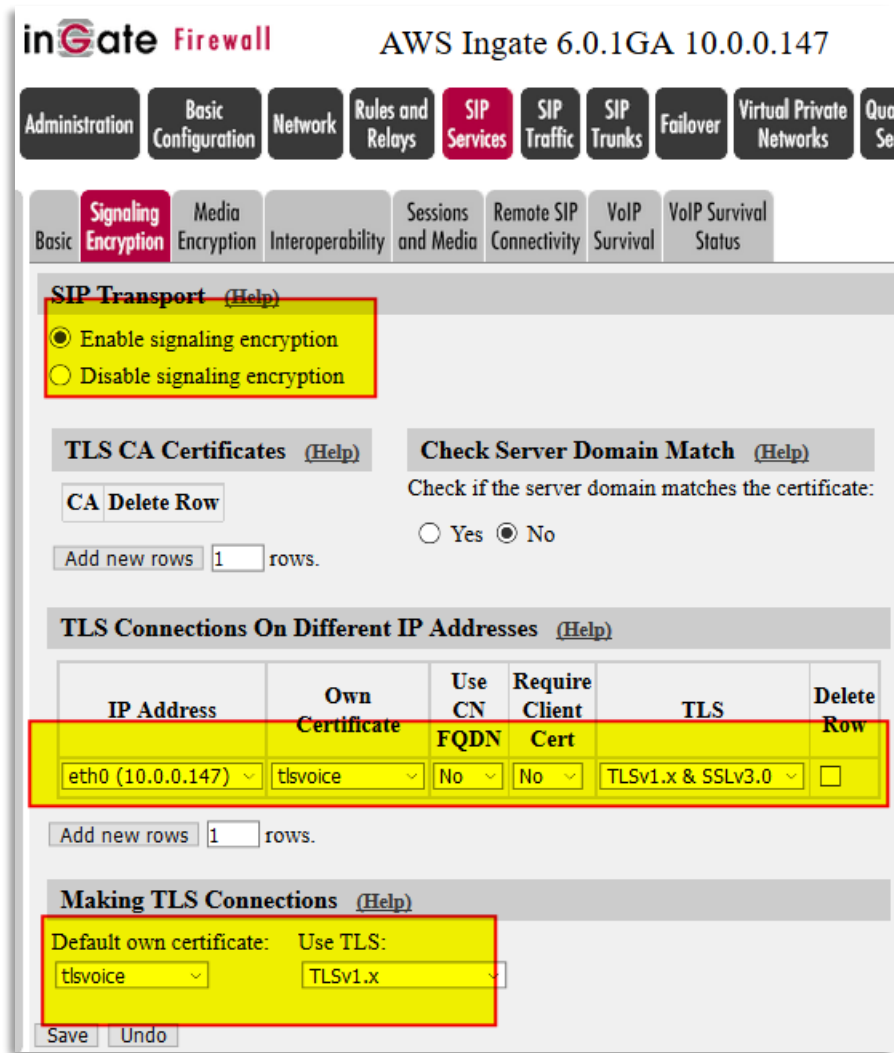


Figure 35

- First of all you need to enable Signaling encryption
- Associate TLS to the external interface, select the certificate. Here you can enforce FQDN validation and Client Required certificate or not. In our case, for Lab purposes we will not enforce them.
- Select type of TLS to use. In our case we will use and accept TLS v1.x and also SSL v3.

Next step will be to enable SRTP to have a fully secure communication between remote users and SIParator/Firewall including Signaling and Media Encryption.

To do so we will go to SIP Services tag → Media Encryption

Basic Signaling Encryption **Media Encryption** Interoperability Sessions and Media Remote SIP Connectivity VoIP Survival VoIP Survival Status

Media Encryption [\(Help\)](#)

Enable media encryption
 Disable media encryption

SIP Media Encryption Policy [\(Help\)](#)

No.	Media Network	Suite Requirements	Allow Transcoding	Delete Row
1	Office	SRTP	Yes	<input type="checkbox"/>

Add new rows: rows.

Default Encryption Policy [\(Help\)](#)

Suite requirements:
Allow transcoding: Yes No

Require TLS [\(Help\)](#)

Require TLS for all cryptos but cleartext
 Do not require TLS

RTP Profile [\(Help\)](#)

Prefer RTP/SAVP (sdescriptions)
 Prefer RTP/AVP (cleartext and legacy encryptions)
 Prefer RTP/AVP (together with sdescriptions)

Multi Profile [\(Help\)](#)

Enable Multi Profile
 Disable Multi Profile

DTLS-SRTP [\(Help\)](#)

Certificate: DTLS to use:
Ignore invalid dates in the client's certificate: Yes No

Keep Established Crypto Within a Dialog [\(Help\)](#)

Keep established crypto within a dialog: Yes No

Add Cryptos in the B2BUA [\(Help\)](#)

Add cryptos in the B2BUA: Yes No

Figure 36

- In our example, we are enabling SRTP for all remote users located in the “Office”, as defines in Networks and Computers (see Figure 6)
- All remaining remotes will not user SRTP in this case, including SIP Trunks

6 Ingate SIParator using AWS VPN Service

This section explains how to create a VPN connection between an Ingate and your VPC on the Amazon cloud service.

As an example, the Ingate will have the public IP address 192.0.2.119 assigned to eth1 and the private network 10.10.10.0/24 connected to eth2.

Furthermore, the AWS VPN endpoints are 192.0.2.200 and 192.0.2.210. The network in the VPC is 10.20.20.0/24.

6.1 AWS VPN Scenarios

There are several scenarios where InGate can be used to connect for instance Remote Offices to have access to Centralized IP-PBX hosted in a VPC.

The following diagrams illustrate single and multiple VPN connections. The VPC has an attached virtual private gateway, and your network includes a customer gateway, which you must configure to enable the VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private gateway.

When you create multiple VPN connections to a single VPC, you can configure a second customer gateway to create a redundant connection to the same external location. You can also use it to create VPN connections to multiple geographic locations.

6.2 Single Office Connection:

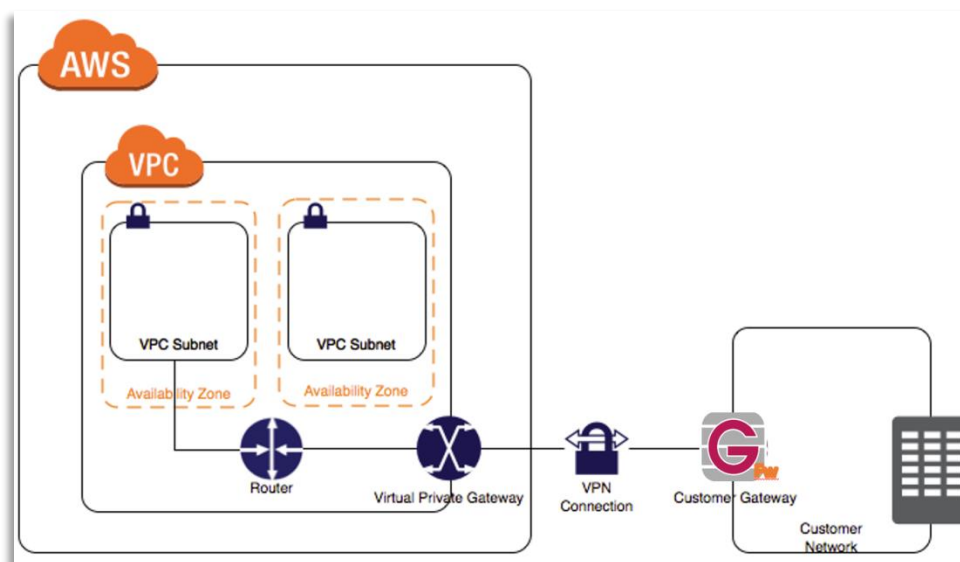


Figure 37

6.3 Multisite with Hosted IP-PBX/UC

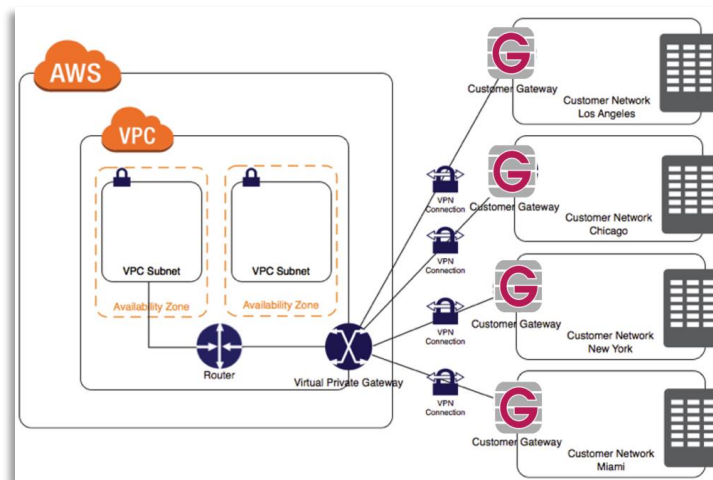


Figure 38

6.4 Configuring Two VPN Tunnels for Your VPN Connection

You use a VPN connection to connect your network to a VPC. Each VPN connection has two tunnels, with each tunnel using a unique virtual private gateway public IP address. It is important to configure both tunnels for redundancy. When one tunnel becomes unavailable (for example, down for maintenance), network traffic is automatically routed to the available tunnel for that specific VPN connection.

The following diagram shows the two tunnels of the VPN connection.

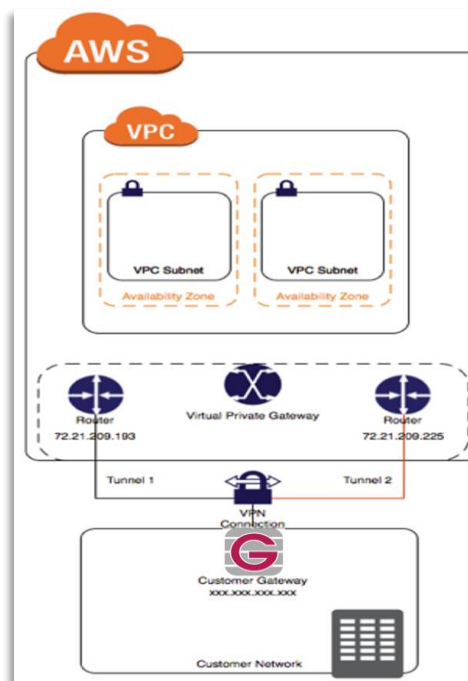


Figure 39

Using Redundant VPN Connections to Provide Failover

As described earlier, a VPN connection has two tunnels to help ensure connectivity in case one of the VPN connections becomes unavailable. To protect against a loss of connectivity in case your customer gateway becomes unavailable, you can set up a second VPN connection to your VPC and virtual private gateway by using a second customer gateway. By using redundant VPN connections and customer gateways, you can perform maintenance on one of your customer gateways while traffic continues to flow over the second customer gateway's VPN connection. To establish redundant VPN connections and customer gateways on your network, you need to set up a second VPN connection. The customer gateway IP address for the second VPN connection must be publicly accessible.

It can be combined with InGate HA capability to have a fully resilient setup

The following diagram shows the two tunnels of each VPN connection and two customer gateways.

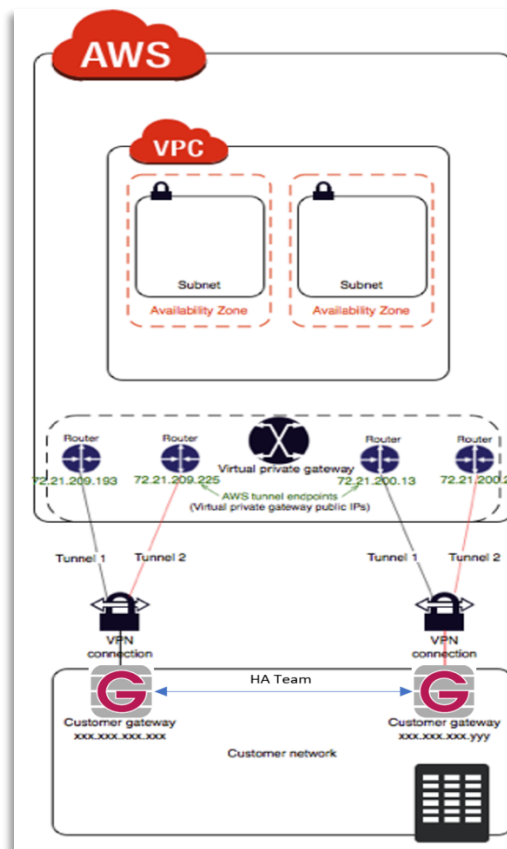


Figure 40

6.4.1 AWS VPN Setup

You can follow [this guide](#) step by step to setup all what you need to enable VPN connection in your VPC.

In short, the following steps need to be taken in order to create a VPN connection via the AWS Management Console:

- 1) Login to the AWS Management Console.
- 2) Go to the VPC service.
- 3) Create a new Customer Gateway.

- a. Name tag: CG_ingate
- b. Routing: Static
- c. IP address: 192.0.2.119
- 4) Create a new Virtual Private Gateway and attach it to VPC.
 - a. Name tag: VPG_ingate
- 5) Create a new VPN Connection.
 - a. Name tag: VPN_ingate
 - b. Virtual Private Gateway: VPG_ingate
 - c. Customer Gateway: GG_ingate
 - d. Routing Options: Static
 - e. Static IP Prefixes: 10.10.10.0/24
- 6) When the connection is created choose Download Configuration and select Vendor Generic.

The downloaded configuration will contain the necessary information to setup the tunnels on the Ingate side. It will contain two IPsec Peer addresses and two pre-shared keys together with connection setup details.

6.4.2 Ingate SIParator VPN Setup

The following Ingate configuration will complete the setup of the VPN connection.

6.4.2.1 IPsec Peers

Go to the IPsec Peers page.

Name	Subgroup	Active	Local Side	Remote Side				ISAKMP Key Lifetime (seconds)	Initiate Re-keying	Encryption	Authentication		Delete Row	
				DNS Name or IP Address	Dynamic	IP Address	RADIUS				Blacklist	Type		Info
* Amazon-IKE-vpn-7		Yes	eth1 (192.0.2.119)	192.0.2.200	<input type="checkbox"/>	192.0.2.200	No		28800	Yes	AES	Preshared secret	Change/View	<input type="checkbox"/>
		Yes	eth1 (192.0.2.119)	192.0.2.210	<input type="checkbox"/>	192.0.2.210	No		28800	Yes	AES	Preshared secret	Change/View	<input type="checkbox"/>

Figure 41

For Tunnel 1:

Name

Select a suitable name. E.g. Amazon-IKE-vpn-74d6a73f-0.

Local Side

Choose interface eth1 (192.0.2.119).

Remote Side

Enter IP address 192.0.2.200 (Virtual Private Gateway for Tunnel 1 in Downloaded Configuration).

ISAKMP Key Lifetime (seconds)

Enter 28800.

Encryption

Select AES.

Authentication

Select Type Pre-shared secret and enter the pre-shared key for Tunnel 1 found in the Downloaded Configuration.

Click the + sign (left to the name Amazon-IKE-vpn-74d6a73f-0) and create

Tunnel 2:

Local Side

Choose interface eth1 (192.0.2.119).

Remote Side

Enter IP address 192.0.2.210 (Virtual Private Gateway for Tunnel 2 in Downloaded Configuration).

ISAKMP Key Lifetime (seconds)

Enter 28800.

Encryption

Select AES.

Authentication

Select Type Pre-shared secret and enter the pre-shared key for Tunnel 2 found in the Downloaded Configuration.

6.4.2.2 IPsec Tunnels

Go to page IPsec Tunnels. In the table IPsec Networks add the lan 10.10.10.0/24 and the lan_vpc 10.20.20.0/24.

Name	DNS Name or Network Address	Network Address	Netmask / Bits	Delete Row
lan	10.10.10.0	10.10.10.0	24	<input type="checkbox"/>
lan_vpc	10.20.20.0	10.20.20.0	24	<input type="checkbox"/>

Add new rows rows.

Figure 42

In the IPsec Tunnels table add a new IPsec tunnel.

Peer	Local Network			Remote Network		IPsec Key Lifetime (seconds, optional)	Encryption	PFS Group	Delete Row
	Address Type	Network	NAT As	Address Type	Network				
Amazon-IKE-vpn-74d6a73f-0	Network	lan	-	Network	lan_vpc	3600	AES	Same as Phase 1 DH	<input type="checkbox"/>

Add new rows groups with rows per group.

Figure 43

Peer

Select the peer you created on the IPsec Peers page.

Local Network

Select Address Type Network and select the Network lan.

Remote Network

Select Address Type Network and select the Network lan_vpc.

IPsec Key Lifetime (seconds, optional)

Enter 3600.

Encryption

Select AES.

PFS Group

Select Same as Phase 1 DH

6.4.2.3 IPsec Advanced

Go to the page **IPsec Advanced** and create a new entry in the **IPsec Peers** table.

Peer	NAT Traversal	Dead Peer Detection				Delete Row
		Enabled	Delay	Timeout	Action	
Amazon-IKE-vpn-74d6a73f-0	Force	Yes	10	30	Restart	<input type="checkbox"/>

Add new rows rows.

Figure 44

Peer

Select the peer you created on the IPsec Peers page.

NAT Traversal

Select Force.

Dead Peer Detection

Select Enabled Yes. Enter 10 in Delay and 30 in Timeout.

Action

Restart.

6.4.2.4 Networks and Computers

Go to the page **Networks and Computers** and add two networks, *lan* 10.10.10.0-10.10.10.255 on interface Ethernet2 (eth2 untagged) and *lan_vpc* 10.20.20.0-10.20.20.255

Name	Subgroup	Lower Limit		Upper Limit (for IP ranges)		Interface/VLAN	Delete Row
		DNS Name or IP Address	IP Address	DNS Name or IP Address	IP Address		
lan	.	10.10.10.0	10.10.10.0	10.10.10.255	10.10.10.255	Ethernet2 (eth2 untagged)	<input type="checkbox"/>
lan_vpc	.	10.20.20.0	10.20.20.0	10.20.20.255	10.20.20.255	.	<input type="checkbox"/>

Add new rows groups with rows per group.

Figure 45

Rules

Go to page **Rules** and create two rules to allow traffic from and to the Amazon VPN tunnel. These rules will allow all TCP, UDP and ICMP traffic to and from the tunnel

Rule No.	Active	Client	From IPsec Peer	Server	To IPsec Peer	Direction	Service	Action	Time Class	Log Class	Comment	Delete Row
1	Yes	lan_vpc	.	lan_vpc	Amazon-IKE-vpn-74d6a73f-0	Ethernet2 -> (VPN)	icmp/udp/tcp	Allow	24/7	.		<input type="checkbox"/>
2	Yes	lan_vpc	Amazon-IKE-vpn-74d6a73f-0	lan_vpc	.	(VPN) -> Ethernet2	icmp/udp/tcp	Allow	24/7	.		<input type="checkbox"/>

Figure 46

6.4.2.5 Save/Load Configuration

Finally, go to the **Save/Load Configuration** page under **Administration** and apply the new settings by pressing **Apply configuration**.

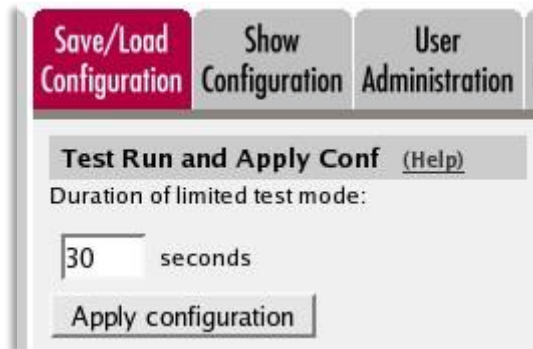


Figure 47

When the configuration is applied you should, under page **IPsec Status** see that one of the two tunnels for peer *Amazon-IKE-vpn-74d6a73f-0* is up.

7 Using Ingate Startup Tool TG

7.1 Introduction

The Ingate Startup Tool TG is designed to simplify the initial “out of the box” configuration of your Ingate Unit. The tool will automatically configure your Ingate Firewall or SIParator to work with the IP-PBX, SIP trunking service provider of your choice, and sets up all the routing needed to enable remote users to access and use the enterprise IP-PBX.

The Ingate Startup Tool TG is free of charge for all Ingate Firewalls and SIParators. Get the latest version of it at

http://www.ingate.com/Startup_Tool_TG.php

For the Ingate SUTG to be able to access the Ingate SIParator Instance launched in your VPC you will need to set up the tool in one of the following ways:

- Install the Ingate Startup Tool TG in a Windows Instance located in the same VPC and subnet where the SIParator was launched.
- Install the Ingate Startup Tool TG on your PC, and make sure you have VPN access to the VPC and subnet where the SIParator was launched.
- Install the Ingate Startup Tool TG on your PC, and make sure your Instance has Internet access and is accessible from the outside via a public IP address (Elastic IP).

Once the Instance has been launched and before you try to use the Startup Tool, you MUST do the steps to install and activate the licenses.

Make sure you have proper connectivity between the Windows machine hosting the Startup Tool and the SIParator Instance. (More details in the next sections)

7.2 Setting up connectivity.

Startup tool uses port TCP 80 to communicate with the SIParator. We will here show what needs to be done to grant connectivity via TCP 80.

Assuming you have already installed and activate proper licensing, you will then setup networking to enable connectivity with the Tool. Follow these steps:

- Access SIParator GUI
- Go to Access Control under Basic Configuration Section
- Add HTTP protocol in the Configuration Transport Section
- Add IP address or subnet from which you will be accessing in the Configuration Computers section.

inGate SIParator CDD6-C5FA-67A6-4FFE-0D29-10A1 [Log Out](#)

Administration Basic Configuration Network SIP Services SIP Traffic SIP Trunks Failover Virtual Private Networks Quality of Service Logging and Tools About

Basic Configuration Access Control RADIUS SNMP Dynamic DNS Update Certificates TLS Advanced SIParator Type

Configuration Allowed Via Interface [\(Help\)](#)

Interface or Tunnel Allowed Delete Row
 Ethernet0 (eth0) Yes

Add new rows: 1 rows.

Configuration Transport [\(Help\)](#)

Protocol	IP Address	Port	Cert	TLS	Delete Row
HTTP	eth0 (eth0)	80	-	-	<input type="checkbox"/>
HTTPS	eth0 (eth0)	443	httpsconfig	TLSv1.x	<input type="checkbox"/>
SSH	-	22	-	-	<input type="checkbox"/>

Add new rows: 1 rows.

User Authentication For Web Interface Access [\(Help\)](#)

Local users
 RADIUS database
 Local users or RADIUS database

Web Interface Access Settings [\(Help\)](#)

Login timeout: 600 seconds

Configuration Computers [\(Help\)](#)

No.	DNS Name or Network Address	Network Address	Netmask / Bits	Range	Via IPsec Peer	SSH	HTTP	HTTPS
1	0.0.0.0	0.0.0.0	0	0.0.0.0 - 255.255.255.255	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	192.168.200.0	192.168.200.0	24	192.168.200.0 - 192.168.200.255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add new rows: 1 rows.

Figure 48

In our example we are enabling connectivity from the network 192.168.200.0/24, and the connection is established via VPN with the VPC where the SIParator has been launched. Note the HTTP box for the originator network is being selected (enabled).

If you, for instance, are planning to establish connectivity from the outside (Internet), you will need to associate an Elastic IP address with eth0 using AWS console, and add your origin public IP address in the Configuration Computers Section.

In case you are running the Startup Tool from a Windows machine in the same VPC (i.e. 10.0.0.0/16) you will use the same network address in the added Configuration Computers row.

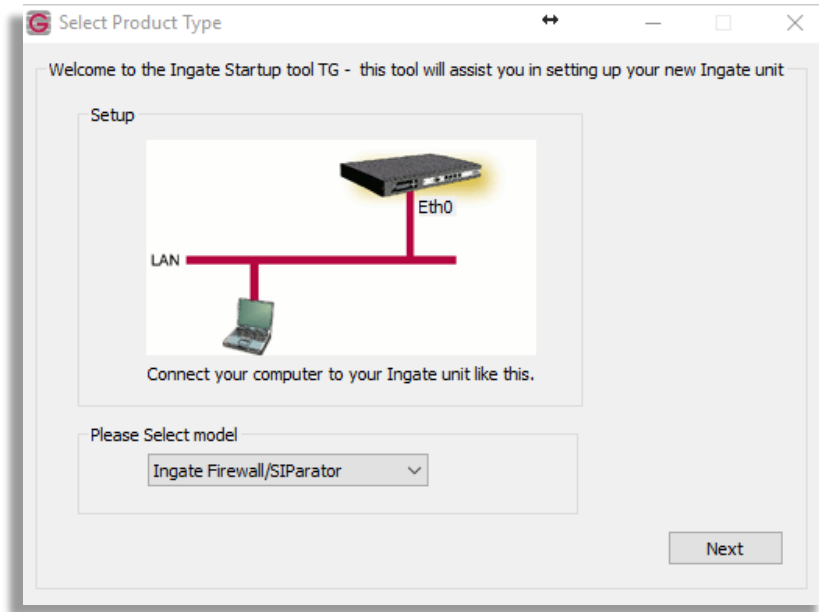


Figure 49

After selecting Ingate Firewall/SIParator option:

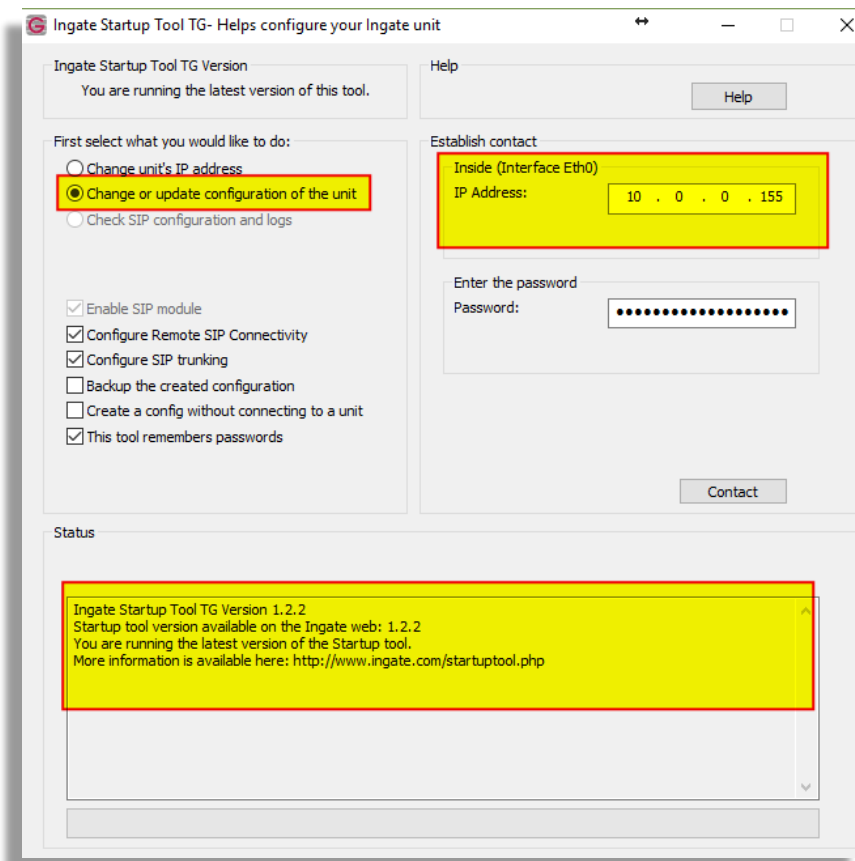


Figure 50

Make sure you select “Change or Update configuration of the unit”. As AWS controls and have full ownership of IP address, you won’t be able to assign IP address to any interface.

If you are accessing from the local network or VPN you will need to use the Internal assigned IP address to eth0 in the Instance.

If you are accessing from the outside, you will need to use the public Elastic IP address associated with eth0.

For detailed instructions on how to use the Startup Tool please refer to the manual here:



Figure 51

8 Additional help or support

If you have questions, suggestions and any other concern feel free to contact Educronix LLC

Web: www.educronix.com

Email: support@educronix.com

Toll-Free: +1 855 866 8854

Ph: +1 954 866 8884

We also provide consulting services as well as remote hands troubleshooting and configuration.