# Twilio integration with Avaya Aura Contact Center (AACC)

July 2023

## Document History

| Rev. No. | Date | Description |
|---|---|---|
| 1.0 | April-24-2023 | Configuration Guide |
| 1.1 | April-28-2023 | Updated review comments |
| 1.1(5) | July-07-2023 | Final version |

# Table of Contents


1 Audience 10
2 Introduction 10
2.1 tekVizion Labs 10
2.2 Use Case 11
2.3 Twilio Studio 13
2.4 Avaya Aura Contact Center 13
3 Topology 14
3.1 Hardware Components 15
3.2 Software Requirements 15
4 Pre-requisites 16
4.1 Twilio 16
4.2 Avaya Aura Contact Center 16
5 Avaya Aura Contact Center Call Flow 17
6 Configuration 18
6.1 Twilio Configuration 18
6.1.1 SIP domains 18
6.1.2 IP access control lists 21
6.1.3 Credential lists 22
6.1.4 Twilio Studio IVR Flow 23
6.1.5 Phone Numbers 52
6.1.6 Functions 54
6.1.7 Services 61
6.2 Avaya SBCE Configuration 63
1.1.1 Avaya SBCE Login 63
1.1.2 Server Interworking 64
1.1.3 SIP Servers 66
1.1.4 Topology Hiding 72
1.1.5 Routing 73
1.1.6 Media Rules 75
1.1.7 Signaling Rules 78
1.1.8 End Point Policy Groups 80
1.1.9 Network Management 81
1.1.10 Media Interface 83
1.1.11 Signaling Interface 85
1.1.12 End Point Flows 87

1.1.13 Signaling Manipulation 90
1.2 Avaya Aura Session Manager Configuration 91
1.2.1 Avaya Aura SM Login 91
1.2.2 Domain 92
1.2.3 Locations 93
1.2.4 Adaptations 95
1.2.5 SIP Entities and Entity Links 100
1.2.6 Routing Policies 105
1.2.7 Dial Patterns 106
1.3 Avaya Aura Communication Manager Configuration 108
1.3.1 Avaya Aura CM Login 108
1.3.2 IP Node Name 109
1.3.3 IP Codec Set 110
1.3.4 IP Network Region 111
1.3.5 Signaling Group 112
1.3.6 Trunk Group 113
1.3.7 Station 116
1.3.8 Application Enablement Services (AES) Configuration 117
1.4 Application Enablement Services (AES) Configuration 121
1.4.1 AES Login 121
1.4.2 Communication Manager Switch Connections 122
1.4.3 Communication Manager Switch Connections CLAN IP 123
1.4.4 CTI link to Communication Manager 124
1.4.5 TR87 on the AES 125
1.4.6 Security on the AES 126
1.4.7 Certificate Authority root certificate in to AES 127
1.4.8 AES Certificate Signing Request 128
1.4.9 Signed Certificate in to AES 130
1.4.10 Avaya Aura Contact Server as a trusted host on AES 131
1.4.11 Certificate Store in Avaya Aura CC 132
1.5 Avaya Aura Media Server (AAMS) Configuration 139
1.5.1 Nodes and Routes 140
1.6 Avaya Aura Contact Center Configuration 141
1.6.1 Avaya Aura Contact Center Login 141
1.6.2 System Control and Monitor Utility 142
1.6.3 Server Configuration 143

1.6.4 SGM Management Client 148
1.6.5 Avaya Aura Contact Center (CC) Manager Configuration 149
1.6.6 CDNs (Route Point) Configuration 150
1.6.7 Media Server 151
1.6.8 Media Server Configuration 152
1.6.9 CCT Server 153
1.6.10 Contact Center Agent 155
1.6.11 TFE REST Configurator 158
1.6.12 Avaya Aura Orchestration Designer Application Script 160
1.6.13 CCMM (Contact Center Multimedia) Administration 172
1.6.14 Avaya Aura Agent Desktop 175

## Table of Figures

*Figure 1 Flowchart View of a use case* 12
*Figure 2 Network Topology* 14
*Figure 3 Call Flow* 17
*Figure 4 SIP domains* 19
*Figure 5 SIP domains Continuation* 20
*Figure 6 IP access control lists* 21
*Figure 7 Credential lists* 22
*Figure 8 Twilio IVR Studio Flow – Create New Flow* 23
*Figure 9 Twilio IVR Studio Flow – Template* 24
*Figure 10 Twilio IVR Studio Flow – End to End Flow* 25
*Figure 11 Twilio IVR Studio Flow – set_lang_eng* 27
*Figure 12 Twilio IVR Studio Flow – set_lang_eng Continuation* 28
*Figure 13 Twilio IVR Studio Flow – say_company_name* 29
*Figure 14 Twilio IVR Studio Flow – prompt_language_es* 30
*Figure 15 Twilio IVR Studio Flow – language_check* 31
*Figure 16 Twilio IVR Studio Flow – language_check* 32
*Figure 17 Twilio IVR Studio Flow – language_check Continuation* 33
*Figure 18 Twilio IVR Studio Flow – no_input_handler_1* 34
*Figure 19 Twilio IVR Studio Flow – split_1* 35
*Figure 20 Twilio IVR Studio Flow – english_error_msg* 36
*Figure 21 Twilio IVR Studio Flow – prompt_for_english* 37
*Figure 22 Twilio IVR Studio Flow – prompt_language* 38
*Figure 23 Twilio IVR Studio Flow – split_key_press_en* 39
*Figure 24 Twilio IVR Studio Flow – add_queue_to_sync* 40
*Figure 25 Twilio IVR Studio Flow – add_queue_to_sync Continuation* 41
*Figure 26 Twilio IVR Studio Flow – connect_call_to_sales* 42
*Figure 27 Twilio IVR Studio Flow – connect_call_to_sales Continuation* 43
*Figure 28 Twilio IVR Studio Flow – no_input_handler_3* 44
*Figure 29 Twilio IVR Studio Flow – gather_input_english* 45

*Figure 30 Twilio IVR Studio Flow – no_input_handler_1* 46
*Figure 31 Twilio IVR Studio Flow – prompt_language* 47
*Figure 32 Twilio IVR Studio Flow – split_speech_result_en* 48
*Figure 33 Twilio IVR Studio Flow – add_queue_speech_to_sync* 49
*Figure 34 Twilio IVR Studio Flow – connect_call_to_support* 50
*Figure 35 Twilio IVR Studio Flow – no_input_handler_4* 51
*Figure 36 Phone Numbers* 52
*Figure 37 Phone Number Continuation* 53
*Figure 38 Functions – Configure Service* 54
*Figure 39 Functions – Create function* 54
*Figure 40 Functions – add_to_sync* 55
*Figure 41 Functions – add_to_sync Continuation* 56
*Figure 42 Functions – Environment Variables* 58
*Figure 43 Functions – Dependencies* 59
*Figure 44 Functions – Deploy All* 60
*Figure 45 Sync Service* 61
*Figure 46 Sync Maps* 62
*Figure 47 Avaya SBCE Login* 63
*Figure 48 Selection of Avaya SBCE Device* 63
*Figure 49 Server Interworking Profile for Avaya SM* 64
*Figure 50 Server Interworking Profile for Avaya SM Continuation* 64
*Figure 51 Server Interworking Profile for Avaya SM Continuation* 65
*Figure 52 SIP Server for Avaya SM* 66
*Figure 53 SIP Server for Avaya SM Continuation* 67
*Figure 54 SIP Server for Avaya SM Continuation* 68
*Figure 55 SIP Server for Twilio* 69
*Figure 56 SIP Server for Twilio Continuation* 69
*Figure 57 SIP Server for Twilio Continuation* 70
*Figure 58 SIP Server for Twilio Continuation* 70
*Figure 59 SIP Server for Twilio Continuation* 71
*Figure 60 Topology Hiding Profile for Twilio* 72
*Figure 61 Routing for Avaya SM* 73
*Figure 62 Routing for Twilio* 74
*Figure 63 Media Rules for Twilio* 75
*Figure 64 Media Rules for Twilio Continuation* 76
*Figure 65 Media Rules for Twilio Continuation* 77
*Figure 66 Signaling Rules for Avaya SM* 78
*Figure 67 Signaling Rules for Avaya SM Continuation* 79
*Figure 68 End Point Policy Group for Twilio* 80
*Figure 69 End Point Policy Group for Avaya SM* 80
*Figure 70 Network Interface for Avaya SM* 81
*Figure 71 Network Interface for Twilio* 82
*Figure 72 Media Interface for Twilio* 83
*Figure 73 Media Interface for Avaya SM* 84
*Figure 74 Signaling Interface for Avaya SM* 85
*Figure 75 Signaling Interface for Twilio* 86
*Figure 76 End Point Flows – Server Flows* 87

*Figure 77 End Point Flow for Avaya SM* 88
*Figure 78 End Point Flow for Twilio* 89
*Figure 79 Sigma script towards Avaya SM* 90
*Figure 80 Avaya SM Login* 91
*Figure 81 Routing* 92
*Figure 82 Add Domain* 92
*Figure 83 Domain* 93
*Figure 84 Locations* 93
*Figure 85 Locations Continuation* 93
*Figure 86 Locations continuation* 94
*Figure 87 Adaptation for Avaya Aura Contact Center* 95
*Figure 88 Regular Expression for Avaya Aura Contact Center* 96
*Figure 89 Regular Expression for Avaya Aura Contact Center Continuation* 97
*Figure 90 Regular Expression for Avaya Aura Contact Center Continuation* 98
*Figure 91 Adaptation for Avaya Aura CM* 99
*Figure 92 SIP Entity for Avaya SM* 100
*Figure 93 SIP Entity and Entity Link for Avaya CM* 101
*Figure 94 SIP Entity and Entity Link for Avaya CM Continuation* 102
*Figure 95 SIP Entity and Entity Link for Twilio* 103
*Figure 96 SIP Entity and Entity Link for Twilio Continuation* 103
*Figure 97 SIP Entity and Entity Link for Avaya Aura CC* 104
*Figure 98 Routing Policy for Avaya Aura CM* 105
*Figure 99 Routing Policy for Avaya Aura CC* 105
*Figure 100 Dial Pattern for Avaya Aura CM* 106
*Figure 101 Dial Pattern for Avaya Aura CC* 107
*Figure 102 Avaya Aura CM Login* 108
*Figure 103 IP Node Name* 109
*Figure 104 IP Codec Set* 110
*Figure 105 IP Network Region* 111
*Figure 106 Signaling Group* 112
*Figure 107 Trunk Group* 113
*Figure 108 Trunk Group Continuation* 114
*Figure 109 Trunk Group Continuation* 115
*Figure 110 Add Station* 116
*Figure 111 List Node Names - AES IP Address* 117
*Figure 112 AES IP Services* 118
*Figure 113 AES IP Services Continuation* 119
*Figure 114 AES CTI Link* 120
*Figure 115 AES Login* 121
*Figure 116 AES Avaya CM Switch Connections* 122
*Figure 117 AES Avaya CM Switch Connections Continuation* 122
*Figure 118 AES Avaya CM Switch Connections CLAN IP* 123
*Figure 119 AES Avaya CM Switch Connections CLAN IP Continuation* 123
*Figure 120 Adding a CTI Link* 124
*Figure 121 Enabling TR87 Port* 125
*Figure 122 Configuring Security* 126
*Figure 123 Importing Root CA in to AES* 127

*Figure 124 Importing Root CA in to AES Continuation* 127
*Figure 125 Generating AES CSR* 128
*Figure 126 Generating AES CSR Continuation* 129
*Figure 127 Importing Signed Certificate in to AES* 130
*Figure 128 Importing Signed Certificate in to AES Continuation* 130
*Figure 129 Adding Avaya Aura CC as a trusted host* 131
*Figure 130 Adding Avaya Aura CC as a Trusted host Continuation* 131
*Figure 131 Security Manager* 132
*Figure 132 Security Manager Login* 132
*Figure 133 Security Manager – Security Store* 133
*Figure 134 Security Manager – Certificate Request* 134
*Figure 135 Security Manager – Add Certificate* 135
*Figure 136 Security Manager – Security Configuration* 136
*Figure 137 Security Manager – Store Maintenance* 137
*Figure 138 Security Manager – Display Certificates* 138
*Figure 139 Avaya Aura Media Server Login* 139
*Figure 140 Avaya Aura Media Server – Add Trusted Node* 140
*Figure 141 Avaya Aura CC Login* 141
*Figure 142 System Control and Monitor Utility* 142
*Figure 143 Server Configuration – Local Settings* 143
*Figure 144 Server Configuration – Licensing* 144
*Figure 145 Server Configuration – SIP Network Settings* 145
*Figure 146 Server Configuration – SIP Local Subscriber* 146
*Figure 147 Server Configuration – CCT Server* 147
*Figure 148 SGM Management Client* 148
*Figure 149 Avaya Aura CC Manager Login* 149
*Figure 150 Launchpad* 150
*Figure 151 CDNs (Route Point) configuration* 150
*Figure 152 Add Media Server* 151
*Figure 153 Media Server Configuration* 152
*Figure 154 Add CCT Server* 153
*Figure 155 Add CCT Server Continuation* 154
*Figure 156 Contact Center Agent* 155
*Figure 157 Contact Center Agent Continuation* 156
*Figure 158 Contact Center Agent Continuation* 157
*Figure 159 Contact Center Agent Continuation* 157
*Figure 160 TFE REST Configurator* 158
*Figure 161 TFE REST Configurator Continuation* 159
*Figure 162 Launch Orchestration Designer* 160
*Figure 163 Orchestration Designer Login* 161
*Figure 164 Orchestration Designer – Create New Application* 161
*Figure 165 Orchestration Designer – Create New Application Continuation* 162
*Figure 166 Orchestration Designer – Graphical Flow* 163
*Figure 167 Orchestration Designer – Graphical Flow Continuation* 164
*Figure 168 Orchestration Designer – Graphical Flow Continuation* 164
*Figure 169 Orchestration Designer – Graphical Flow Continuation* 165
*Figure 170 Orchestration Designer – Graphical Flow Continuation* 166

*Figure 171 Orchestration Designer – Graphical Flow Continuation* 166
*Figure 172 Orchestration Designer – Graphical Flow Continuation* 167
*Figure 173 Orchestration Designer – Graphical Flow Continuation* 168
*Figure 174 Orchestration Designer – Graphical Flow Continuation* 169
*Figure 175 Orchestration Designer – Activate Application* 170
*Figure 176 Orchestration Designer – Master Script mapping with Application* 171
*Figure 177 CCMM* 172
*Figure 178 CCMM Continuation* 172
*Figure 179 CCMM Continuation* 173
*Figure 180 CCMM Continuation* 174
*Figure 181 Avaya Aura Agent Desktop* 175
*Figure 182 Avaya Aura Agent Desktop Continuation* 175

# 1 Audience

This document is intended for technical staff and Value Added Resellers (VAR) with installation and operational responsibilities. This configuration guide provides steps for integrating **Twilio Studio**, a tool to build IVRs and chatbots, with **Avaya Aura Contact Centre** in a Contact Center environment

# 2 Introduction

## 2.1 tekVizion Labs

tekVizion Labs™ is an independent testing and Verification facility offered by tekVizion PVS, Inc. ("tekVizion"). tekVizion Labs offers several types of testing services including:

- Remote Testing – provides secure, remote access to certain products in tekVizion Labs for pre-Verification and ad hoc testing
- Verification Testing – Verification of interoperability performed on-site at tekVizion Labs between two products or in a multi-vendor configuration
- Product Assessment – independent assessment and verification of product functionality, interface usability, assessment of differentiating features as well as suggestions for added functionality, stress and performance testing, etc.

tekVizion is a systems integrator specifically dedicated to the telecommunications industry. Our core services include consulting/solution design, interoperability/Verification testing, integration, custom software development and solution support services. Our services helps service providers achieve a smooth transition to packet-voice networks, speeding delivery of integrated services. While we have expertise covering a wide range of technologies, we have extensive experience surrounding our practice areas which include: SIP Trunking, Packet Voice, Service Delivery, and Integrated Services.

The tekVizion team brings together experience from the leading service providers and vendors in telecom. Our unique expertise includes legacy switching services and platforms, and unparalleled product knowledge, interoperability and integration experience on a vast array of VoIP and other next-generation products. We rely on this combined experience to do what we do best: help our clients advance the rollout of services that excite customers and result in new revenues for the bottom line. tekVizion leverages this real-world, multi-vendor integration and test experience and proven processes to offer services to vendors, network operators, enhanced service providers, large enterprises and other professional services firms. tekVizion's headquarters, along with a state-of-the-art test lab and Executive Briefing Center, is located in Plano, Texas.

*For more information on tekVizion and its practice areas, please visit tekVizion Labs website at [www.tekVizion.com](http://www.tekVizion.com)*

## 2.2 Use Case

This Configuration Guide "blueprint," while modeled after a specific customer use case, is generic enough that it can represent almost any attempt to front-end a customer's Avaya Aura Contact Center with a customer-built customized/bespoke modern, cloud-based IVR (Interactive Voice Response) from Twilio, where the sending of entered or queried data to Avaya is needed.

The Twilio IVR in this scenario can be a basic, DTMF-based IVR, a speech rec-enabled-IVR, or a fully conversational IVR built with an AI-based virtual agent (such as Google's DialogFlow CX) on Twilio, as the customer may prefer to build – and its job done (as successfully validated by this blueprint) is to replace the existing IVR built into a customer's Avaya system, while still providing the necessary context for calls being redirected to the Avaya Aura Contact Center, so that an agent can process the remaining incoming calls still being escalated, same as before, if the call cannot be dealt with via new self-service modalities as enabled by the customer's more modern Twilio IVR, versus all calls needing to be routed to an agent, just because some of those calls required assistance by a live human being.

In this particular case (see Figure 1, below), we've modeled our scenario on that of a typical customer who contacted Twilio for help, because they had an IT department with agents providing phone technical support, from agent stations hung off their Avaya Aura Contact Center platform, and who were getting flooded with password reset calls (a very common and persistent support need in many corporate IT environments, but calls that are nevertheless sometimes tricky to handle in an automated manner). While most password reset calls would usually addressable via stepping callers through some new, more interactive info-gathering and self-service directions provided via a newly built custom Twilio IVR flow, one that could include two-factor authentication and other enterprise data dipping measures for added security (or personalization), a calls few would still need some "hand-holding" or trouble-shooting by agents – but not so many that if most calls could be handled self-service, agents would not then have a lot more time to process more complicated calls, or do other tasks.

In the case of this typical customer, their existing Avaya IVR system also had several limitations leaving it unable to address this use case by itself – such as the inability to also send text messages (with helpful self-service or TFA - Two Factor Authentication - links), an inability (at least not without expensive upgrades) to add speech rec or conversational IVR capabilities, the practical inability for the IVR to be shared across multiple sites / Aura on-premises installations from the cloud, difficulty in making changes and be centrally managed from the cloud -- all leading this customer to want to front-end their Avaya Contact Center system with something more modern, which they could customize as needed, and build for themselves in the cloud, on Twilio, servicing all of their locations and enabling the IT department of this customer to provide higher degree of self-service to their end-users.

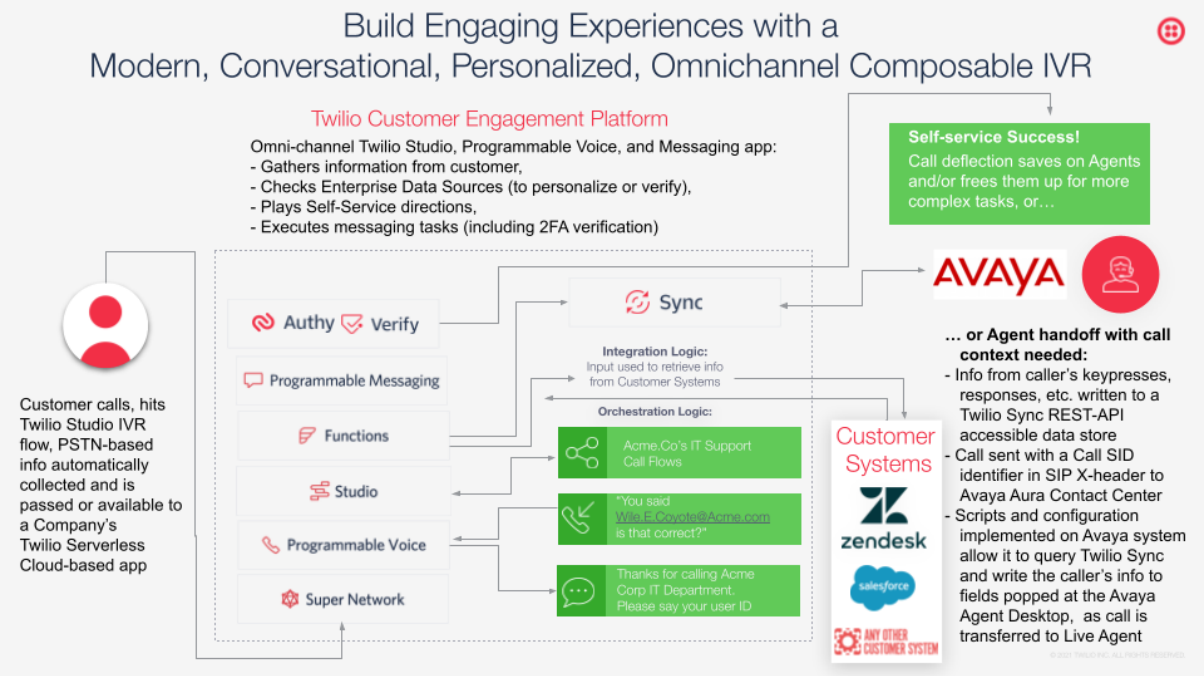



*Figure 1 Flowchart View of a use case*

When we therefore began constructing our Configuration Guide with this use case in mind, we were seeking to specifically address a typical call flow similar to that faced by a caller into this company's IT department needing help resetting their password, as follows:

- The caller would first call into a Twilio IVR (same phone number, owned by or ported to Twilio, of the company's IT department) similar to the one created here, and enter various data, such as giving their username or email and credentials to the Twilio IVR for a lookup.
- Then the caller would be directed to some self-service directions, delivered via played prompts and/or text messages sent, for resetting their password from the IVR, if that was their issue, (including steps such as being sent a TFA verification message or link to their mobile device on record in the company's database for that user, as an added layer of security) from the IVR, and,
- If that solved the caller's problem, in a self-service/automated manner, then… success (as in most cases)!
- Else… in cases where the verification link, lookup or new password failed, if the caller still needed support, then the caller could elect to have their call sent (queued) at that point to a live IT Support Agent – and most importantly, along with their call, have all of the context input into the Twilio platform and IVR up to that point sent also, such as keypresses and whatever number the caller had called from, language preference, what login credentials they had provided, what number the TFA message was sent to and link sent, success/fail result of reset attempt, etc. – so that the the IT support agent on the Avaya Contact Center eventually picking up the call would have all of that context already in front of them upon answering in their Avaya Agent desktop application, without the need for the caller to repeat or to the agent re-gather this information

While not every literal step in the above flow is covered here in this Configuration Guide (it would be too long), the rest can predominantly be found among Twilio's Quickstart guides

online. The link for Twilio's Quickstart guides online is https://www.twilio.com/docs/voice. However, the basic framework and steps for how to collect and successfully pass entered call context from a customer-built Twilio app to an Avaya on-premise Aura Contact Center system are layed out here, step by step.

The main tools used to build this validated call flow include Twilio Studio, and the Twilio Console interface to other Twilio services and tools, as well as the Avaya Aura Contact Center's various admin interfaces, as described below.

One interesting key wrinkle solved in validating this Configuration Guide was that the Avaya Aura Contact Center, as is typical for an on-premises contact center system, naturally assumes the existence of an on-premise database from which info like call context would be assembled to be sent on to agents (instead of all the info arriving with the call itself from off-board) – logical enough in the "old days" of single-vendor vertically integrated on-premise systems, but limiting the Avaya system's usefulness and extensibility today – whereas more modern, multi-party cloud-based bespoke and/or customer-built solutions typically use data sources also centralized in the cloud for sending data with arriving calls as they are passed around. Read on below to see how ancillary Twilio tools like Twilio Sync have enabled the key "cloud database-to-premises system connectivity" aspect necessary to get this configuration working for Avaya customers.

## 2.3 Twilio Studio

Twilio Studio is a visual, drag-and-drop editor for creating applications. It helps to build an IVR system that gathers key presses from callers and connects their calls to an agent at a Contact Center. When using Twilio Studio, customers will also often use Twilio Functions, Assets, Sync, and other aspects of Twilio's platform in building their applications.

The Twilio product mentions should have links as follows:

- Twilio Studio: https://www.twilio.com/en-us/serverless/studio
- Twilio Functions: https://www.twilio.com/en-us/serverless/functions
- Assets: https://www.twilio.com/docs/serverless/functions-assets/assets
- Sync: https://www.twilio.com/docs/sync/api

Additionally, Twilio Studio comes with many "one-click" built-in widgets that make adding in partner capabilities, such as linking to Google DialogFlow CX AI-based Virtual Agents for Conversational IVRs, much easier.

https://twilio.postclickmarketing.com/on-demand-level-up-with-twilio-voice-google-dialogflow-cx-building-conversational-ai-experiences

## 2.4 Avaya Aura Contact Center

Avaya Aura Contact Center (AACC) is a solution that delivers intelligent call routing, network-to-desktop Computer Telephony Integration (CTI), and multichannel contact management to contact center agents over an IP network – including Avaya AES, Avaya Aura MM, Avaya Aura CM, Avaya Aura SM, Avaya Aura Media Server, and Avaya Aura Contact Center modules, Avaya SBC, and Avaya Aura Agent Desktop application, as detailed below.

# 3 Topology

The network topology is illustrated below and is representative of Twilio with Avaya Aura Contact Center

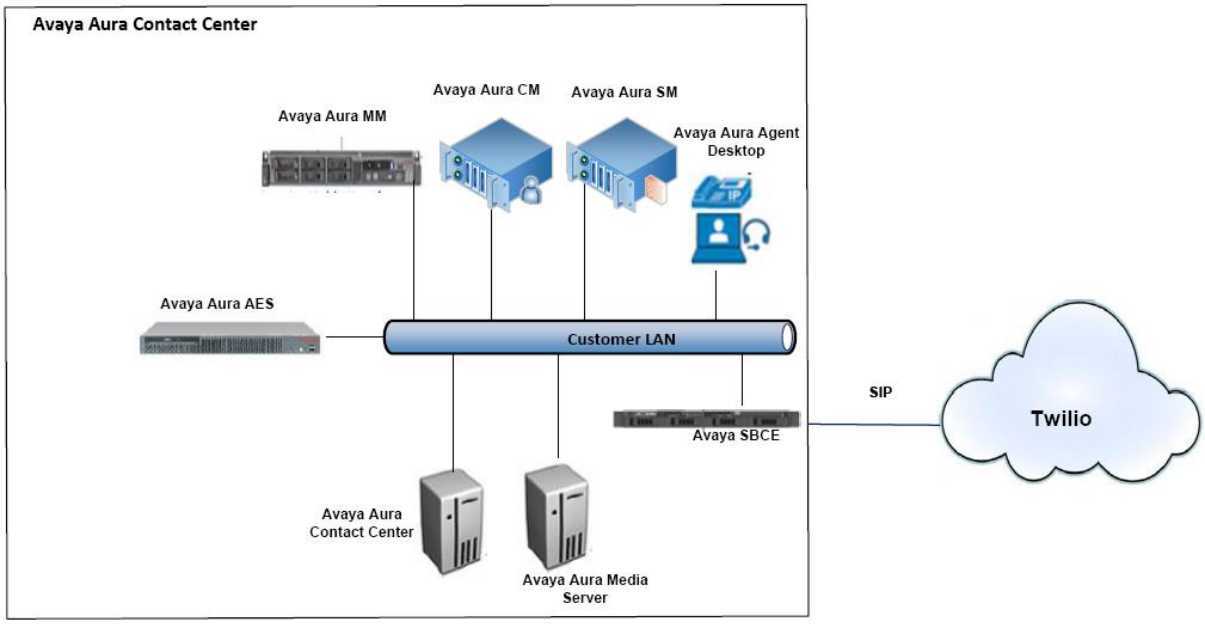



*Figure 2 Network Topology*

## 3.1 Hardware Components

- Avaya one-X Communicator
- Avaya Aura Agent Desktop

The following components are hosted in VMWare ESXi version 6.5

- Avaya Aura Contact Center (AACC)
- Avaya Aura Media Server (AAMS)

The following components are hosted in UCS-C240 VMWare server ESXi 6.0

- Avaya Aura Communication Manager (CM)
- Avaya Aura Session Manager (SM)
- Avaya Session Border Controller for Enterprise (SBCE)

## 3.2 Software Requirements

- Avaya One-X Communicator 6.2.4.06-FP4
- Avaya Aura Agent Desktop 7.1
- Avaya Aura Contact Center 7.1.2. Avaya Aura Contact Center is upgraded with Feature Pack 2 Service Pack 1 GA patches
- Avaya SBCE 10.1.0.0-32-21432

Avaya Aura

- Communication Manager 10.1
- Session Manager 10.1
- System Manager 10.1

# 4 Pre-requisites

## 4.1 Twilio

- Twilio account
- Twilio phone number
- Twilio Studio to build IVR flow

## 4.2 Avaya Aura Contact Center

- Avaya SBCE to accept calls from Twilio and to route calls to Avaya SM
- Avaya Aura SM setup with all the mandatory elements to handle the inbound Twilio calls to the Avaya Aura Contact Center
- Avaya Aura Orchestration Designer to create an application script containing a call flow for routing calls to a queue and agents.

# 5 Avaya Aura Contact Center Call Flow

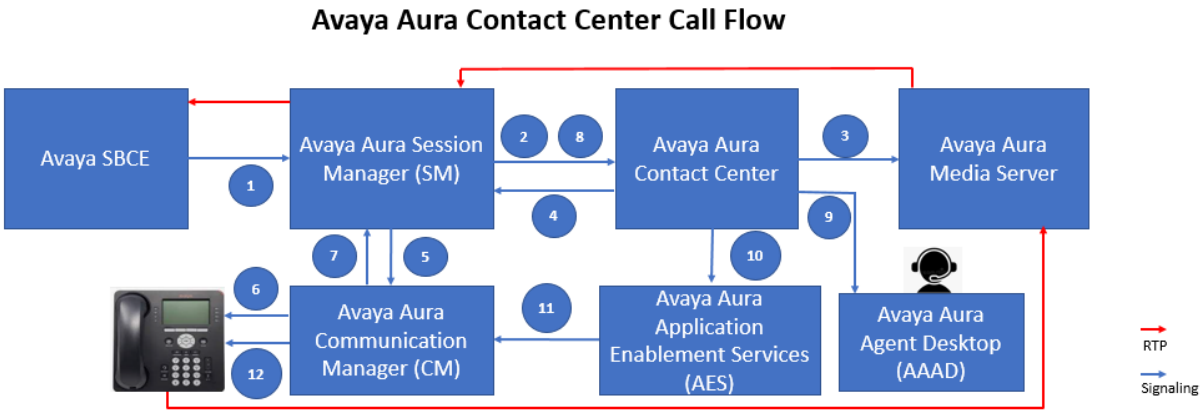



*Figure 3 Call Flow*

1. Avaya SBCE constructs a SIP User-to-User header from the X-parentCall header value received from Twilio and sends the header to Avaya SM
2. Avaya Aura SM constructs a SIP User-Agent header using the adaptation rules from the SIP User-to-User header received and sends it to Avaya Aura Contact Center (AACC)
3. AACC receives the call with the SIP User-Agent header. The call is sent to a skillset group defined in an AACC application script and then to Avaya Aura Media Server. Music is played to the PSTN user when the user waits in the queue.
4. AACC sends the call to Avaya Aura SM to route the call to an agent assigned to the skillset group.
5. Avaya Aura SM sends the call to Avaya Aura CM.
6. Avaya Aura CM sends the call to Avaya agent desk phone.
7. Avaya Aura CM sends the agent ringing information to Avaya Aura SM.
8. Avaya Aura SM forwards that to AACC.
9. AACC delivers the call to Avaya Aura Agent Desktop (AAAD) and the agent answers the call.
10. AACC sends offhook information to Avaya Aura Application Enablement Services (AES).
11. Avaya Aura Application Enablement Services (AES) notifies Avaya Aura CM to offhook Avaya agent desk phone.
12. Avaya agent desk phone goes offhook.

# 6 Configuration

As described above, this configuration assumes the customer already has a Twilio account set up, and phone number purchased. For more information on how to do that, see the Twilio Programmable Voice Quickstart Guide. The link for Twilio's programmable Voice Quickstart Guide is https://www.twilio.com/docs/voice.

## 6.1 Twilio Configuration

### 6.1.1 SIP domains

- Navigate to **Voice > Manage > SIP domains**
- Under **Configure**,
  - *FRIENDLY NAME:* **AvayaPV**
  - *SIP URI:* **avaya**
- Under **Voice Authentication**,
  - *IP ACCESS CONTROL LISTS:* Select the IP access control lists to authenticate inbound calls to Twilio (Refer Section 6.1.2)
  - *CREDENTIAL LISTS:* Select the appropriate Credential list for authentication (Refer Section 6.1.3)

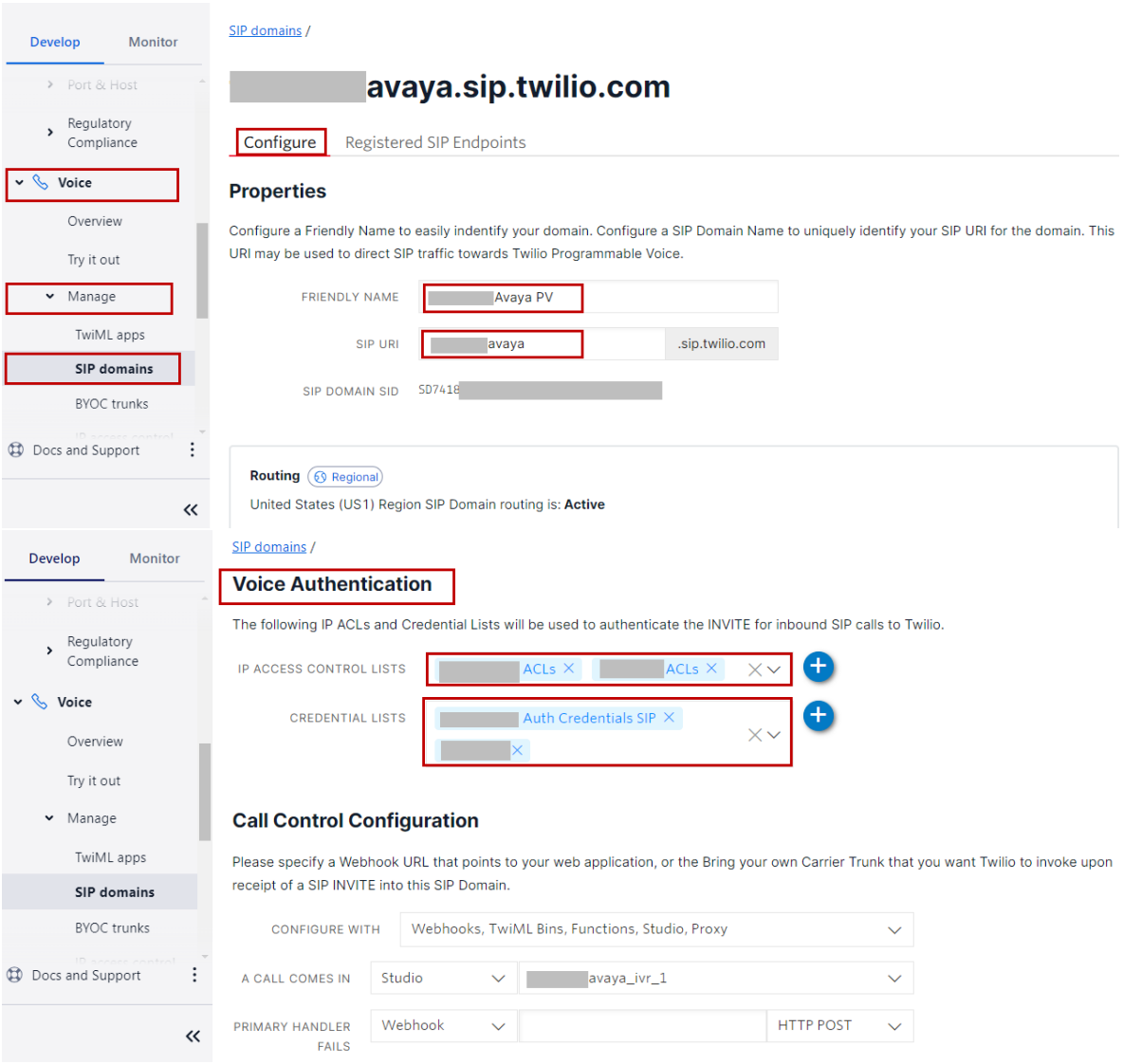



*Figure 4 SIP domains*

- Under **SIP Registration**,
  - *Allow SIP Endpoints to register:* **ENABLED**
- Under **SIP Registration Authentication**,
  - *CREDENTIAL LISTS:* Select the appropriate Credential list for authentication of SIP Endpoint (Refer Section 6.1.3)
- Click **Save**

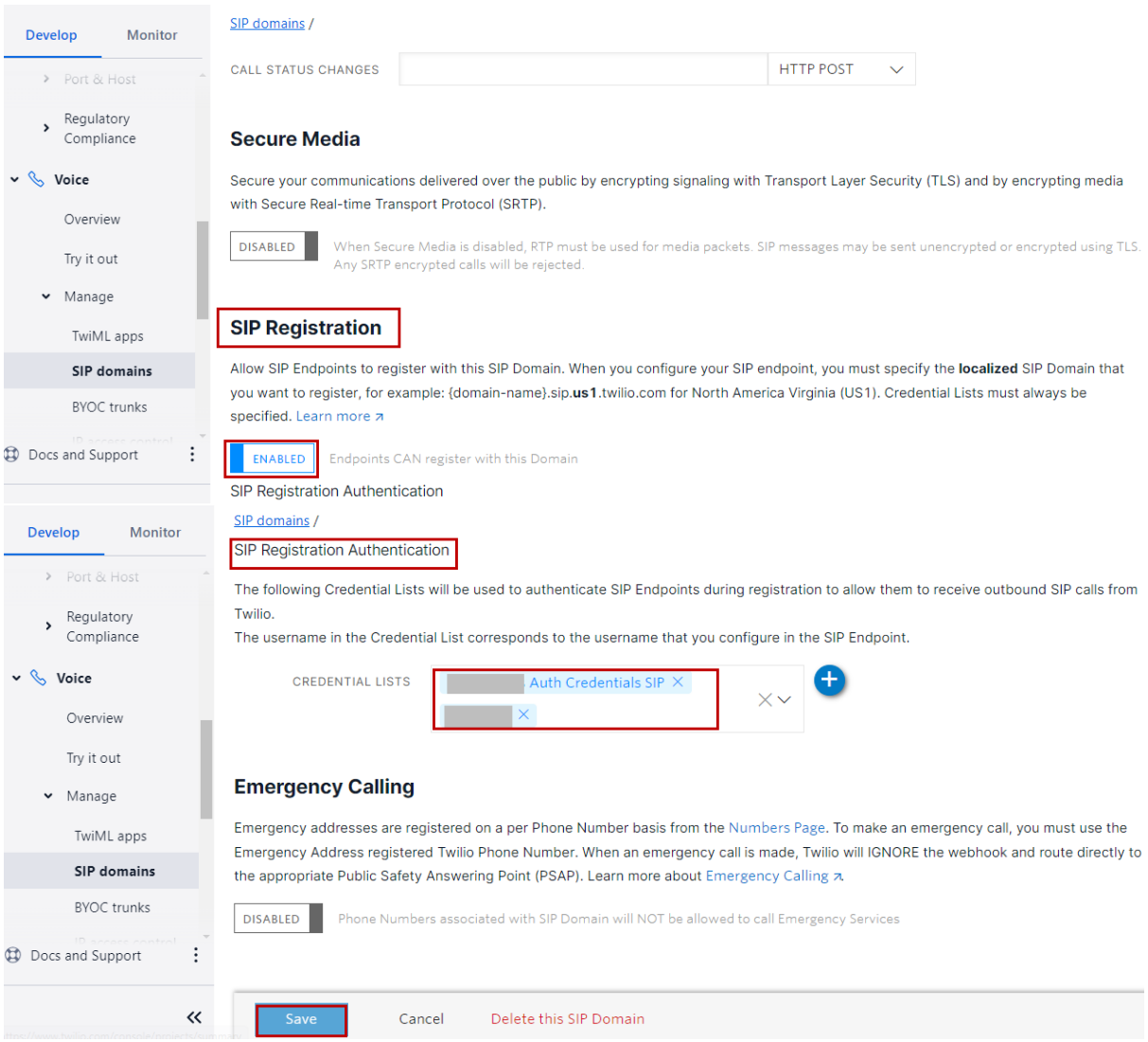



*Figure 5 SIP domains Continuation*

### 6.1.2 IP access control lists

- Navigate to **Voice > Manage > IP access control lists**
- Click **Create new IP Address Range**
- *Friendly Name*: **ACLs**
- *IP Address Range*: Enter the Avaya SBCE WAN interface IP Range
- *Friendly Name*: **IP**

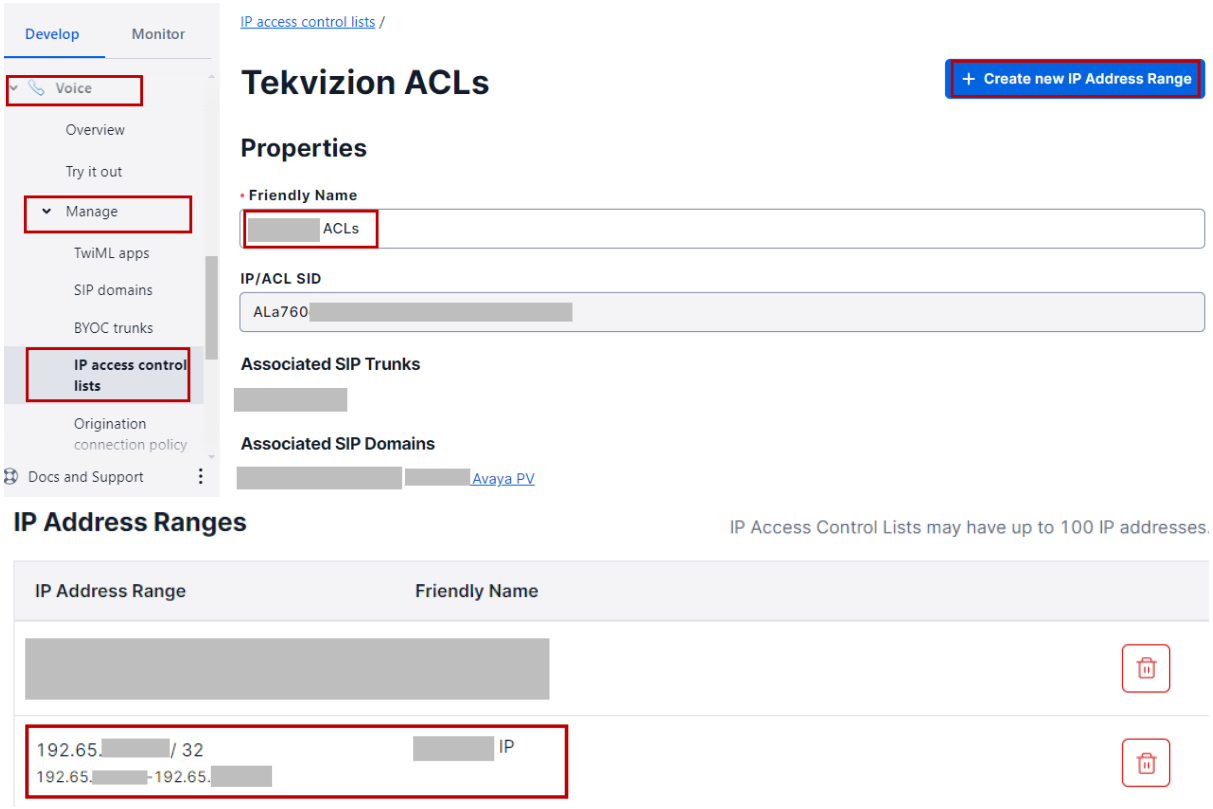



*Figure 6 IP access control lists*

### 6.1.3 Credential lists

- Navigate to **Voice > Manage > Credential lists**
- *Username*: Enter the **Username**
- *Password:* Enter the **Password**

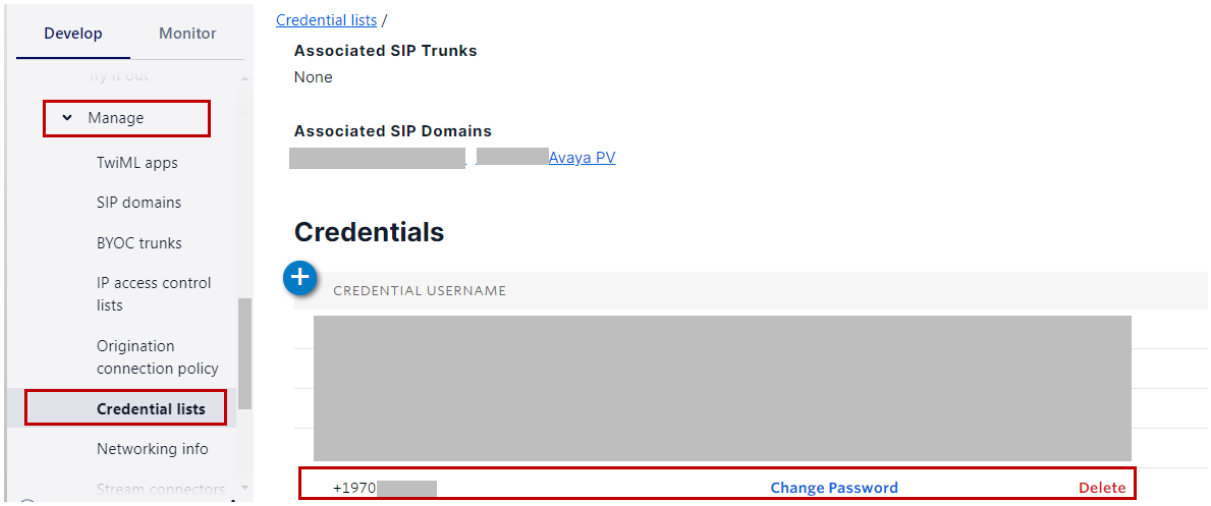



*Figure 7 Credential lists*

### 6.1.4 Twilio Studio IVR Flow

Twilio IVR Studio flow is created to play IVR, collect a digit from the PSTN user, store it in a repository, transfer the call to Avaya Aura agent and send the digit or queue information (queue number 1 for Sales and 2 for Support) to the Avaya Aura agent as a SIP intrinsic

- Navigate to **Studio > Flows**
- Click **Create new Flow**
- *FlOW NAME*: **Dual Language IVR Test**
- Click **Next**
- *Choose a template to get going*: **Enterprise IVR/Phone Tree**
- Click **Next**

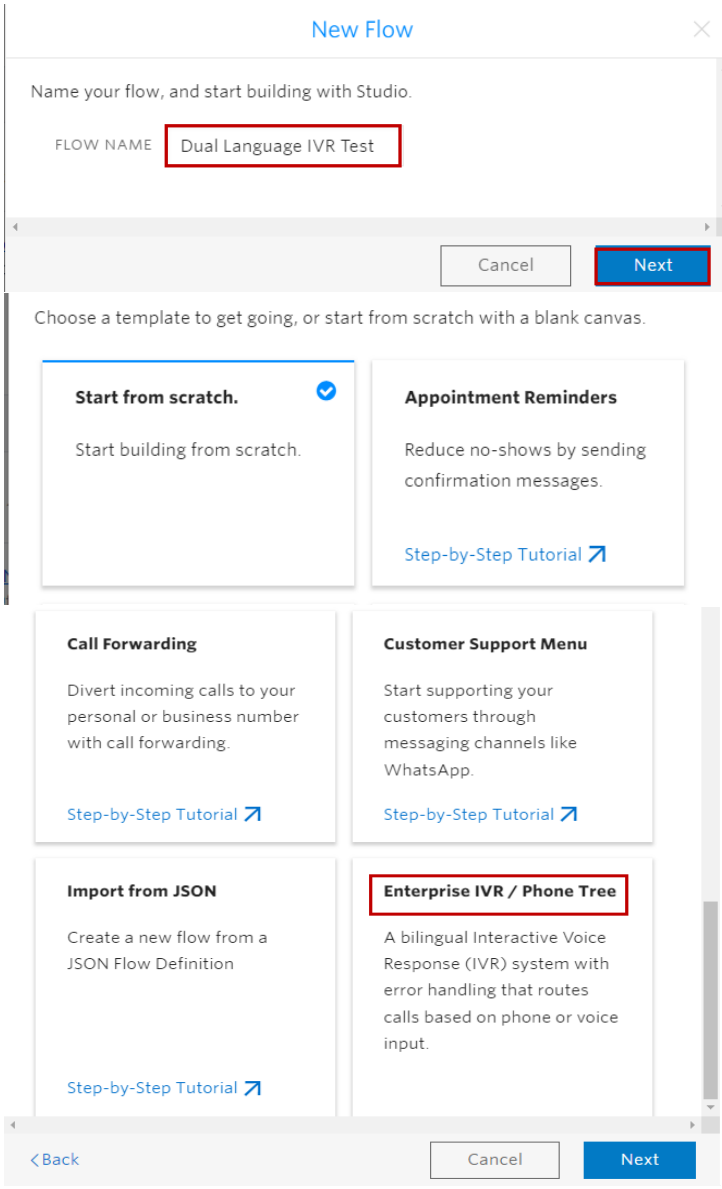



*Figure 8 Twilio IVR Studio Flow – Create New Flow*

This creates an IVR template as shown below.

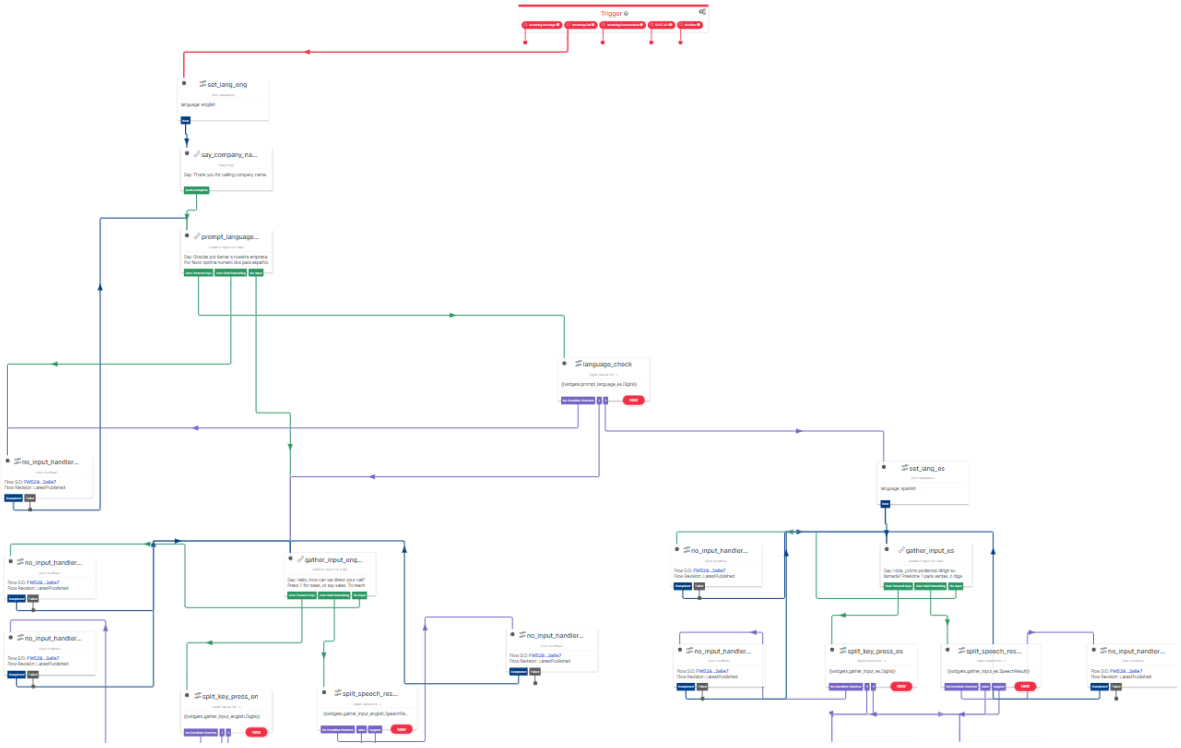


*Figure 9 Twilio IVR Studio Flow – Template*

End to End Twilio IVR Studio flow built from the template for this interoperability testing is shown below.

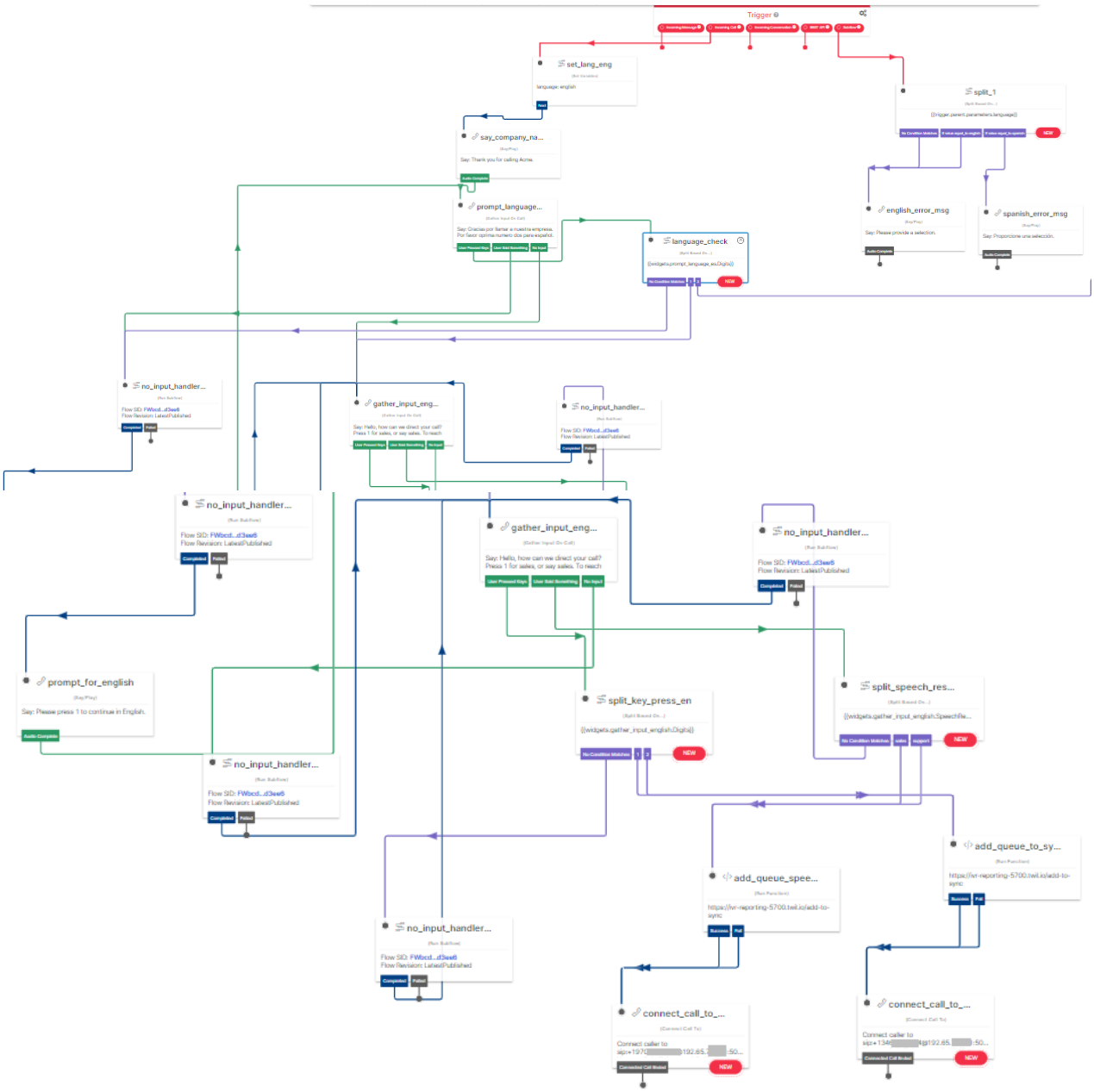



*Figure 10 Twilio IVR Studio Flow – End to End Flow*

The End to End IVR Flow (Figure 10) interacts with the PSTN user, through the following questions

E.g. Flow number IVR navigation 1

1. PSTN user calls Twilio IVR DID
2. PSTN user hears “Thank you calling Acme” followed by the announcement in Spanish “Gracias por llamar”
3. PSTN does not provide any input
4. PSTN user hears “Hello, how can we direct your call? Press 1 for sales, or say sales. To reach support, press 2 or say support.”
5. PSTN user presses 1
6. PSTN user is redirected to Avaya Aura agent for further interaction

E.g. Flow number IVR navigation 2

1. PSTN user calls Twilio IVR DID
2. PSTN user hears “Thank you calling Acme” followed by the announcement in Spanish “Gracias por llamar”
3. PSTN does not provide any input
4. PSTN user hears “Hello, how can we direct your call? Press 1 for sales, or say sales. To reach support, press 2 or say support.”
5. PSTN user presses 2
6. PSTN user is redirected to Avaya Aura agent for further interaction.

**Detailed view of Twilio IVR Studio flow**

**Widget: set_lang_eng**

- Create a widget **Set Variables** named **set_lang_eng** and this is the entry point when the incoming call is arrived to the Twilio IVR DID
- **WIDGET LIBRARY** can be used to create different kinds of widget. e.g. To set variables, the Set Variables Widget can be used
- Drag and Drop a link from the **Incoming Call** to **set_lang_eng** to create a link between them
- Expand the side panel **<<** to configure the variables for a widget
- This widget sets the variable **language** with value **english**
- The configuration needs to be Published using **Publish** once the end to end flow configuration is completed

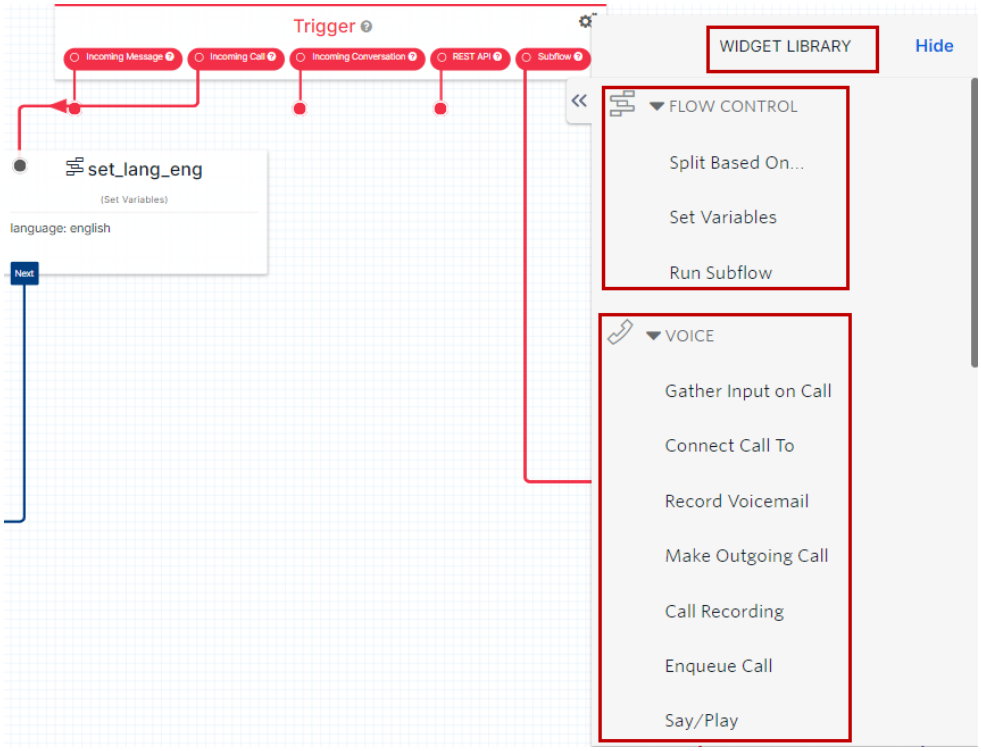



*Figure 11 Twilio IVR Studio Flow – set_lang_eng*

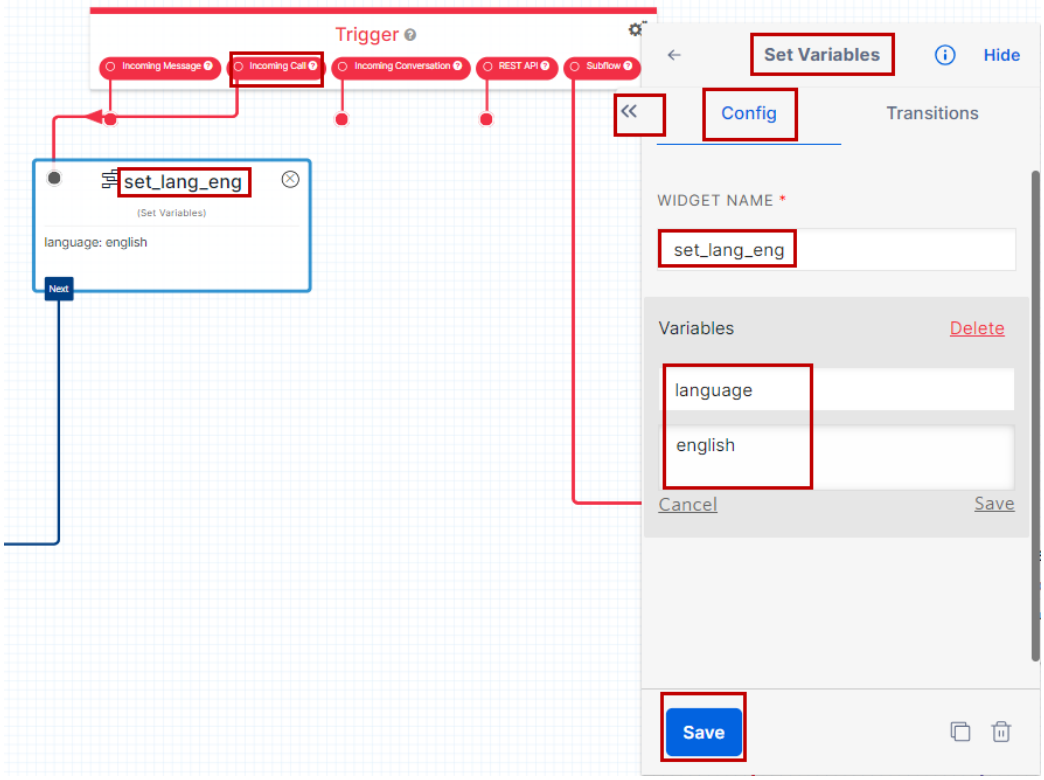



*Figure 12 Twilio IVR Studio Flow – set_lang_eng Continuation*

**Widget: set_company_name**

- **Next** option from **set_lang_eng** is connected to widget **Say/Play** named **say_company_name** and this plays the message “Thank you for calling Acme”
- The other values configured for this widget are highlighted below.

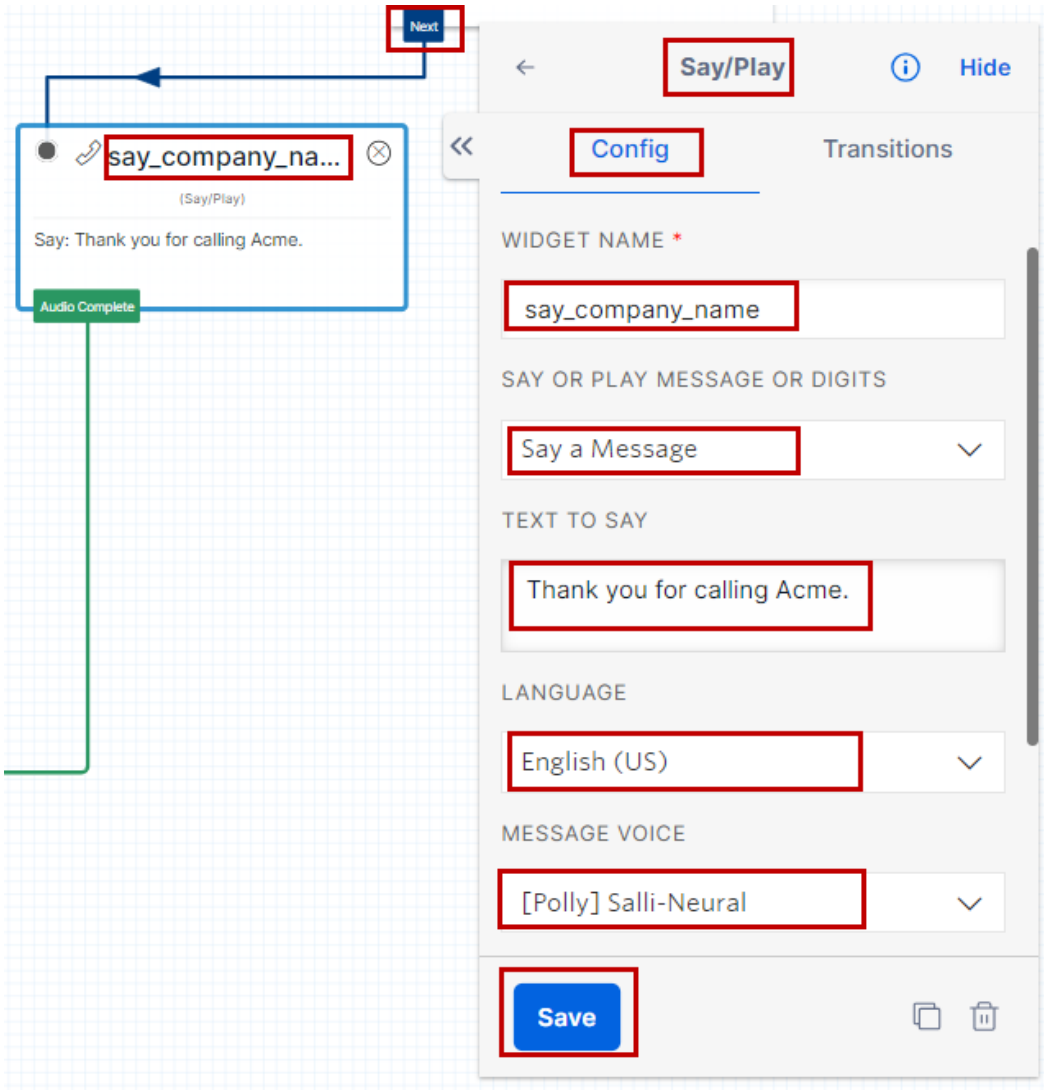



*Figure 13 Twilio IVR Studio Flow – say_company_name*

**Widget: prompt_language_es**

- **Audio Complete** option from **say_company_name** is connected to widget **Gather Input On Call** named **prompt_language_es** and this plays the Spanish message "Gracias por llamar"
- The other values configured for this widget are highlighted below.

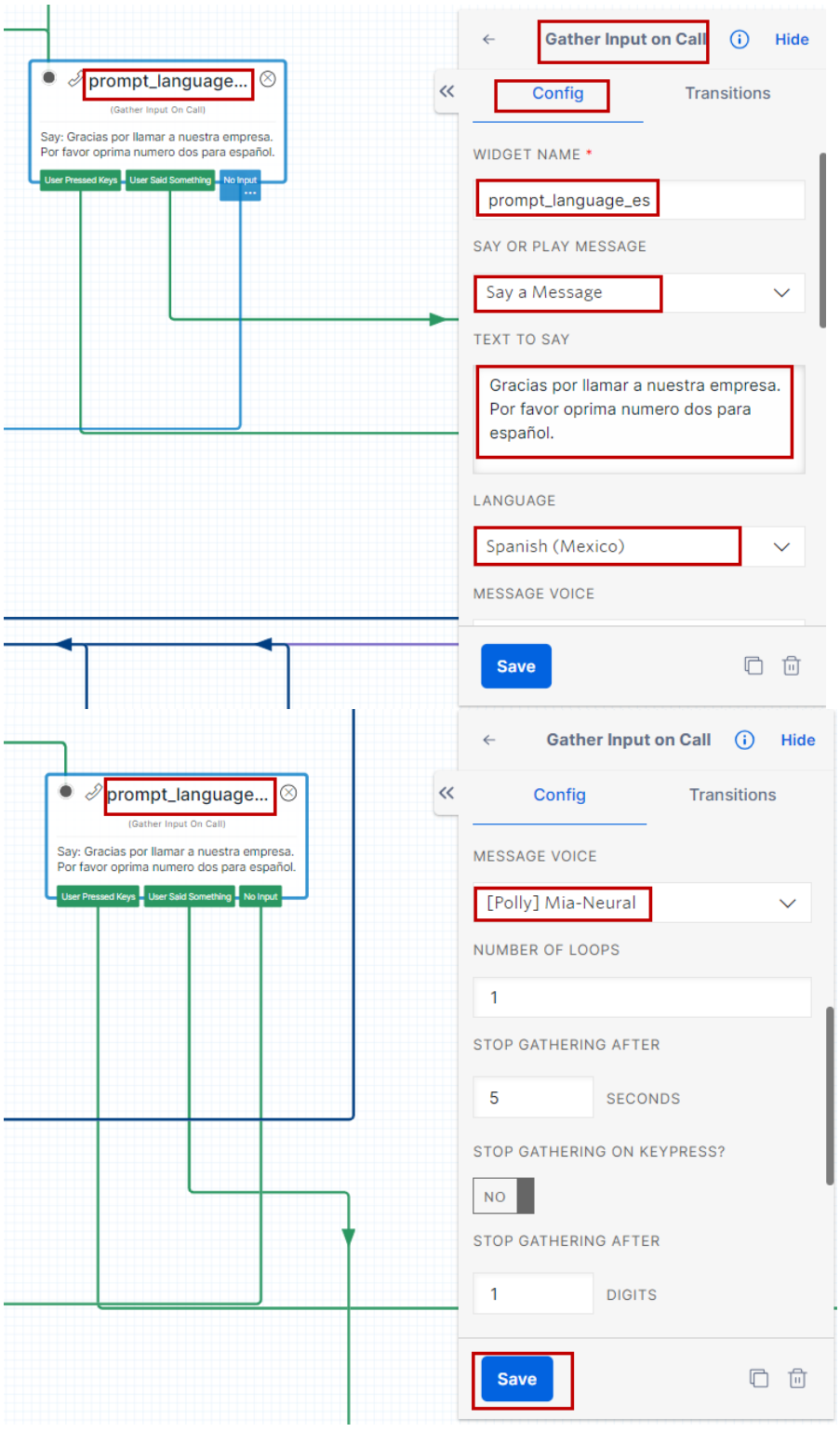



*Figure 14 Twilio IVR Studio Flow – prompt_language_es*

**Widget: language_check**

- **User Pressed Keys** (in the widget **prompt_language_es**) option is connected to widget **Split Based On** named **language_check**. This splits the language English/Spanish based on the digit which is pressed.
- The other values configured for this widget are highlighted below.

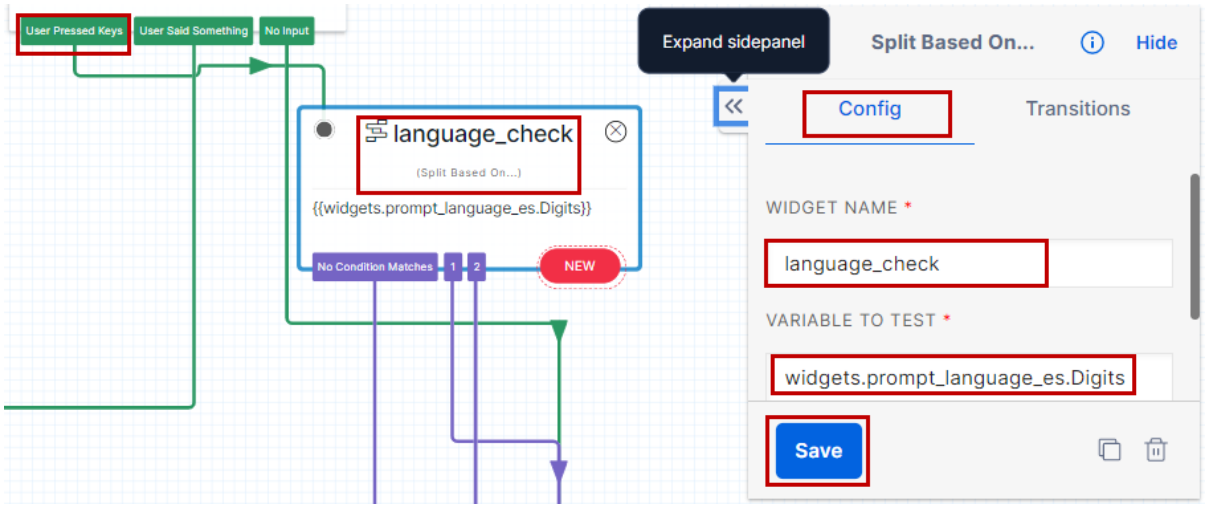



*Figure 15 Twilio IVR Studio Flow – language_check*

**Widget: gather_input_english**

- **Digit 1** option (in the widget **language_check**) is connected to widget **Gather input On Call** named **gather_input_english**. This plays the IVR “Hello, how can we direct your call”.
- The other values configured for this widget are highlighted below.

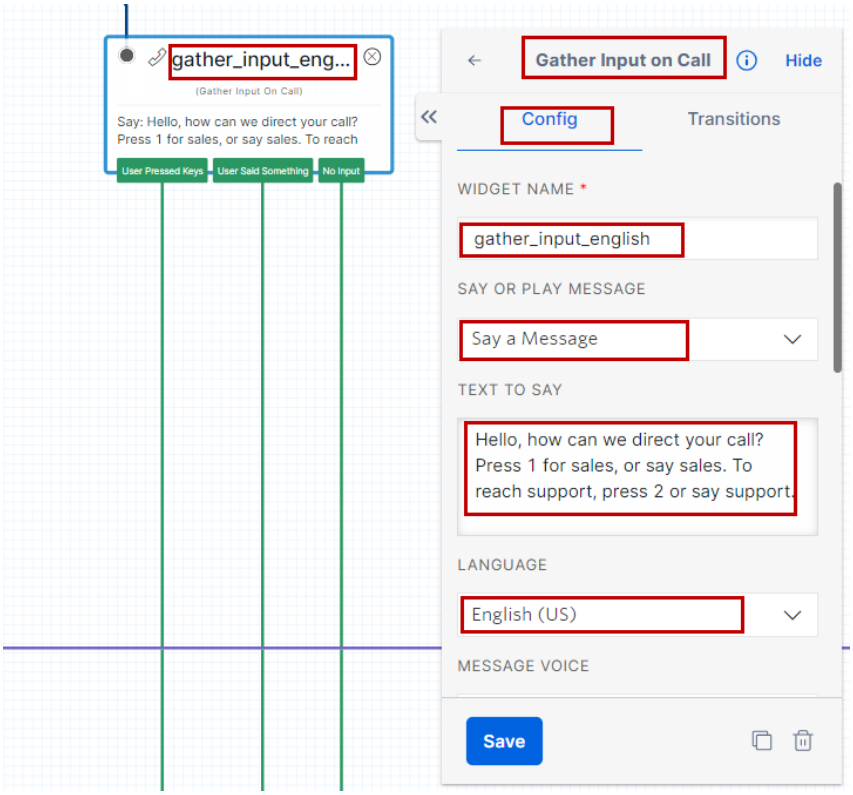



*Figure 16 Twilio IVR Studio Flow – language_check*

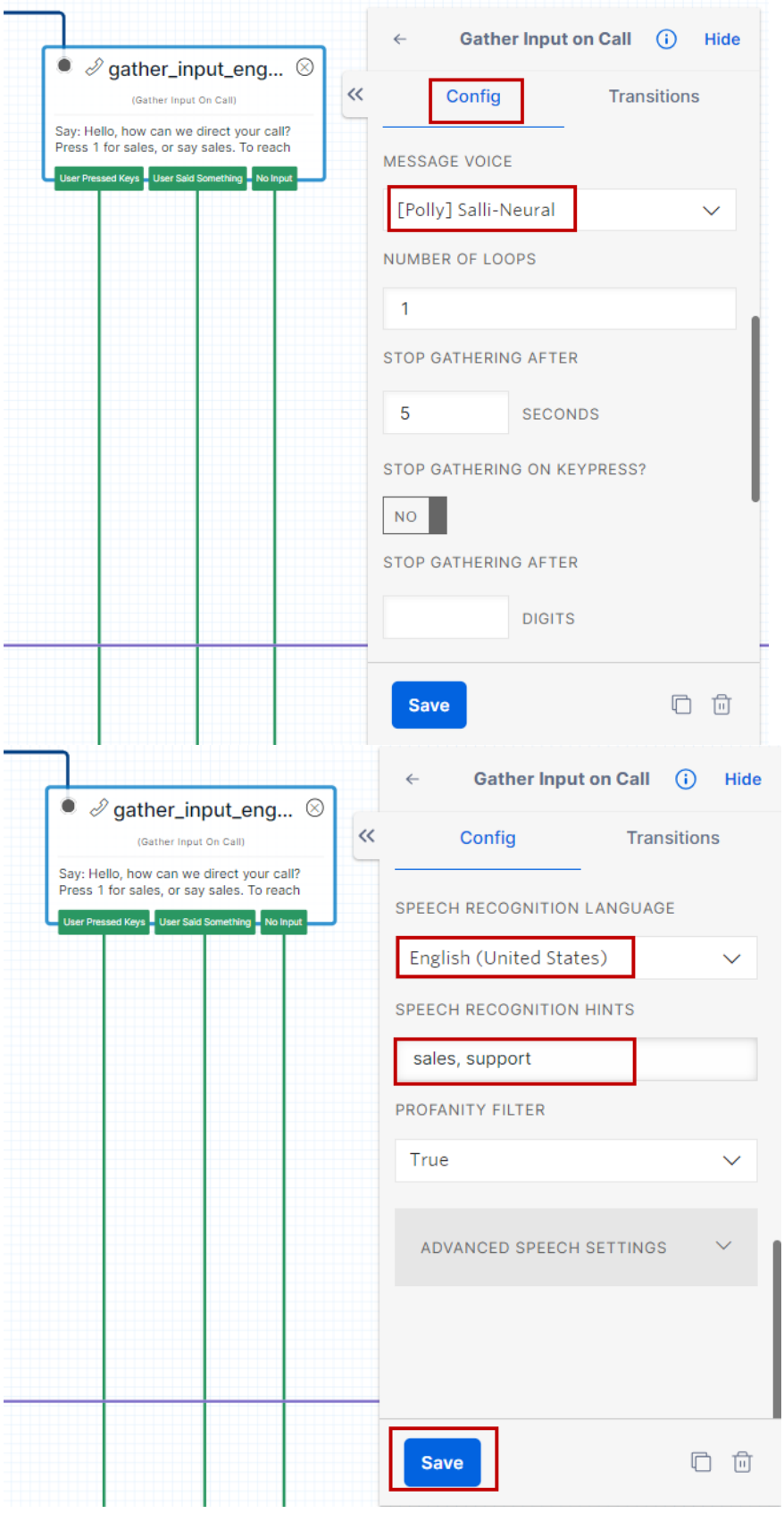



*Figure 17 Twilio IVR Studio Flow – language_check Continuation*

**Widget: no_input_handler**

- **No Condition Matches** option (in the widget **language_check**) is connected to widget **Run Subflow** named **no_input_handler_1**.
- The parameter **flow.variables.language (i.e. english)** is invoked by the Widget named **Split_1 (Figure 19)**
- The other values configured for this widget are highlighted below.

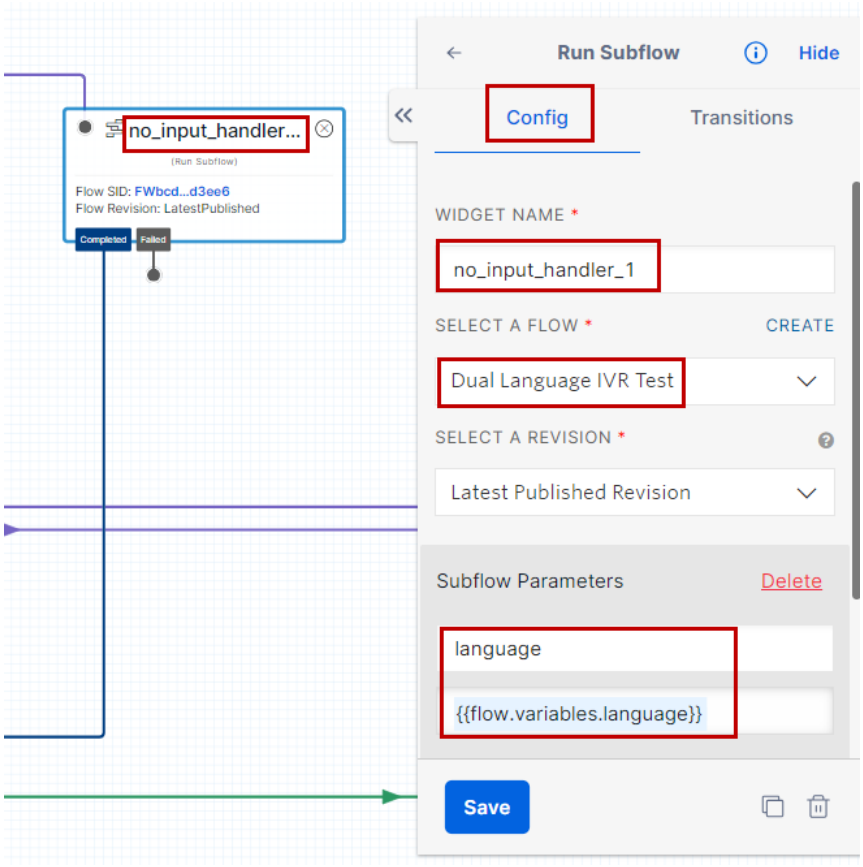



*Figure 18 Twilio IVR Studio Flow – no_input_handler_1*

**Widget: split_1**

- Widget **no_input_handler_1 (Figure 18)** invokes the Subflow Widget **Split Based On**…named **split_1**
- **flow.variables.language (i.e. english)** is referenced at runtime inside the Subflow via the trigger object **trigger.parent.parameters.language**

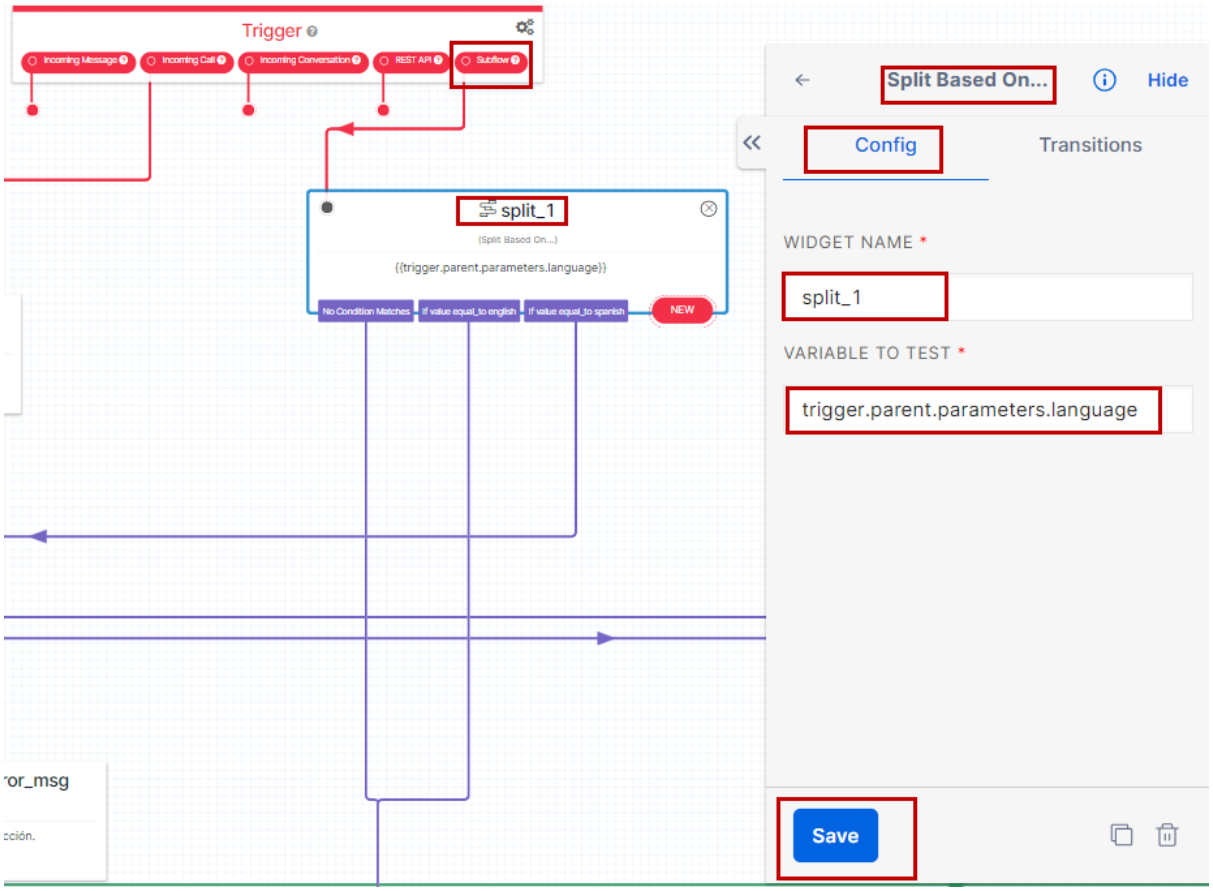



*Figure 19 Twilio IVR Studio Flow – split_1*

**Widget: english_error_msg**

- Widget **split_1 (No Condition Matches, If value equal_to_english)** invokes **Say/Play** Widget named **english_error_msg**
- This widget plays **"Please provide a selection".**
- The other configured values are highlighted below.

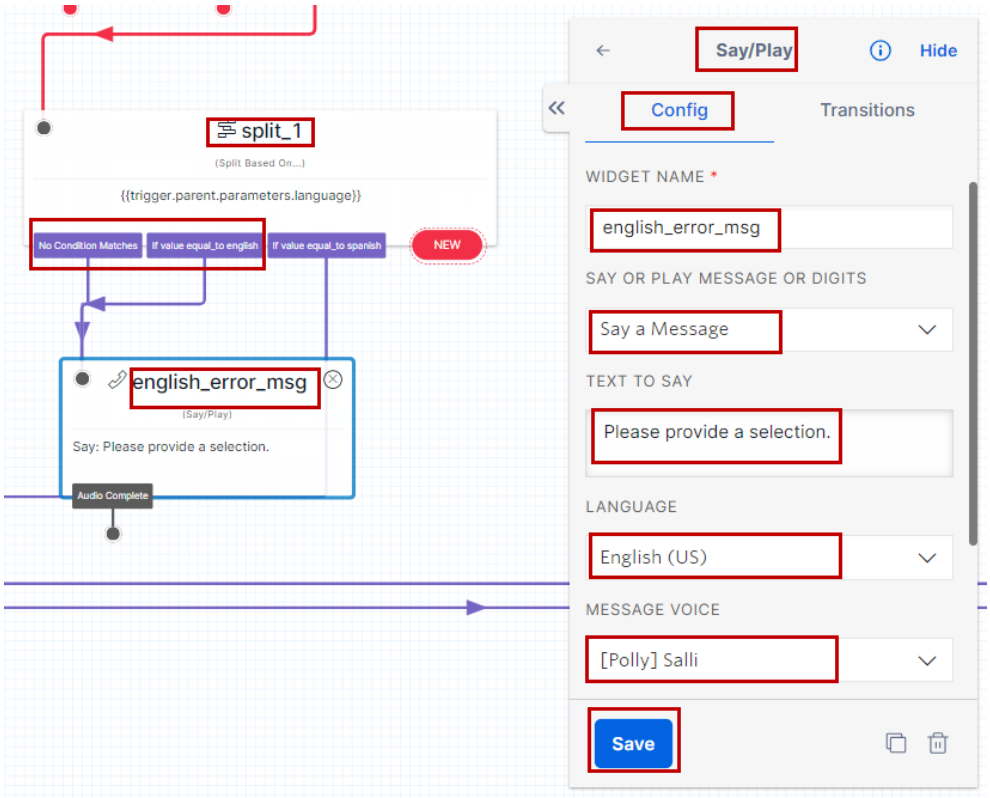



*Figure 20 Twilio IVR Studio Flow – english_error_msg*

**Widget: prompt_for_english**

- Run Subflow Widget named **no_input_handler_1** invokes Say/Play Widget **prompt_for_english**
- This widget plays **"Please press 1 to continue in English".**
- The other configured values are highlighted below.

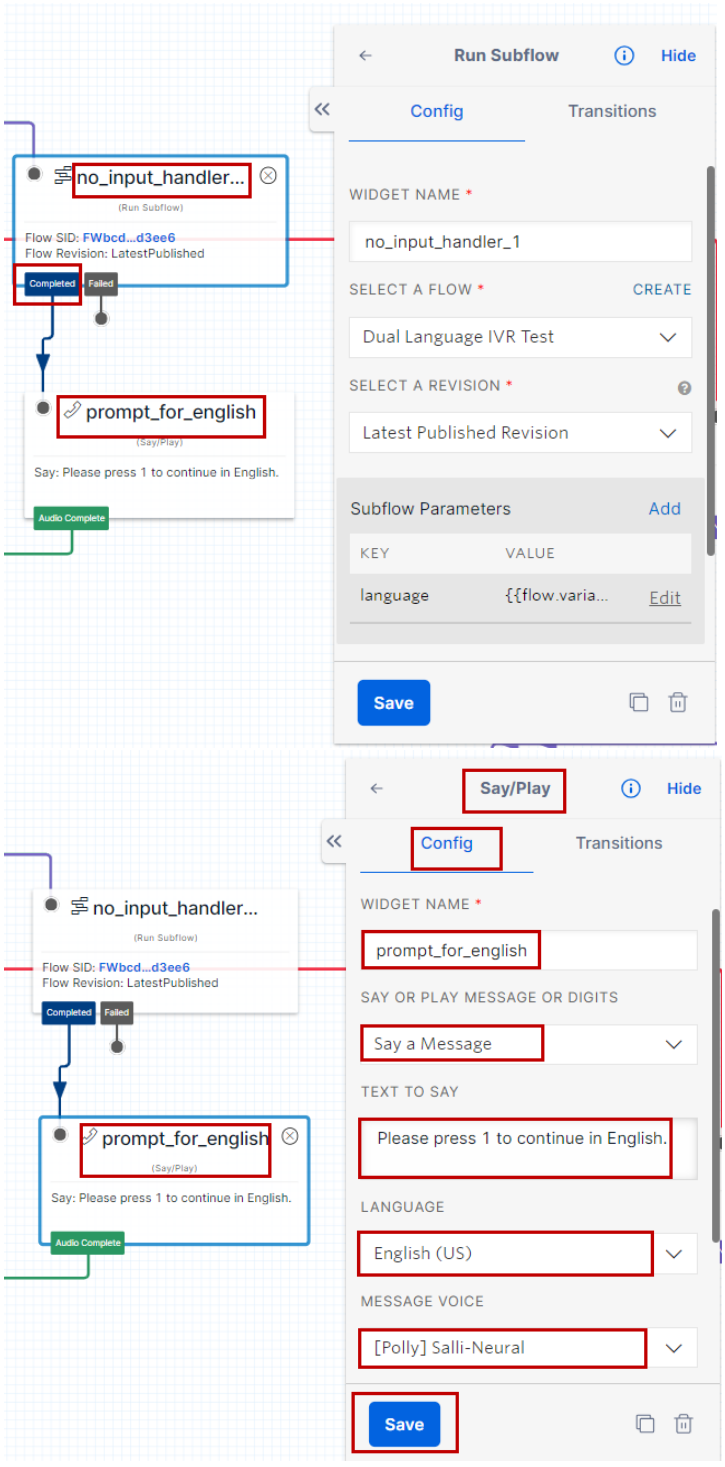



*Figure 21 Twilio IVR Studio Flow – prompt_for_english*

**Widget: prompt_language_es**

- Say/Play Widget named **prompt_for_english** invokes Gather Input on Call Widget named **prompt_language**
- The other configured values are highlighted below.

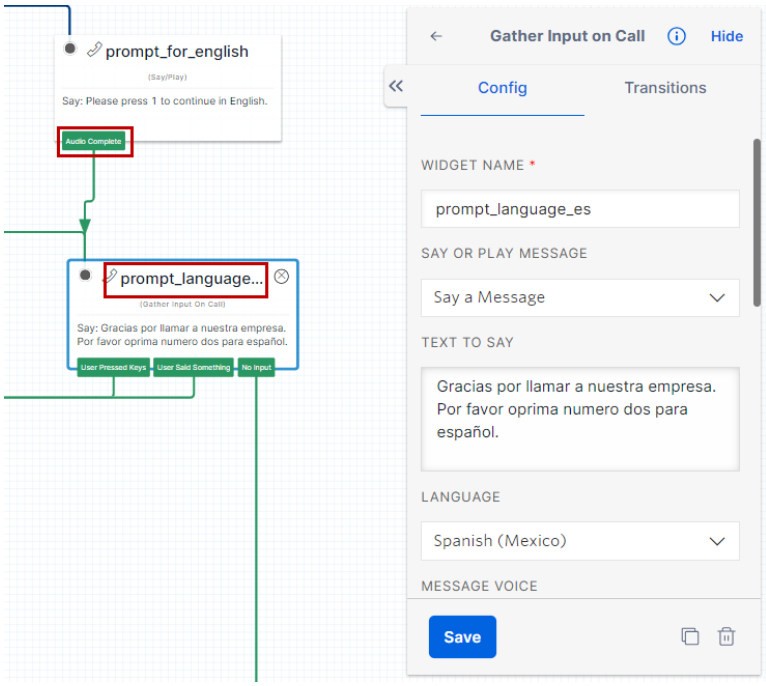



*Figure 22 Twilio IVR Studio Flow – prompt_language*

**Widget: split_key_press_en**

- **User Pressed Keys** option from the Widget **gather_input_english** is linked to Widget Split Based On… named **split_key_press_en**
- This Widget gathers input digits 1 or 2.
- The other configured values are highlighted below.

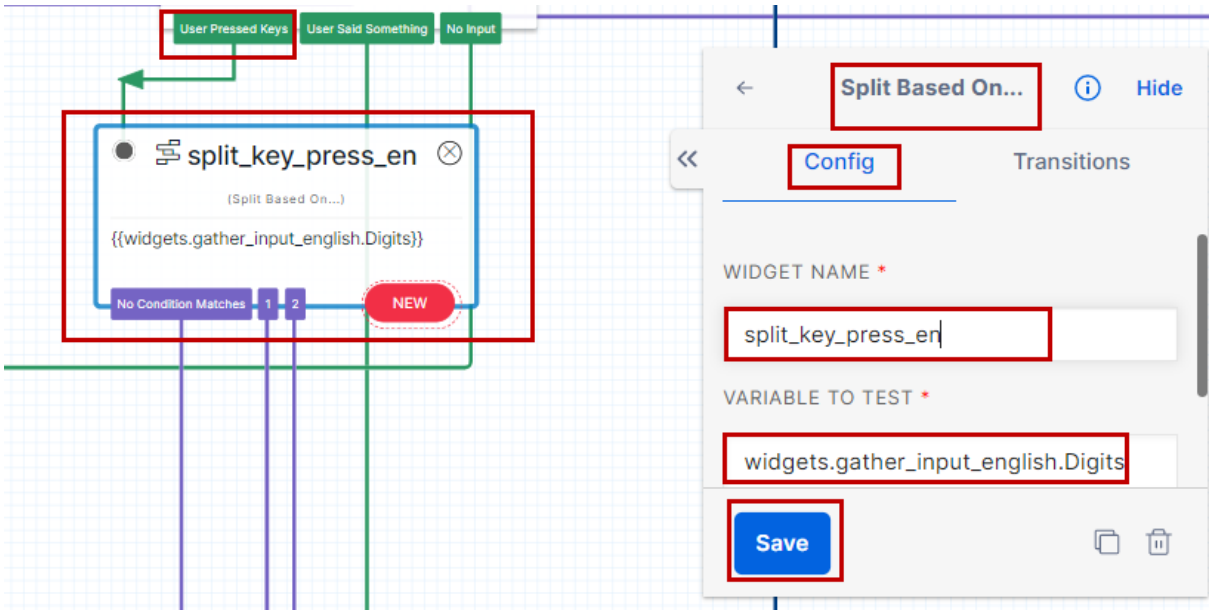



*Figure 23 Twilio IVR Studio Flow – split_key_press_en*

**Widget: add_queue_to_sync**

- The digits which are pressed (1 or 2) from the Widget **split_key_press_en** is linked to Widget Run Function named **add_queue_to_sync**
- This Widget calls the Function **add-to-sync** to store the user entered digit in the Twilio database.
- *FUNCTION URL*: **https://ivr-reporting-5700.twil.io/add-to-sync**

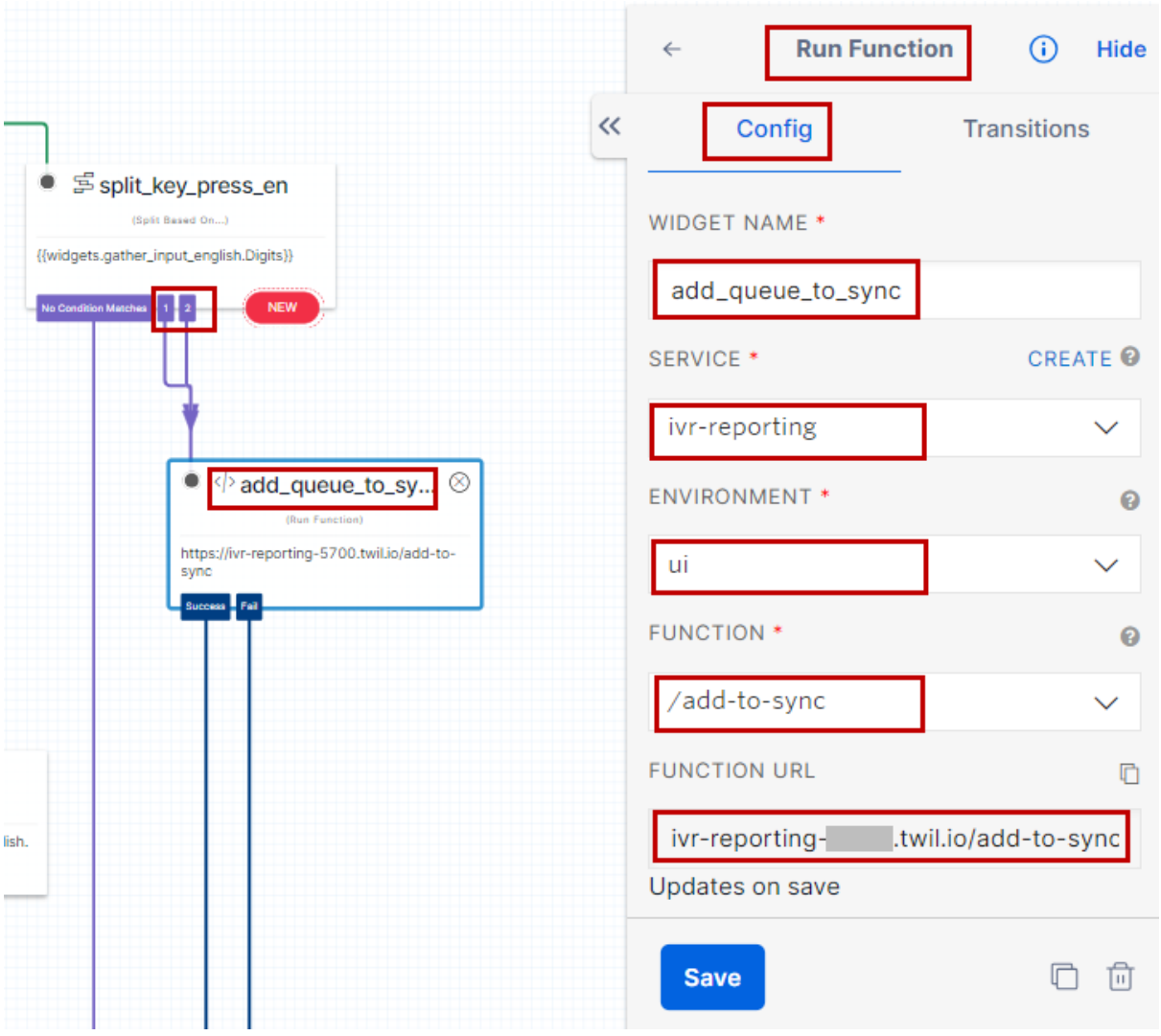



*Figure 24 Twilio IVR Studio Flow – add_queue_to_sync*

- *callSid*: **{{trigger.call.CallSid}}.** This value is set to Twilio CallSid for each call
- *queue*: **{{widgets.gather_input_english.Digits}}.** This value is set to queue number 1 or 2 entered by the PSTN user.
- *callSid* and *queue* are the parameters which are passed to the function **add-to-sync.**

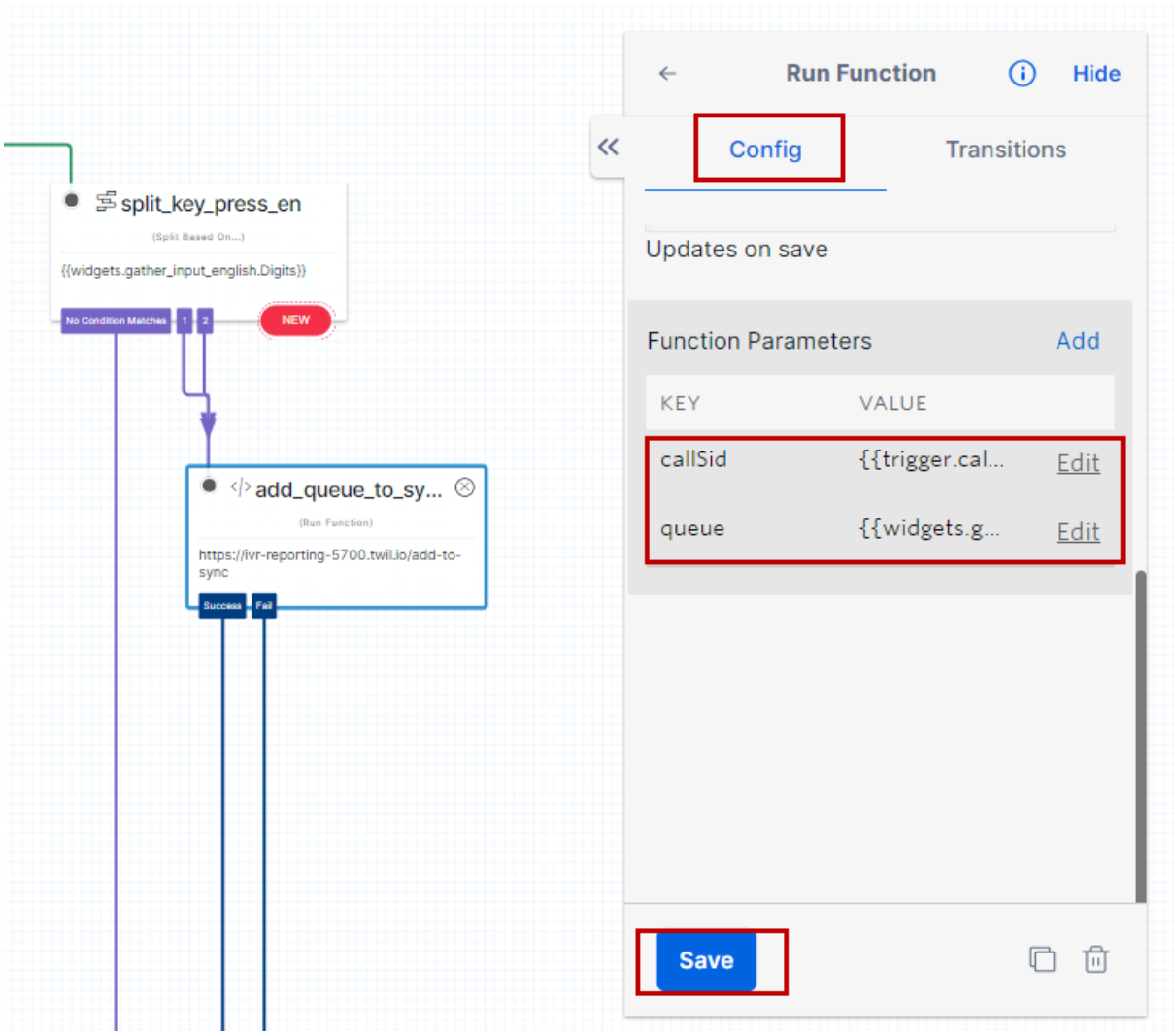



*Figure 25 Twilio IVR Studio Flow – add_queue_to_sync Continuation*

**Widget: connect_call_to_sales**

- **Success or Fail** option from the Widget **add_queue_to_sync** is linked to Widget Connect Call To named **Connect_call_to_sales**
- *SIP ENDPOINT*: **sip:+134XXXXXXXX@192.65.XX.XX:5060?X-parentCall={{trigger.call.CallSid}}**
- This Widget sends the call/INVITE towards Avaya SBCE with the header X-parentCall = Twilio CallSid value. The X-parentCall ID is used by Avaya Aura contact center to retrieve the queue information from Twilio.

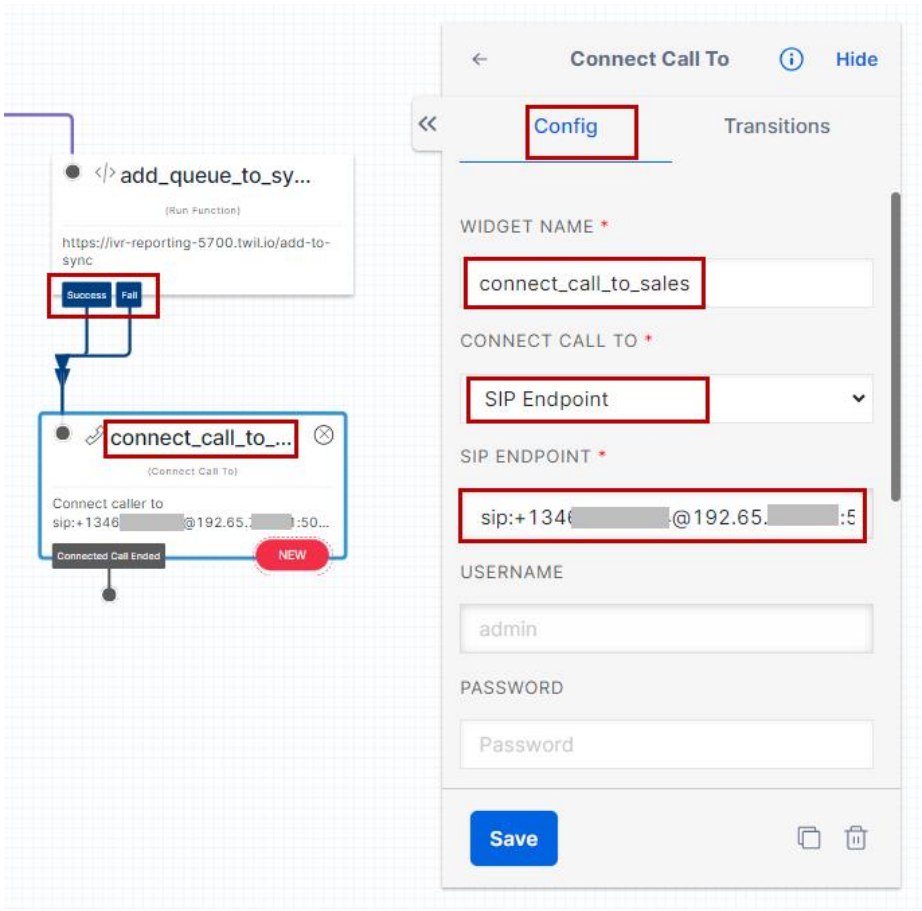



*Figure 26 Twilio IVR Studio Flow – connect_call_to_sales*

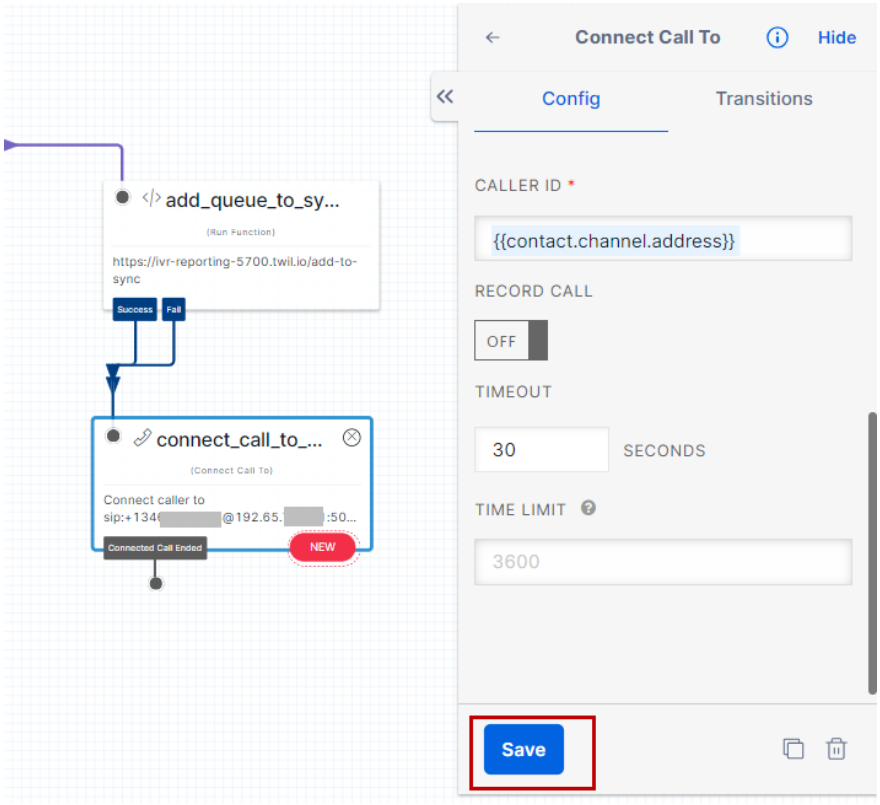



*Figure 27 Twilio IVR Studio Flow – connect_call_to_sales Continuation*

**Widget: no_input_handler_3**

- **No Condition Matches** option from the Widget **split_key_press_en** is linked to Widget Run Subflow named **no_input_handler_3**
- **no_input_handler_3** runs the Subflow (Same as **Figure 19**) and linked to Widget **gather_input_english**

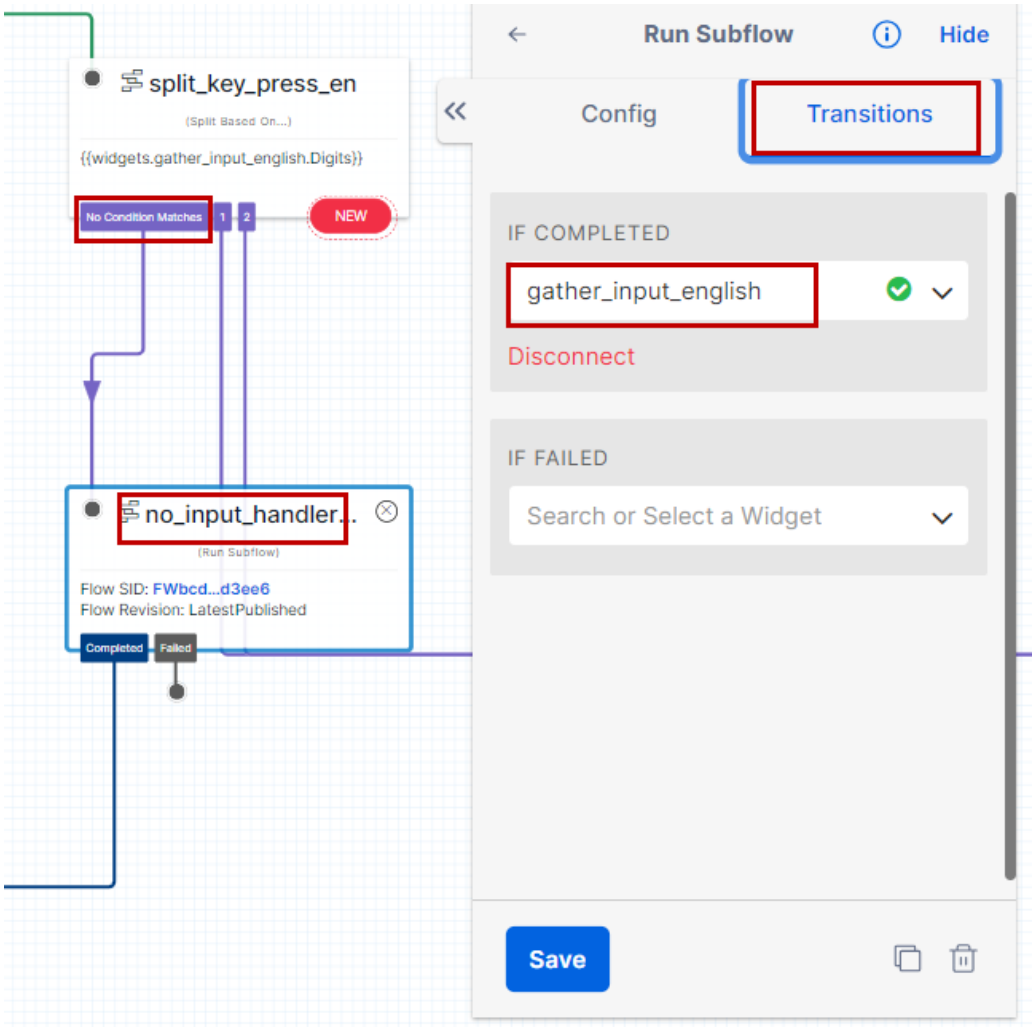



*Figure 28 Twilio IVR Studio Flow – no_input_handler_3*

**Widget: gather_input_english**

- **Completed** option from the Widget **no_input_handler_3** is linked to Widget Gather Input On Call named **gather_input_english**
- **gather_input_english** follows the same flow as **Figure 16**

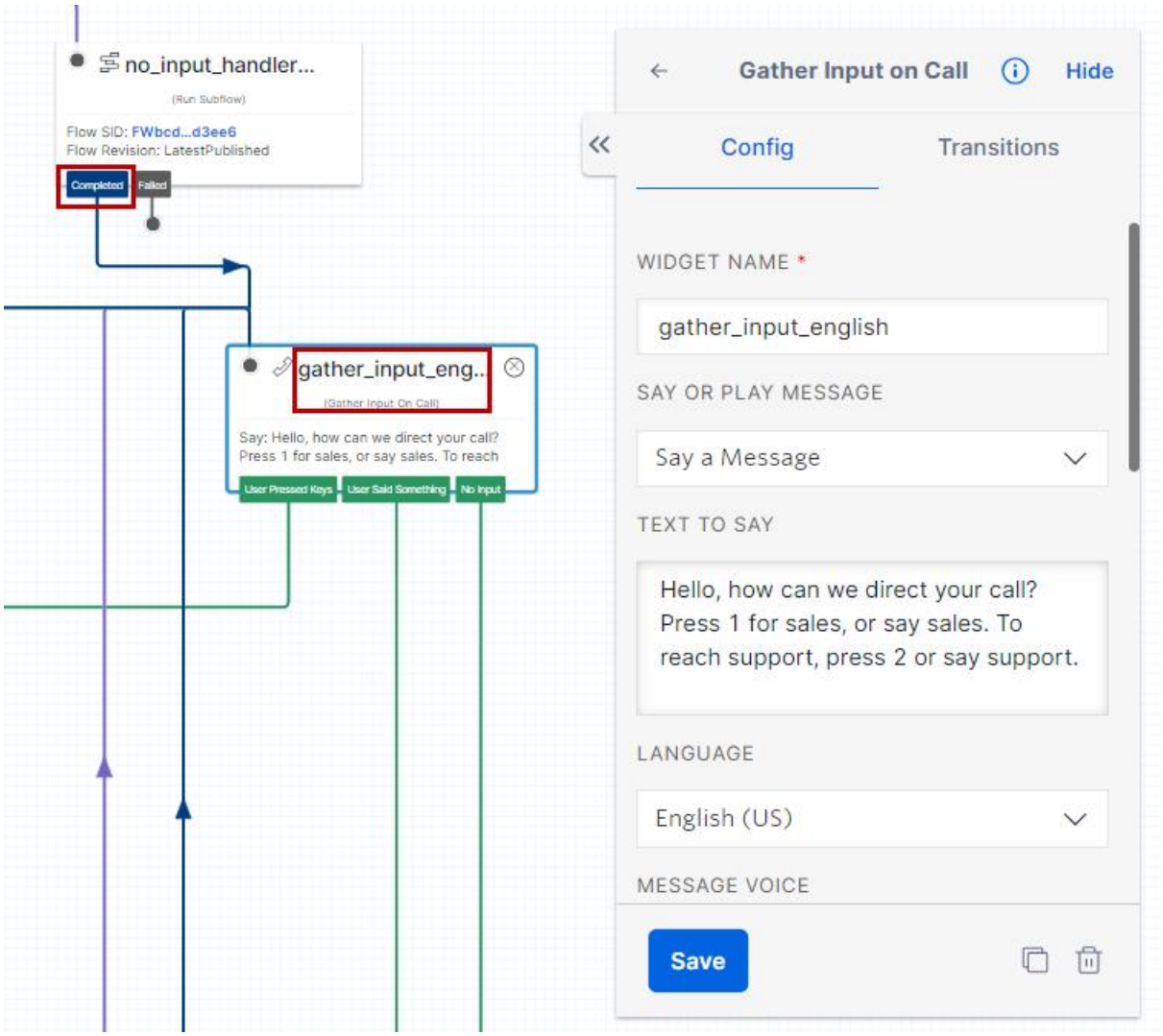



*Figure 29 Twilio IVR Studio Flow – gather_input_english*

**Widget: no_input_handler_1**

- **User Said Something** option from the Widget **prompt_language** is linked to Widget Run Subflow named **no_input_handler_1**
- **no_input_handler_1** follows the same flow as **Figure 18**

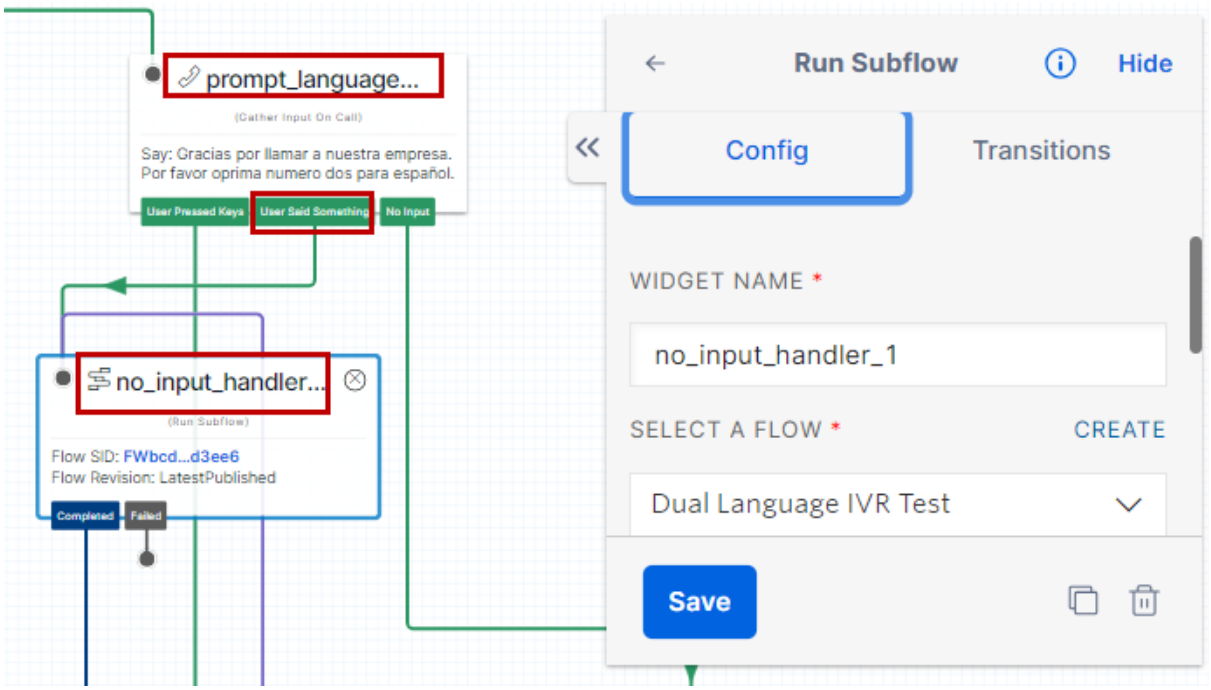



*Figure 30 Twilio IVR Studio Flow – no_input_handler_1*

**Widget: prompt_language**

- **No Input** option from the Widget **prompt_language** is linked to Widget Gather Input On Call named **gather_input_english**

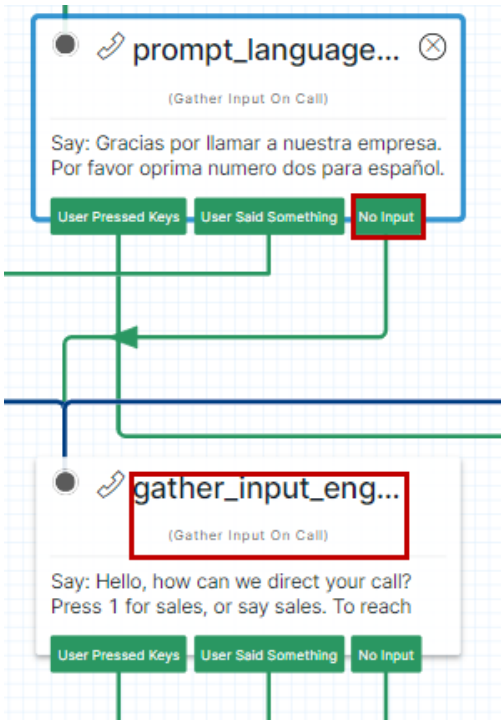



*Figure 31 Twilio IVR Studio Flow – prompt_language*

**Widget: gather_input_english**

- **User Said Something** option from the Widget **gather_input_english** is linked to Widget Split_Based_On… named **split_speech_result_en**
- *VARIABLE TO TEST:* **widgets.gather_input_english.SpeechResult**

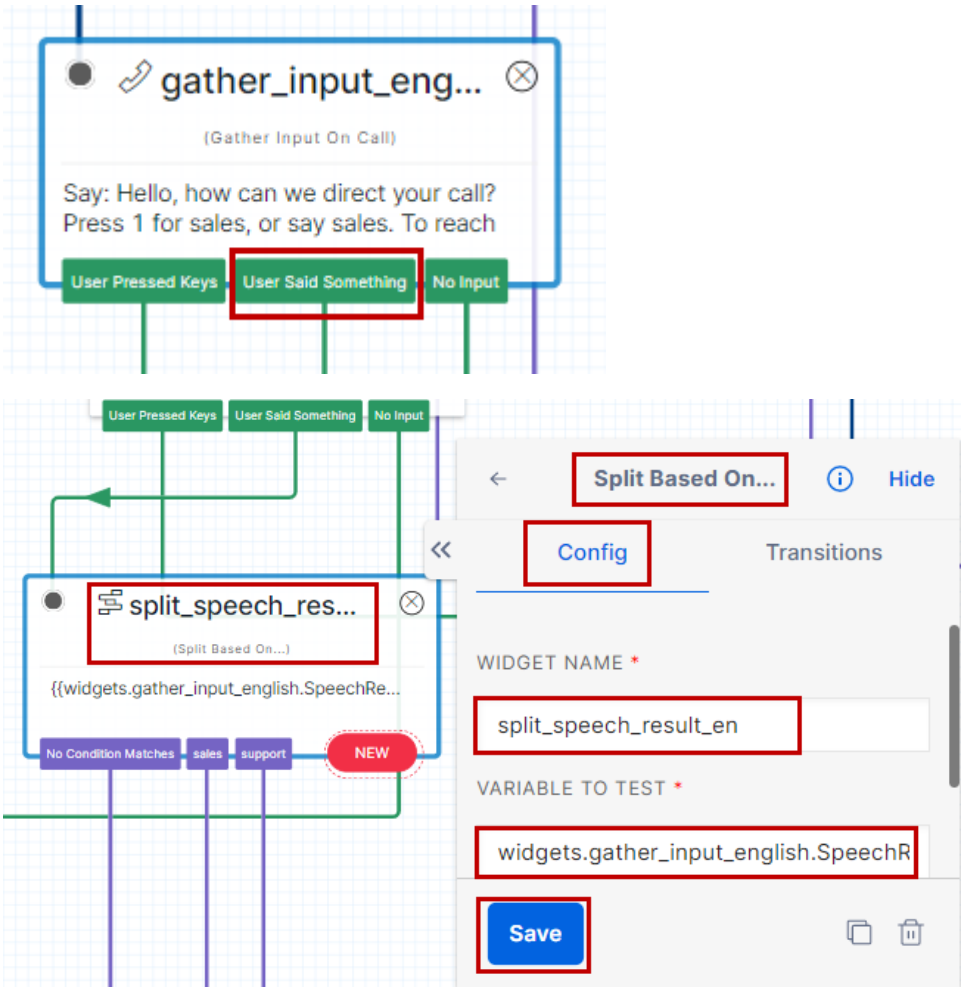



*Figure 32 Twilio IVR Studio Flow – split_speech_result_en*

**Widget: add_queue_speech_to_sync**

- **Sales, Support** options from the Widget **split_speech_result_en** is linked to Widget Run Function named **add_queue_speech_to_sync**
- **add_queue_speech_to_sync** Widget configuration is same as **add_queue_to_sync (Refer Figure 24)**

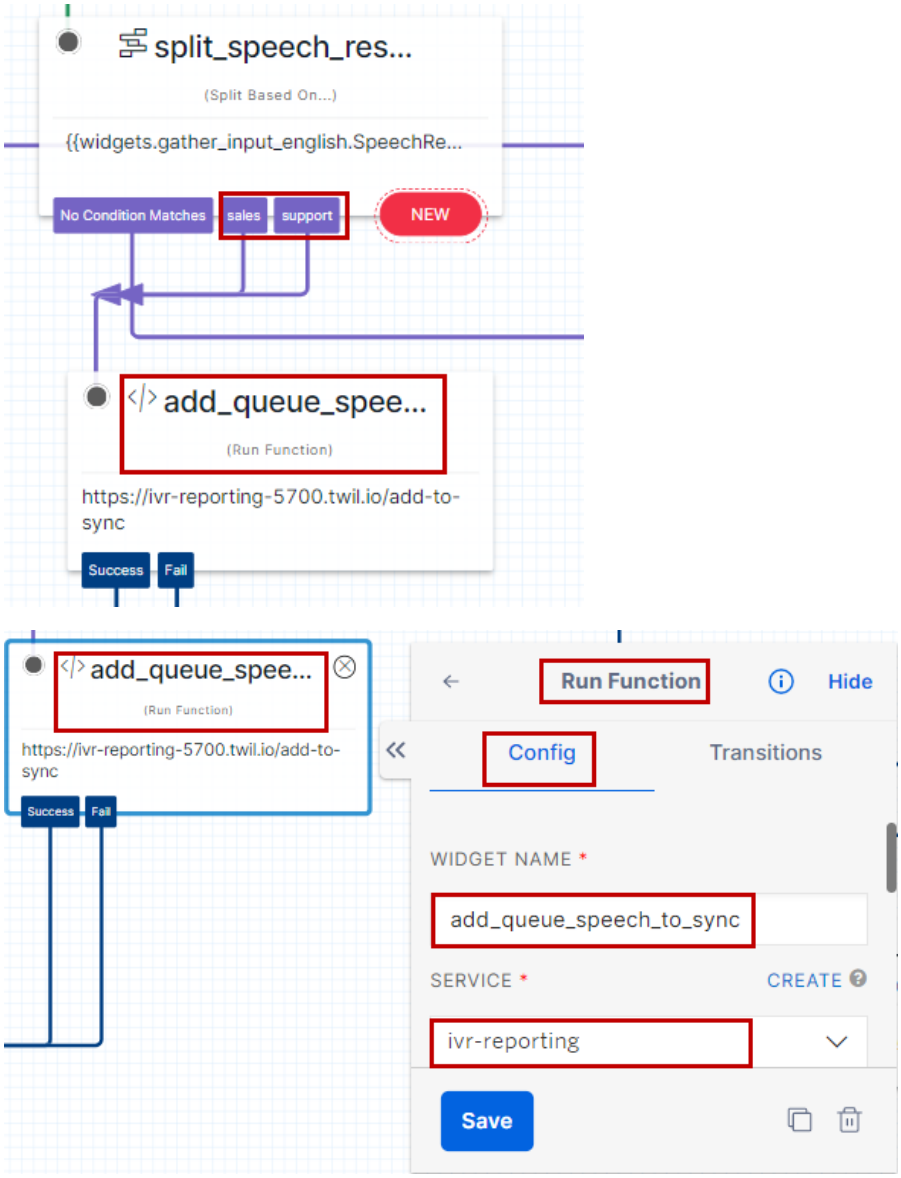



*Figure 33 Twilio IVR Studio Flow – add_queue_speech_to_sync*

**Widget: connect_call_to_support**

- **Success, Fail** options from the Widget **add_queue_speech_to_sync** is linked to Widget Connect Call To named **connect_call_to_support**
- **connect_call_to_support** Widget configuration is same as **connect_call_to_sales (Refer Figure 26)**

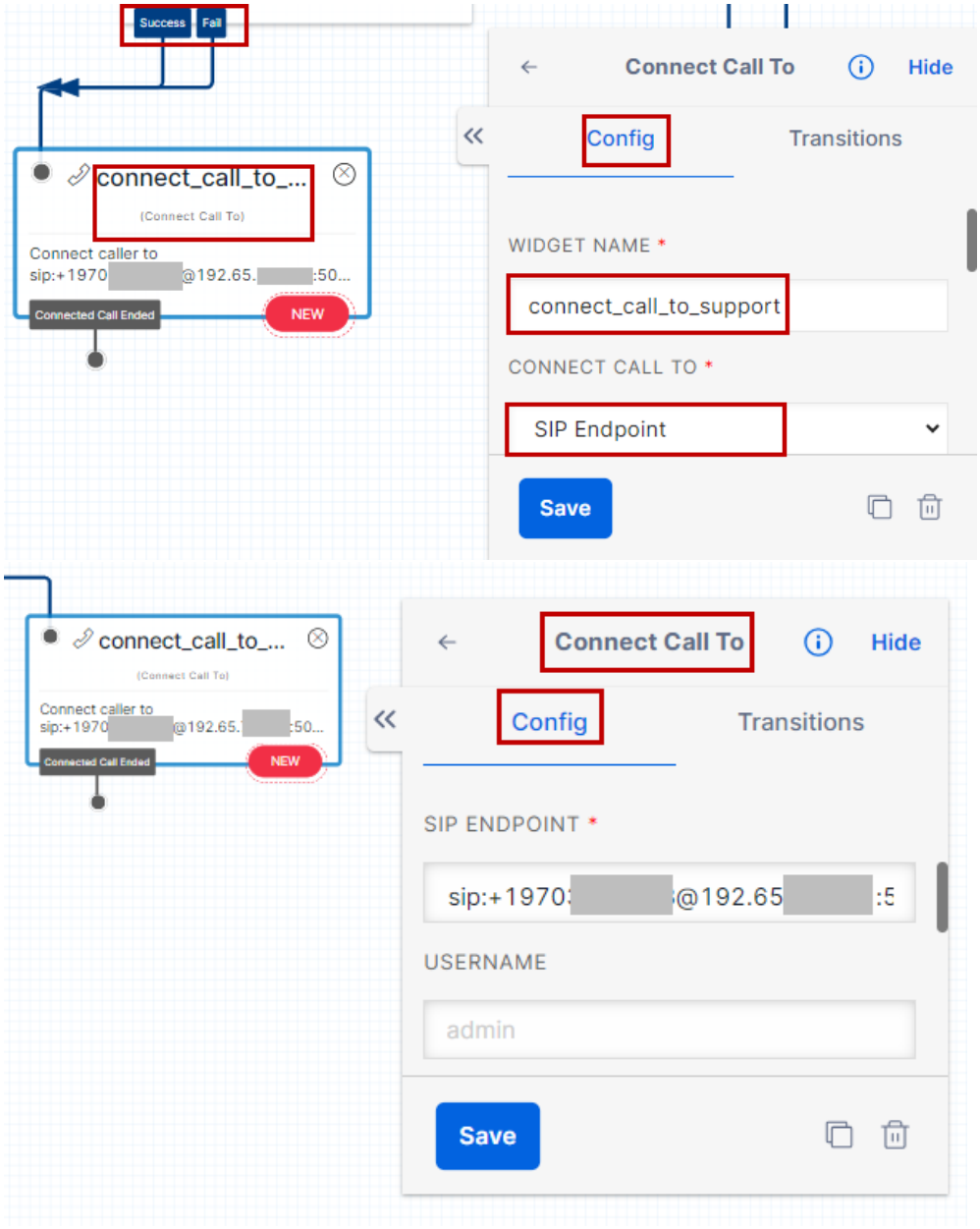



*Figure 34 Twilio IVR Studio Flow – connect_call_to_support*

**Widget: no_input_handler_4**

- **No Condition Matches** option from the Widget **split_speech_result_en** is linked to Widget Run Subflow named **no_input_handler_4**
- **no_input_handler_4** Widget configuration is same as **no_input_handler_1 (Refer Figure 18)**

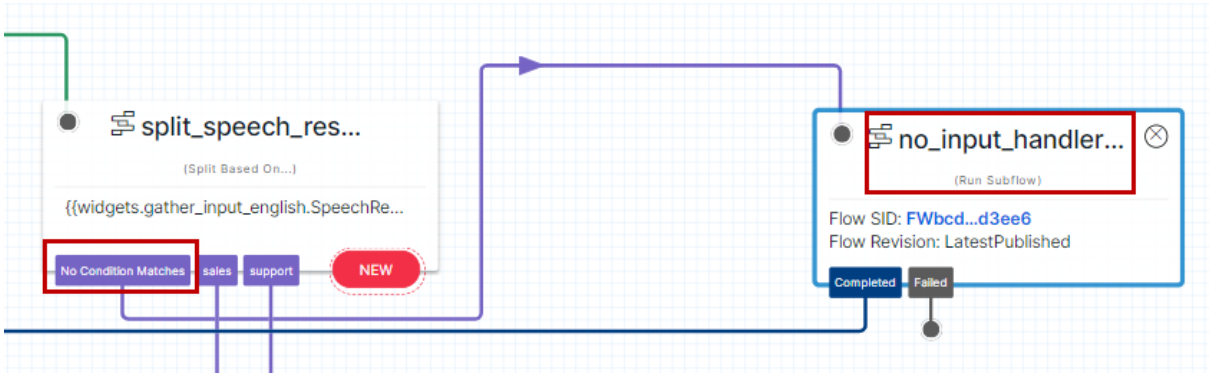



*Figure 35 Twilio IVR Studio Flow – no_input_handler_4*

### 6.1.5 Phone Numbers

- Navigate to **Phone Numbers > Manage > Active numbers**
- Under **Configure**,
  - *Configure with:* **Webhook, TwiMLBin, Function, Studio Flow, Proxy Service**
  - *A call comes in:* **Studio Flow**
  - *Flow:* **Dual Language IVR Test** (an IVR flow created in the Section 6.1.4)

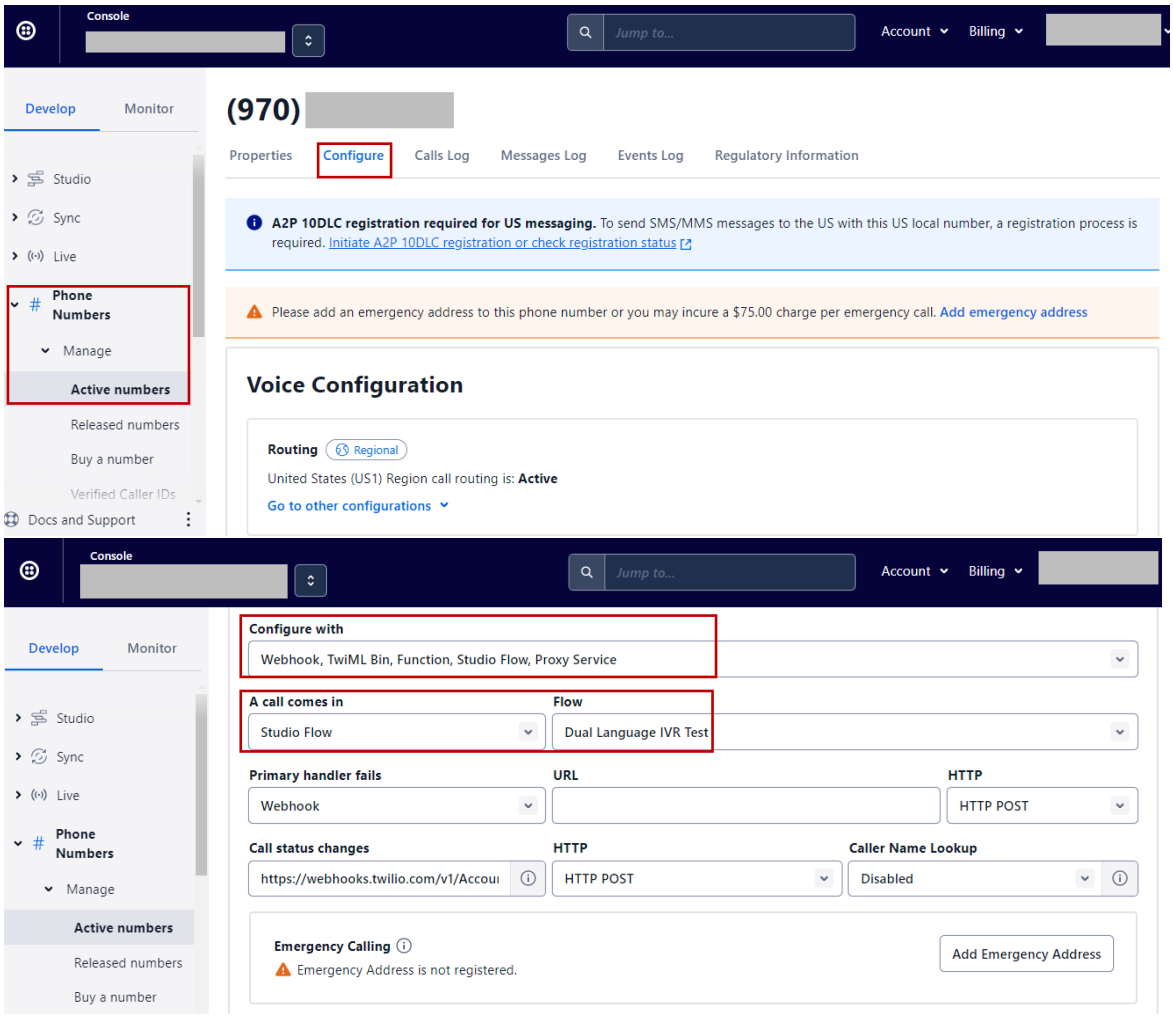



*Figure 36 Phone Numbers*

- Click **Save configuration**

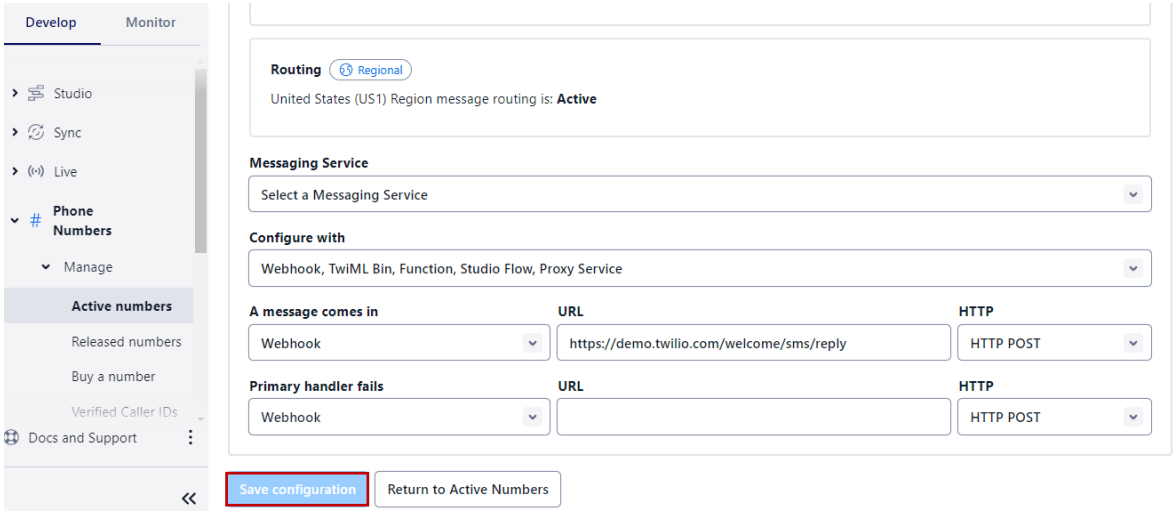



*Figure 37 Phone Number Continuation*

### 6.1.6 Functions

- Navigate to **Functions and Assets > Services**
- Click **Create Service**
- *Service Name*: **ivr_reporting**
- Click **Next**

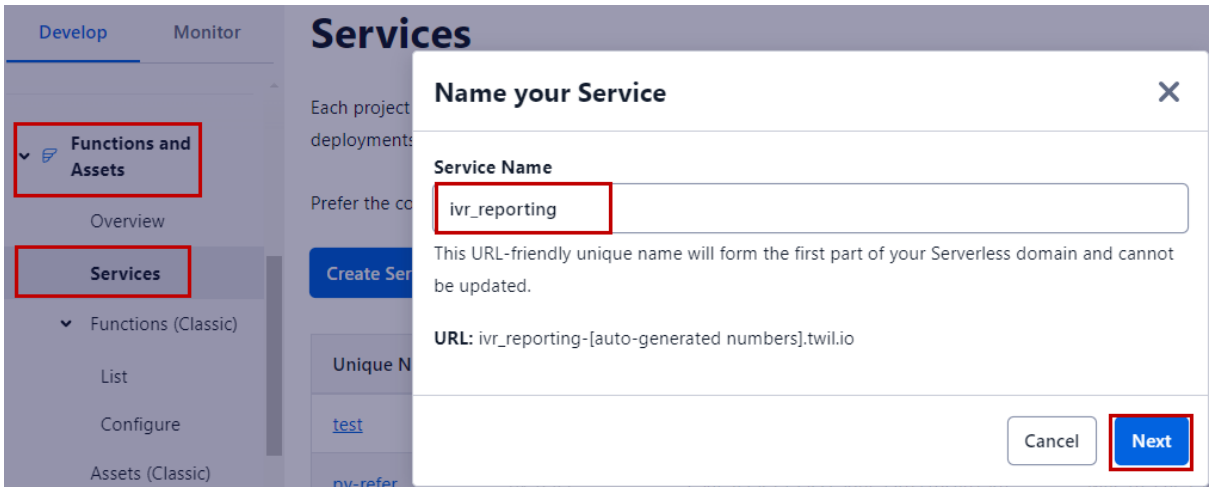



*Figure 38 Functions – Configure Service*

- Click **Create your function**
- *Functions:* Type the function name e.g. **add_to_sync**
- Click Add+ to create more functions


*Figure 39 Functions – Create function*

- **add_to_sync** function is made as protected using the **Protected with Lock** icon
- Javascript for the add_to_sync function is shown in the right pane

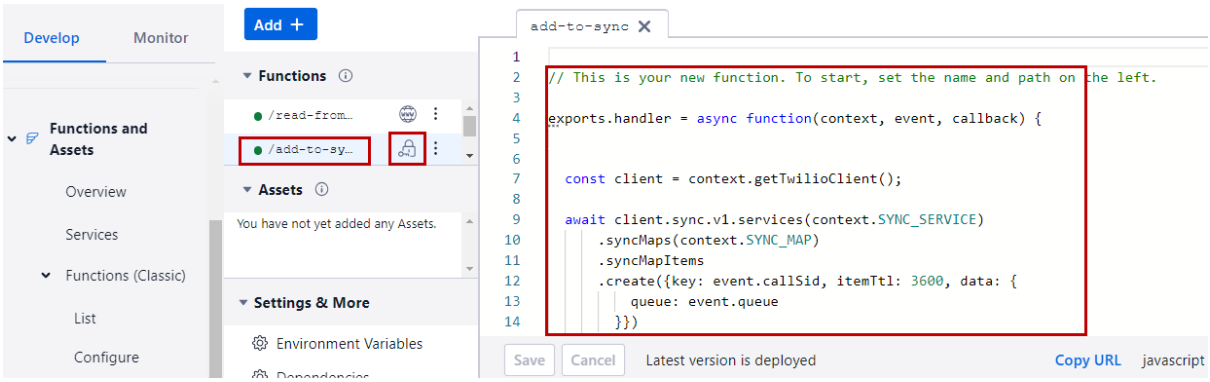



*Figure 40 Functions – add_to_sync*

- Full screen view of the entire code is shown below.
- The below code stores the **Twilio CallSid** and the **queue number** (1 or 2) for each inbound call to Twilio IVR DID in the Maps database
  The function parameters CallSid and queue number for add_to_sync is passed from the Twilio Studio IVR Flow (**Section 6.1.4 Figure 24**)
- Below is the javascript for **add_to_Sync**:

```
// This is your new function. To start, set the name and path on the left.


exports.handler = async function(context, event, callback) {
  const client = context.getTwilioClient();
  await client.sync.v1.services(context.SYNC_SERVICE)
      .syncMaps(context.SYNC_MAP)
      .syncMapItems
      .create({key: event.callSid, itemTtl: 3600, data: {
          queue: event.queue
        })
      .then(sync_map_item => {
        console.log(sync_map_item.key)
        return callback(null, sync_map_item.key);
      })
      .catch(error => {
        console.log(error);
```

```
        return callback(error, null);
      });
};
```

- Click on the **three dots** and Click **Copy URL** to copy the URL of add_to_sync function
- The URL for add_to_sync function is **https://ivr-reporting-XXXX.twil.io/add-to-sync**
- The above URL is given as input to the Twilio IVR studio flow (**Refer Section 6.1.4 Figure 24**)

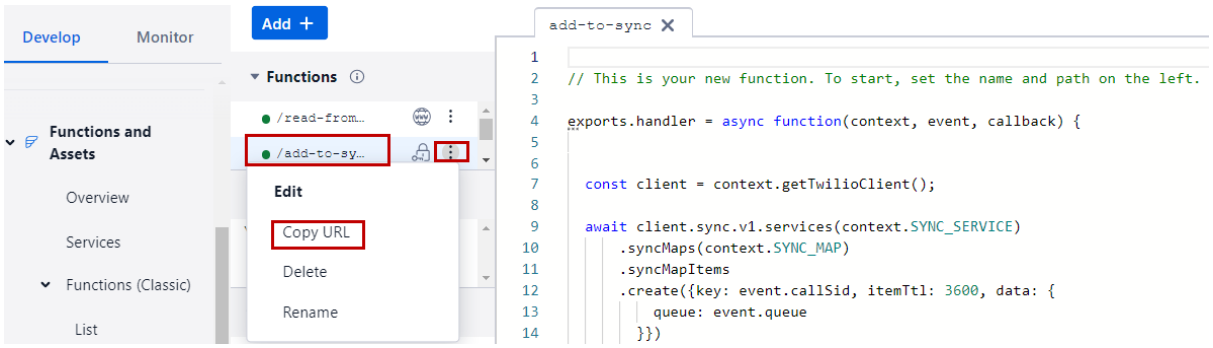



*Figure 41 Functions – add_to_sync Continuation*

- Click Add+ to create a function **read_from_sync**
- The URL for read_from_sync function is https://ivr-reporting-XXXX.twil.io/read-from-sync
- **read_from_sync** function is made as Publicly accessible using the **Globe** icon
- This function retrieves the Queue data and the digits pressed from the MAPS database for each Twilio callSid
- Below is the javascript for **read_from_sync**

```
// This is your new function. To start, set the name and path on the left.

exports.handler = async function(context, event, callback) {

  const client = context.getTwilioClient();

 await client.sync.v1.services(context.SYNC_SERVICE)

    .syncMaps(context.SYNC_MAP)

    .syncMapItems(event.callSid)

    .fetch()

    .then(sync_map_item => {
```

```
        console.log(sync_map_item.key + ": "+ sync_map_item.data);
        return callback(null, sync_map_item.data);
      })
      .catch(error => {
        console.log(error);
        return callback(error, null);
      });
};
```

- Navigate to **Settings & More > Environment Variables**
- *Key*: **SYNC_SERVICE**
- *Value*: Enter the Service SID value of **AvayaCallContext** Sync Service (**Refer Section 6.1.7**)
- Click **Add**
- *Key*: **SYNC_MAP**
- *Value*: Enter the Maps SID value of **AvayaCallContext** Sync Service (**Refer Section 6.1.7**)
- Click **Add**

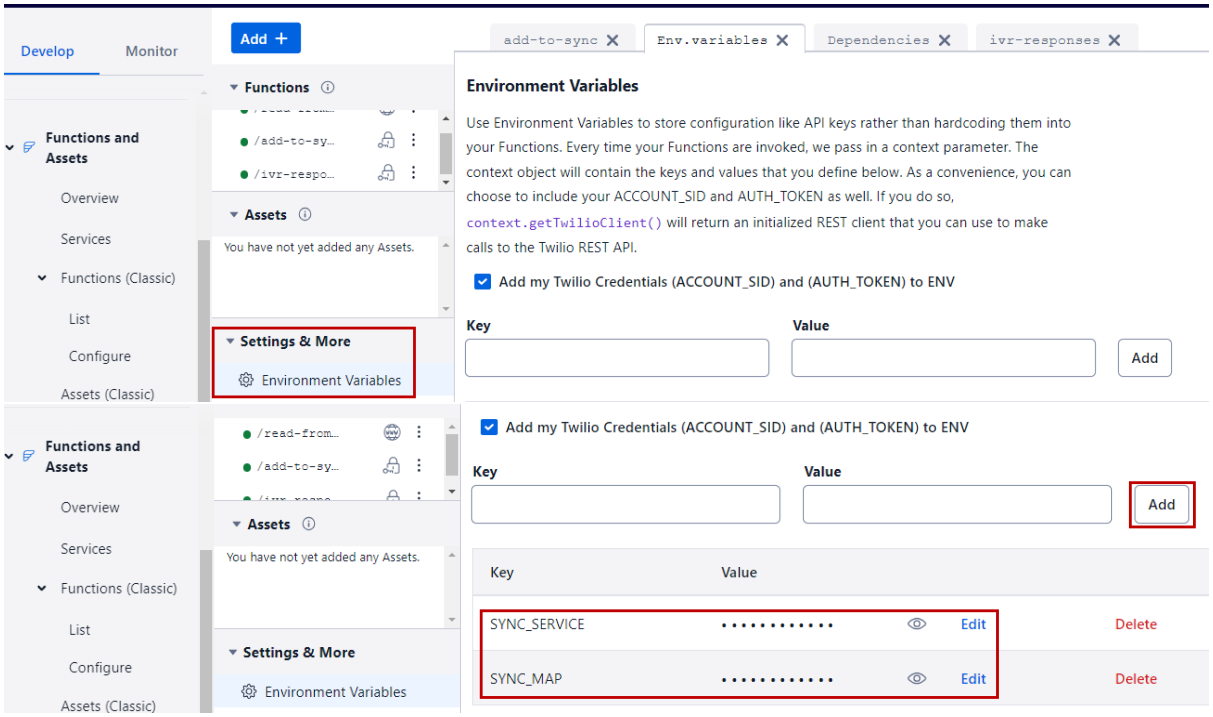



*Figure 42 Functions – Environment Variables*

- Navigate to **Settings & More > Dependencies**
- *Node Version*: **Node.js v16**
- Below are the **Modules** and the corresponding **Versions** which are added as dependencies

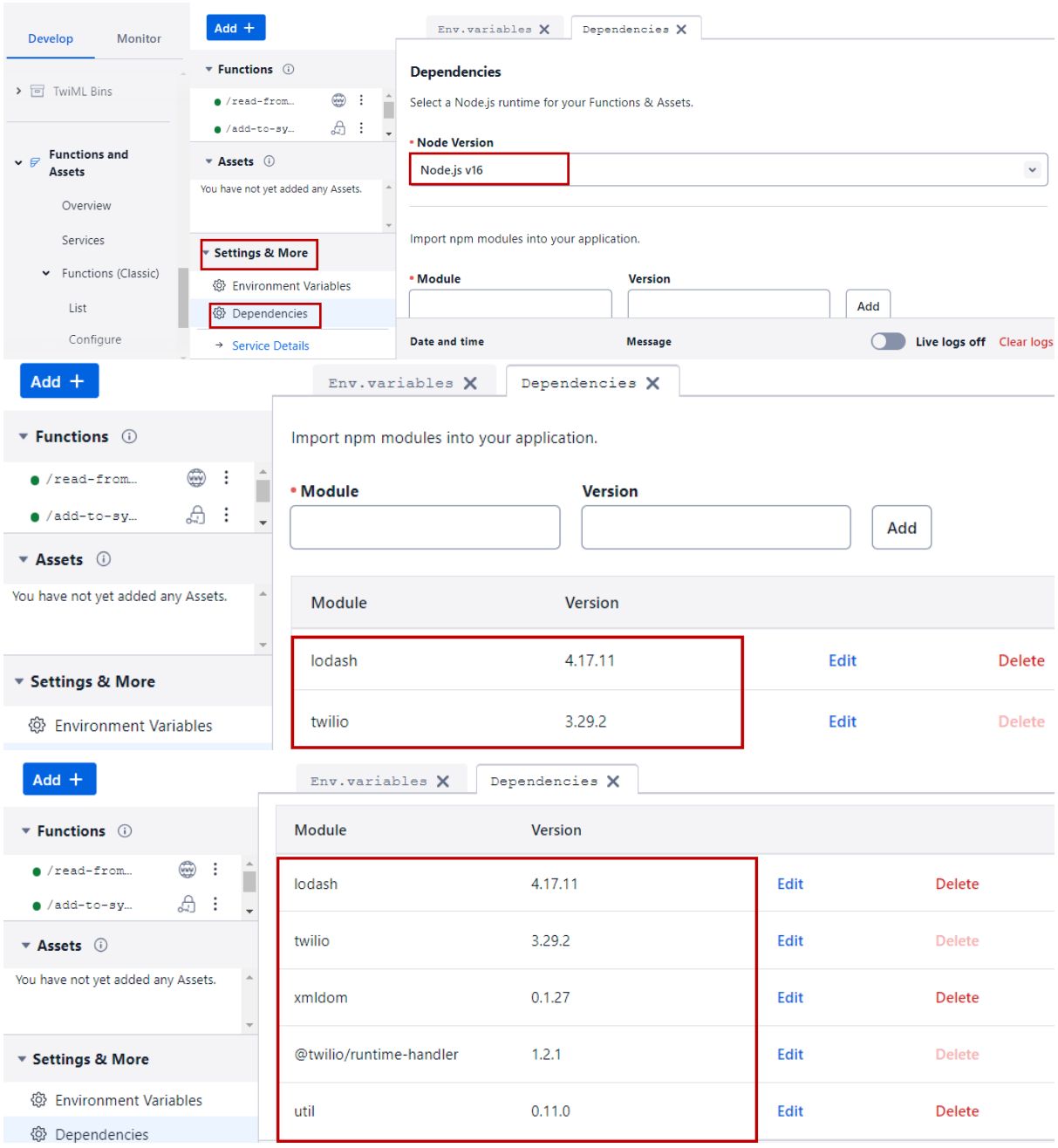



*Figure 43 Functions – Dependencies*

- Click **Deploy All** to validate and deploy the code

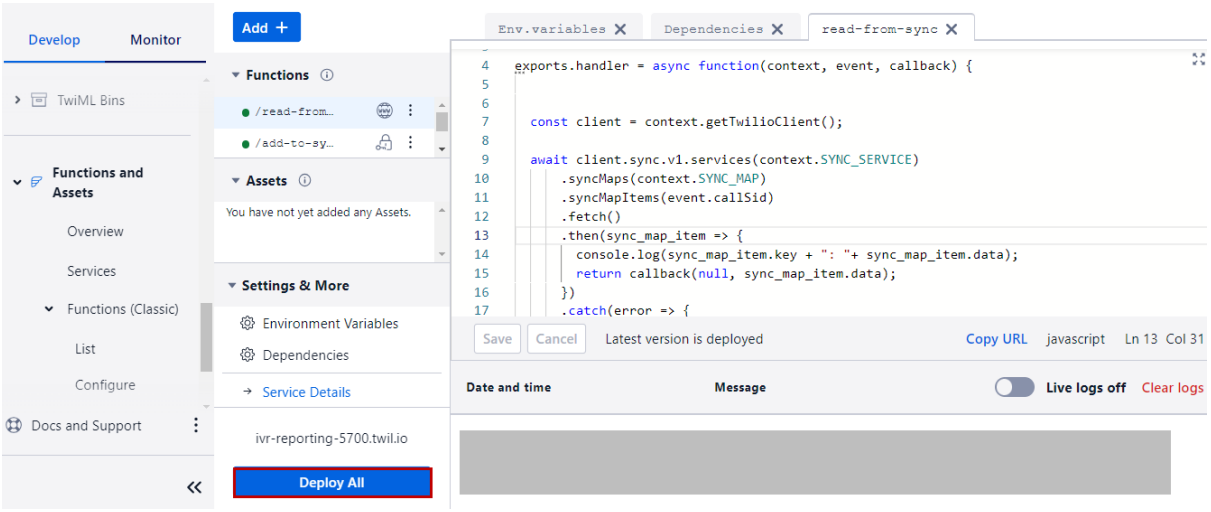



*Figure 44 Functions – Deploy All*

### 6.1.7 Services

- Navigate to **Sync > Services**
- Click **Create new Sync Service**
- Sync Service named **AvayaCallContext** is shown below
- Click **Save**

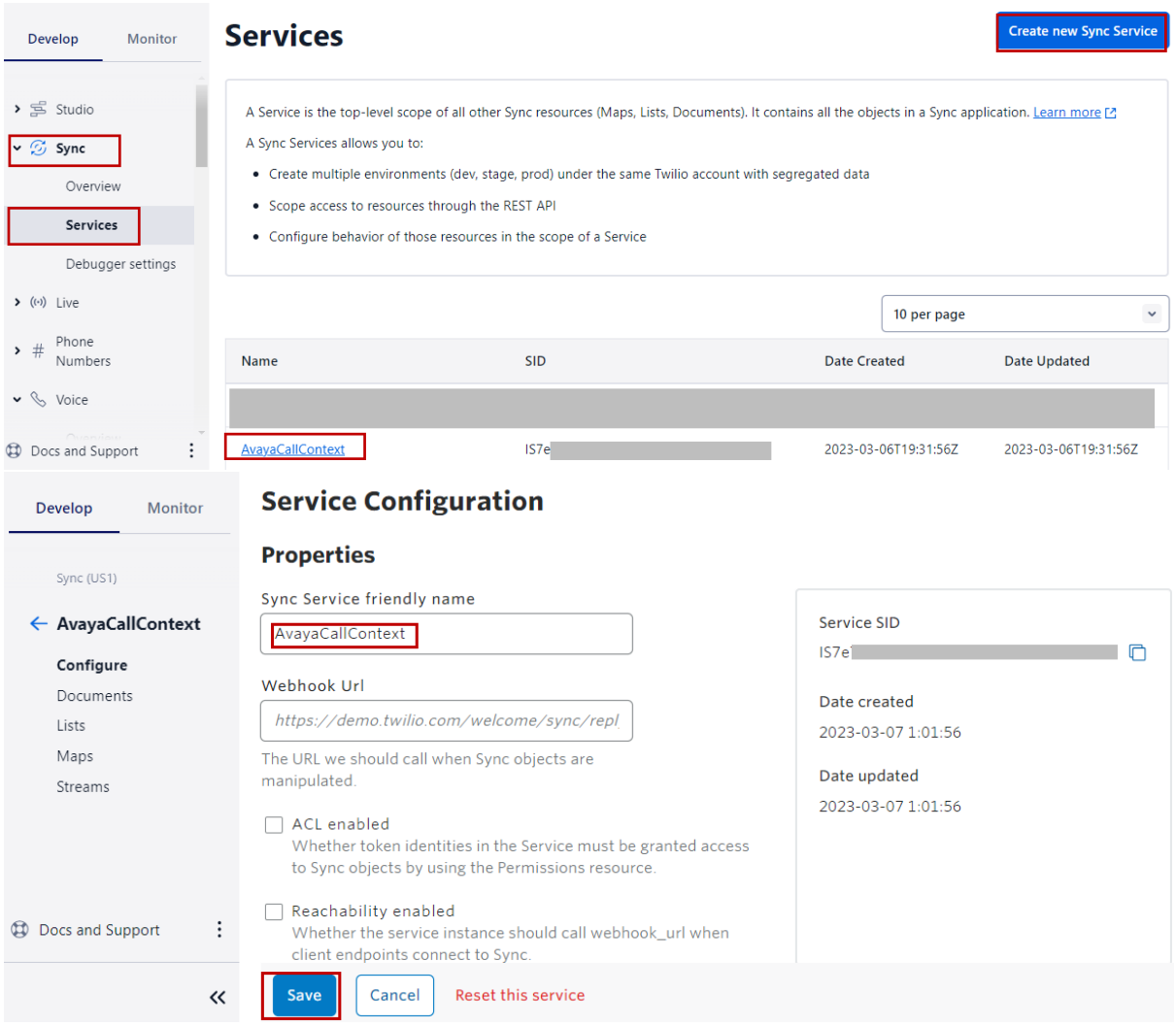



*Figure 45 Sync Service*

- Navigate to **AvayaCallContext > Maps**
- Click **Create new Sync Map**
- Sync Map named **MP21XXX** is shown below
- Click **Map Name MP21XXX**
- **KEY** value i.e. Twilio callSid and the **Map Item Data** is added in to the database as shown below

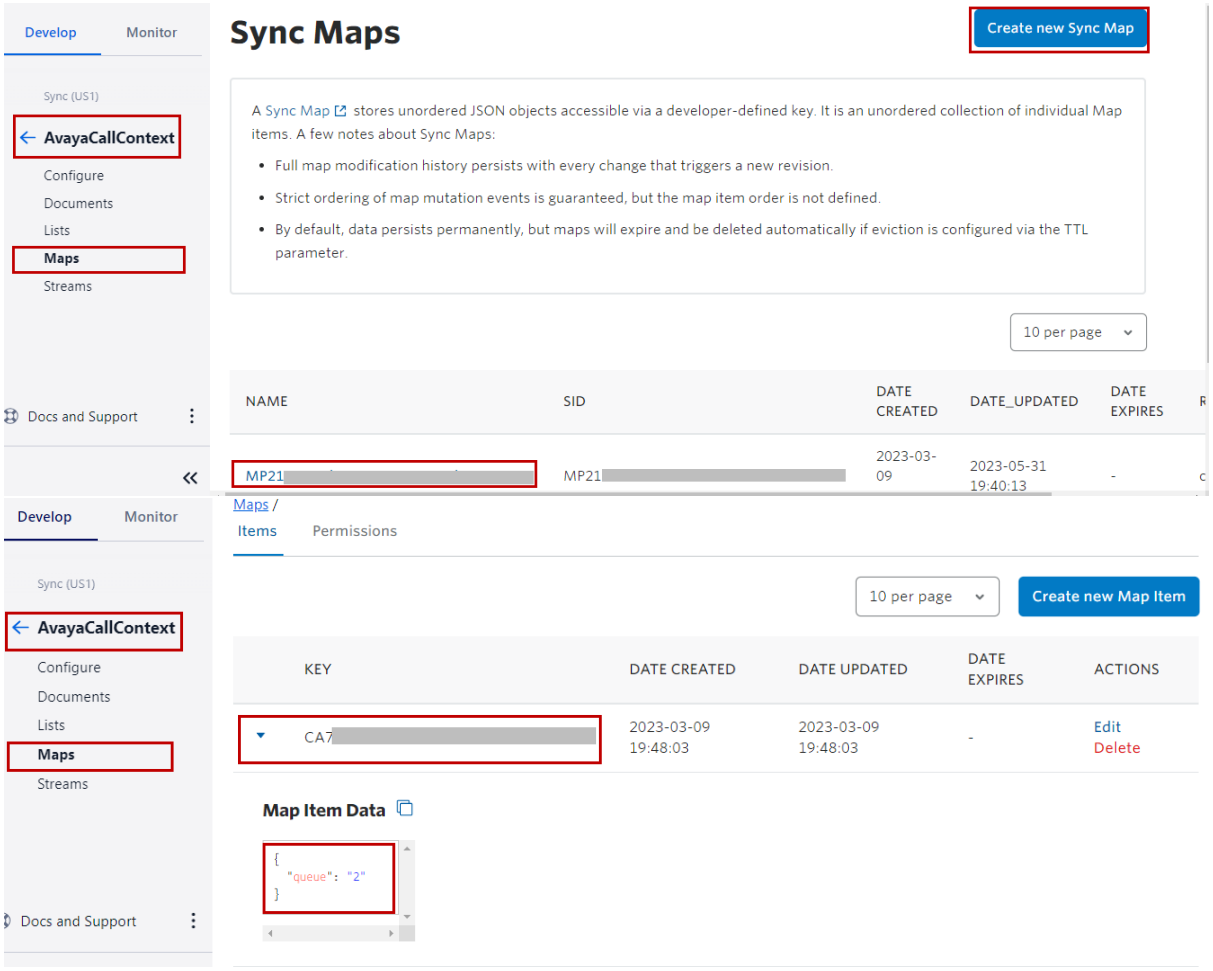



*Figure 46 Sync Maps*

## 6.2 Avaya SBCE Configuration

### 1.1.1 Avaya SBCE Login

- Log into Avaya Session Border Controller for Enterprise (SBCE) web interface by typing **“https://X.X.X.X/sbc”**
- Enter the **Username** and **Password**
- Click **Log In**


*Figure 47 Avaya SBCE Login*

- Under **Device**, select **ASBCE10** from drop down to expand the configuration for Avaya SBCE

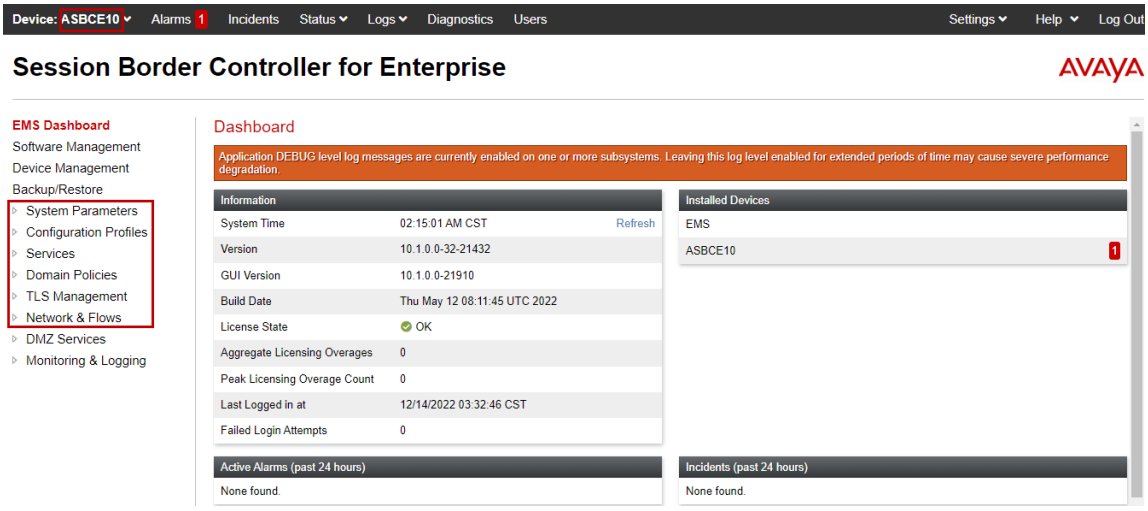



*Figure 48 Selection of Avaya SBCE Device*

### 1.1.2 Server Interworking

**Server Interworking for Avaya SM**

- Navigate to **Configuration Profiles > Server Interworking**
- Select the predefined Interworking Profile **avaya-ru**, click **Clone**
- Set *Clone Name*: **AASM.10.1**
- Click **Finish**

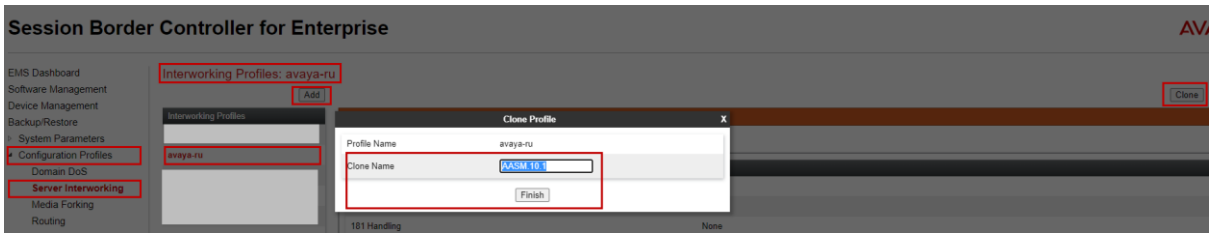



*Figure 49 Server Interworking Profile for Avaya SM*

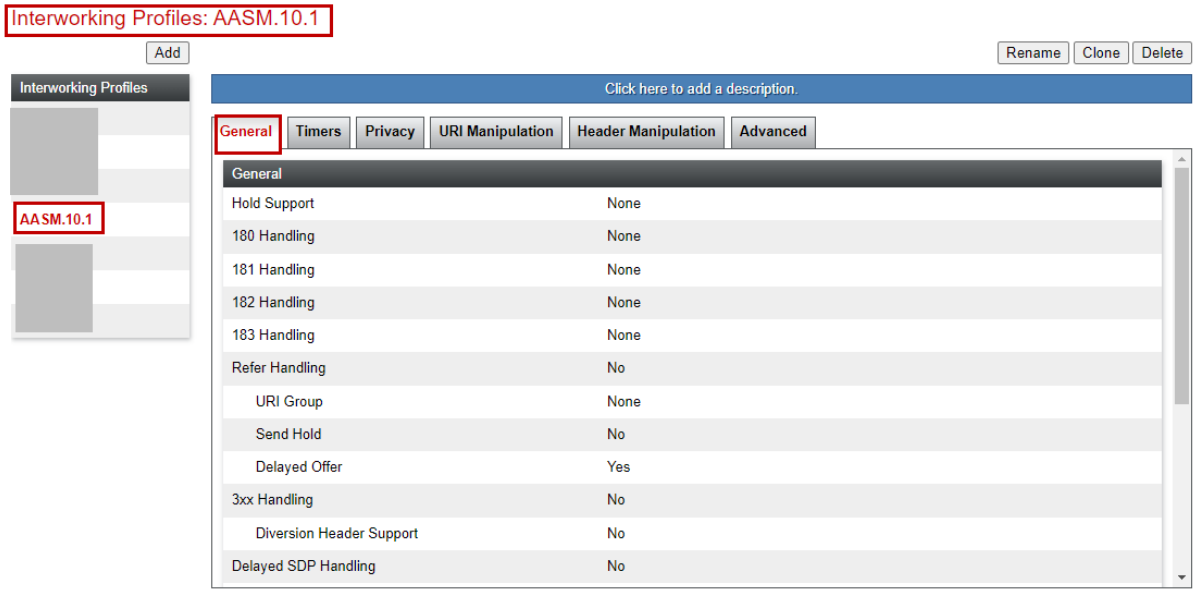
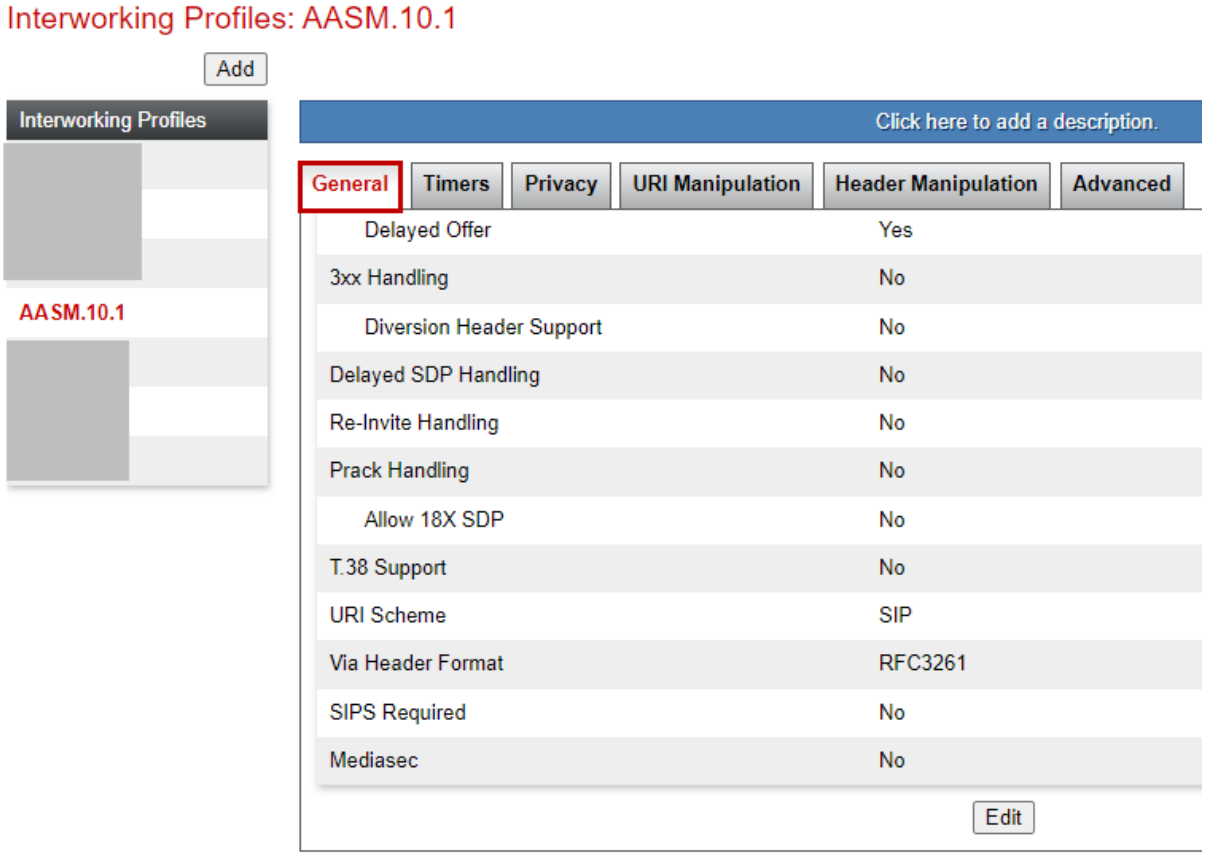



*Figure 50 Server Interworking Profile for Avaya SM Continuation*



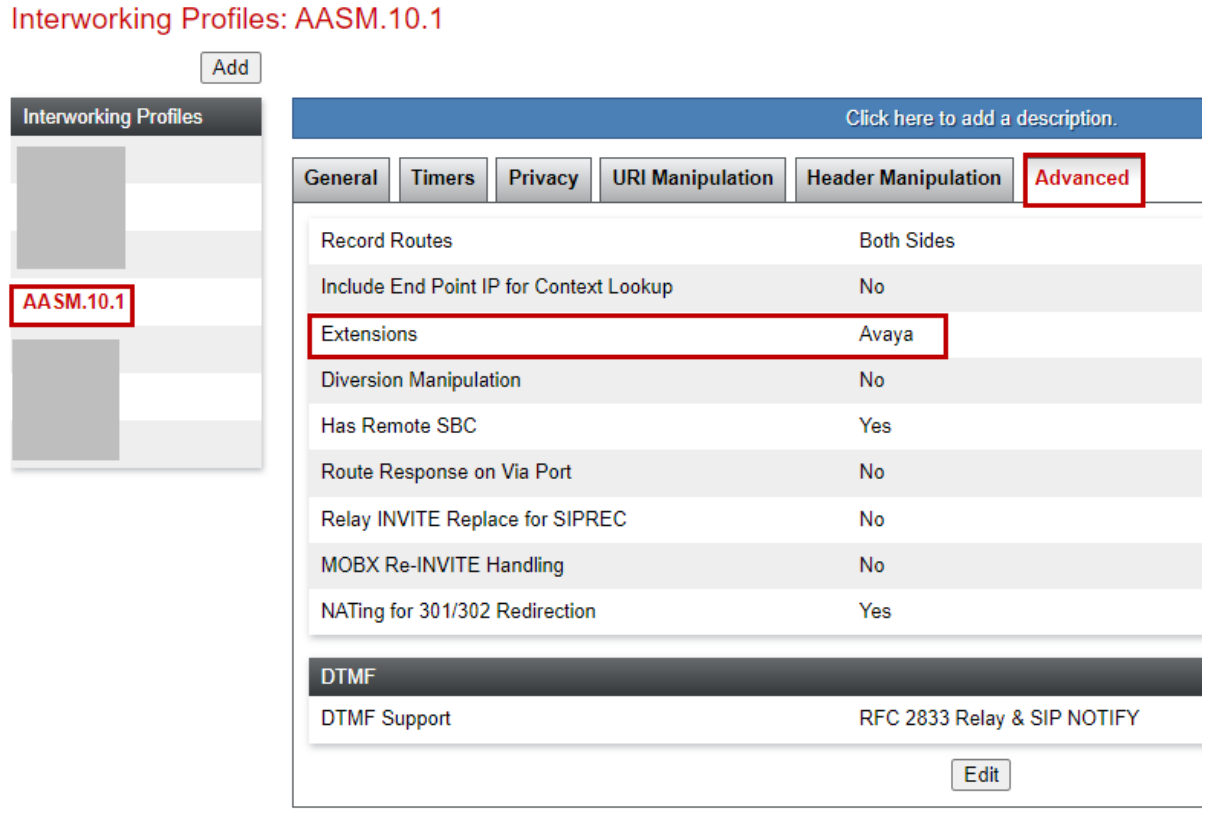




*Figure 51 Server Interworking Profile for Avaya SM Continuation*

### 1.1.3 SIP Servers

**SIP Server for Avaya SM**

- Navigate to **Services > SIP Servers**
- Click **Add**
- Set *Profile Name*: **Avaya**
- Click **Next**
- Set *Server Type*: Select **Trunk Server** from the drop down
- Set *IP Address/FQDN*: Enter the **Avaya Aura Session Manager SIP IP Address**
- Set *Port*: **5060**
- Set *Transport*: **UDP**
- Click **Finish**

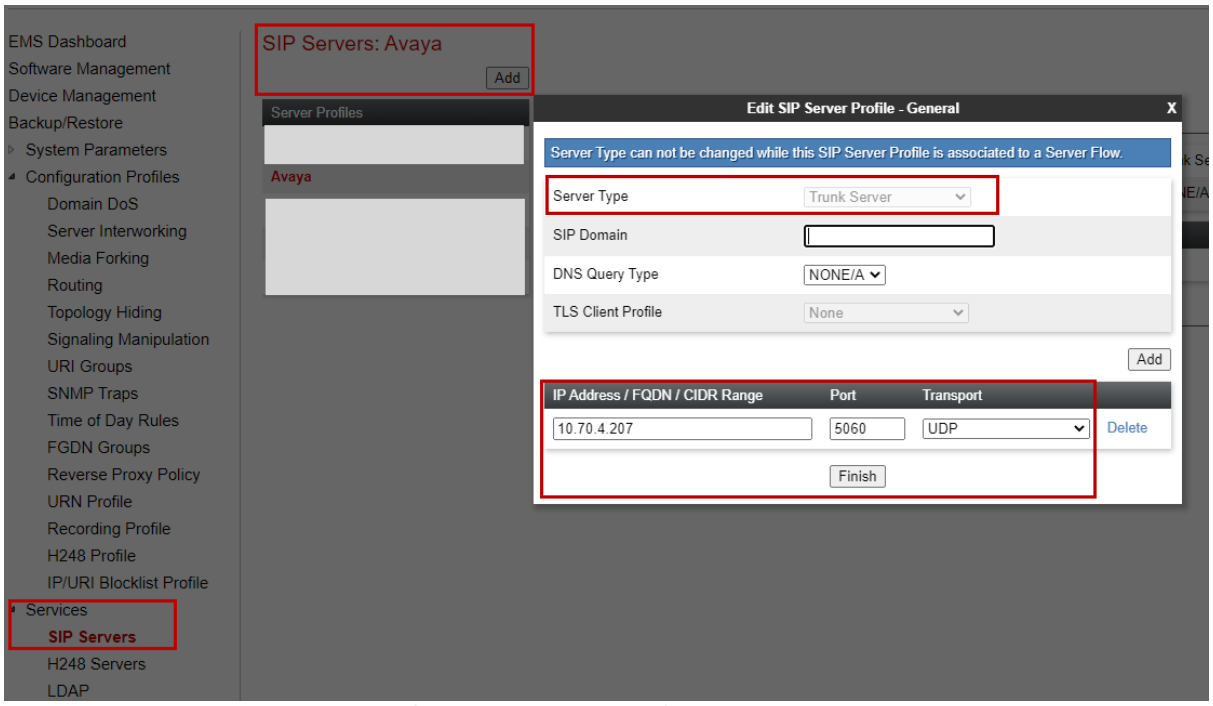



*Figure 52 SIP Server for Avaya SM*

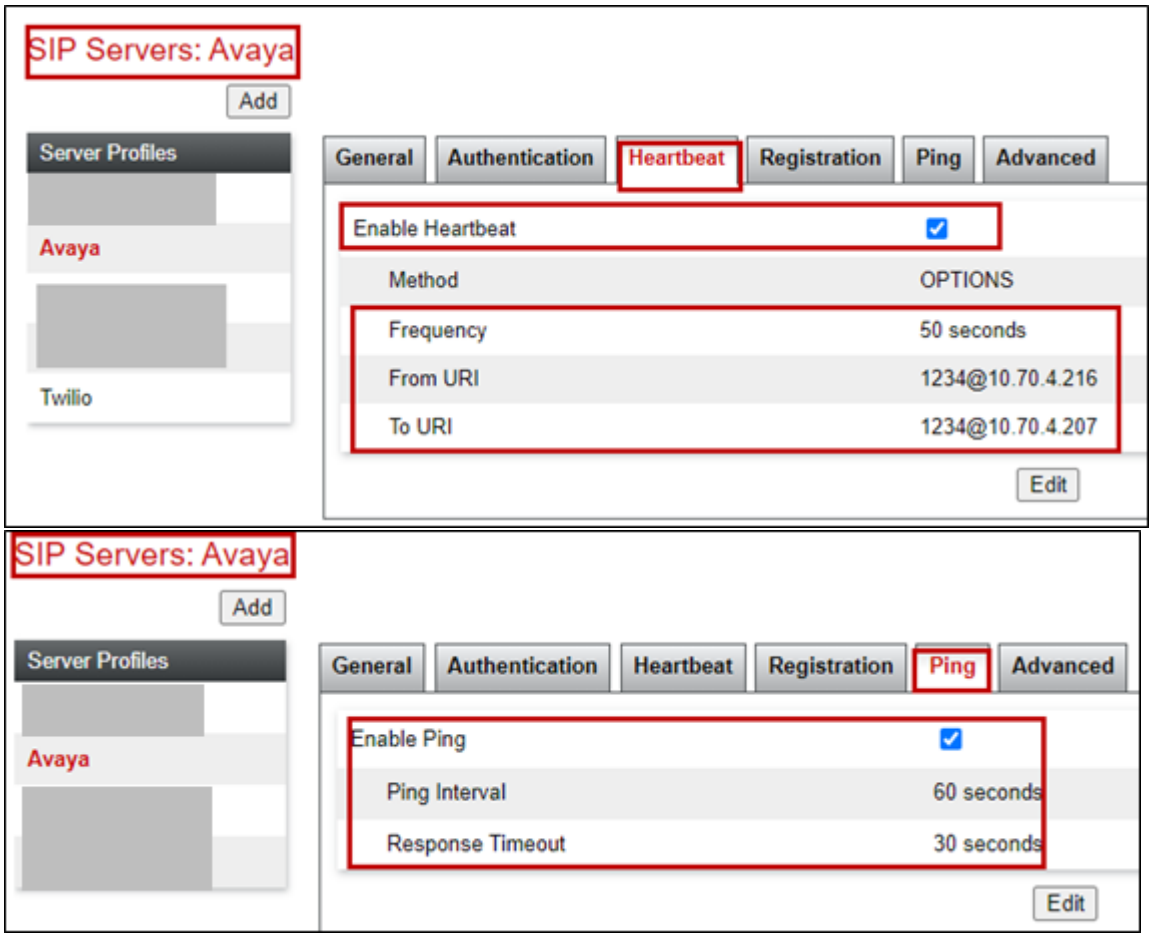



*Figure 53 SIP Server for Avaya SM Continuation*

- Navigate to **Advanced** tab
- Set *Enable Grooming*: **Checked**
- Set *Interworking Profile*: Select **AASM.10.1** (created in section 6.2.2)

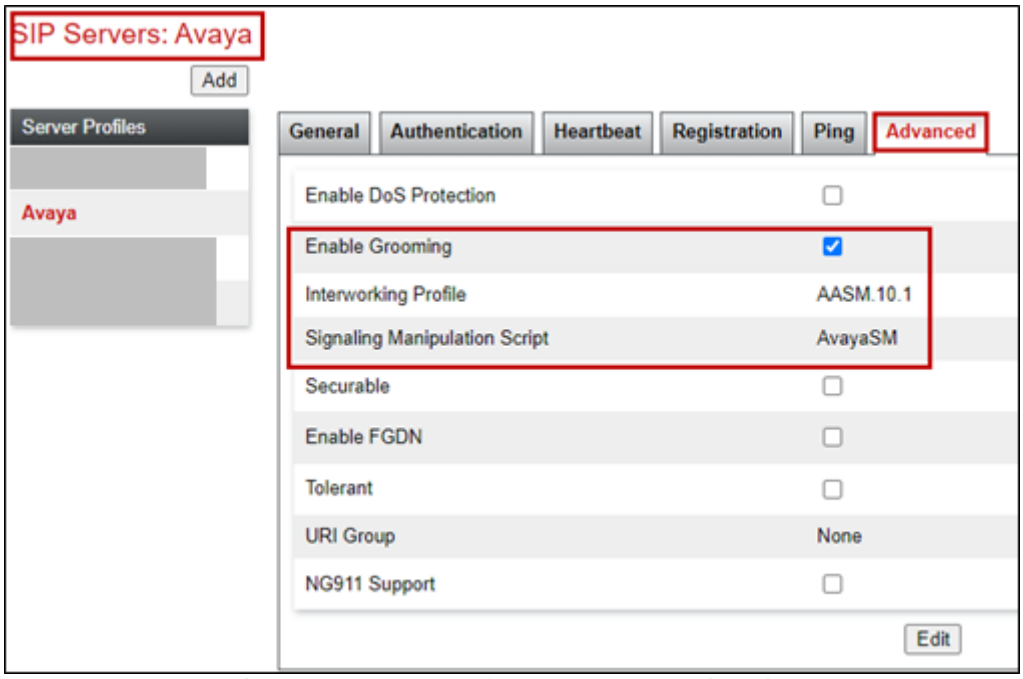



*Figure 54 SIP Server for Avaya SM Continuation*

### SIP Server for Twilio

- Navigate to **Services > SIP Servers**
- Click **Add**
- Set *Profile Name*: **Twilio**
- Click **Next**
- Set *Server Type*: Select **Trunk Server** from the drop down
- Set *IP Address/FQDN*: Enter the **Twilio Host Name**
- Set *Port*: **5060**
- Set *Transport*: **UDP**
- Click **Finish**

## Session Border Controller for Enterprise

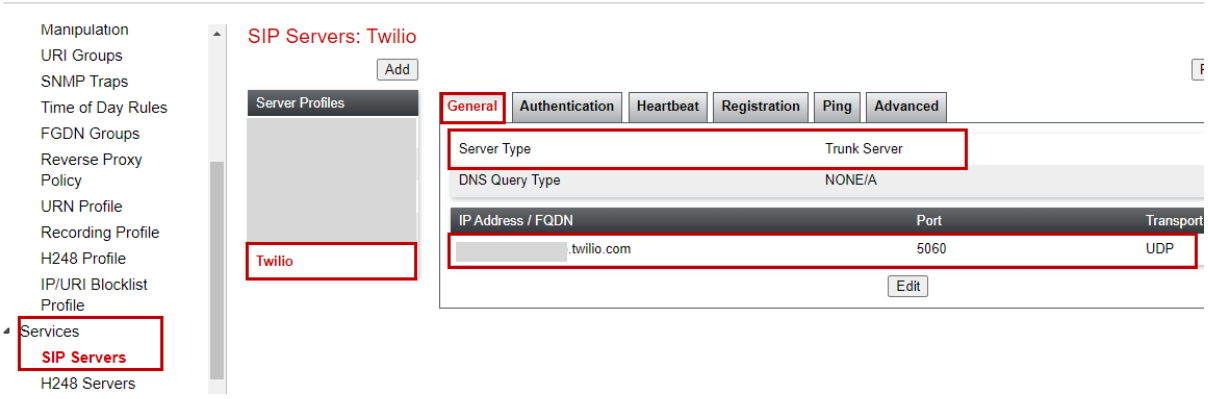



*Figure 55 SIP Server for Twilio*

- Navigate to **Authentication** tab
- *Enable Authentication*: **Enabled**
- *User Name*: Enter the **User Name** provided by Twilio
- *Password*: Enter the **Password** provided by Twilio

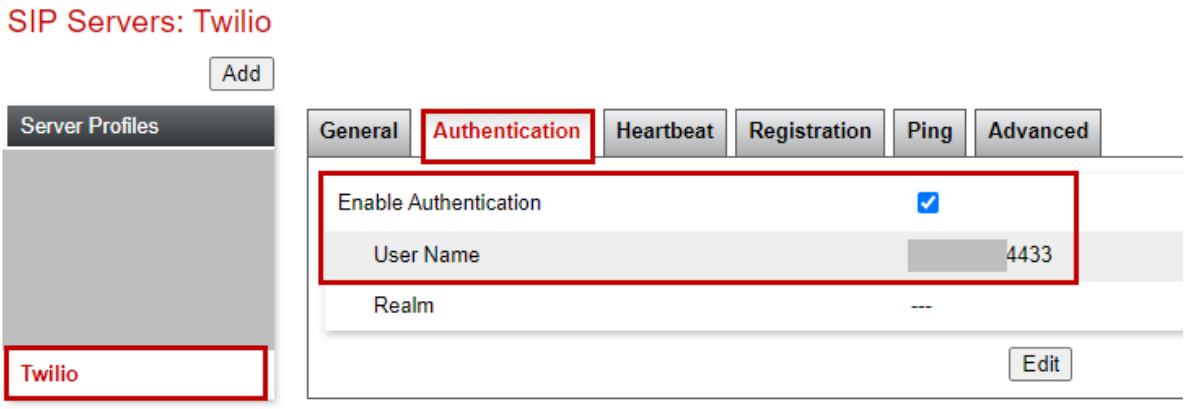



*Figure 56 SIP Server for Twilio Continuation*

- Navigate to **Hearbeat** tab
- *Enable Heartbeat*: **Enabled**
- *Method:* **OPTIONS**
- *Frequency:* **60 Seconds**
- *From URI:**1234@192.65.x.x***
- *To URI***: 1234@xxxx.twilio.com**

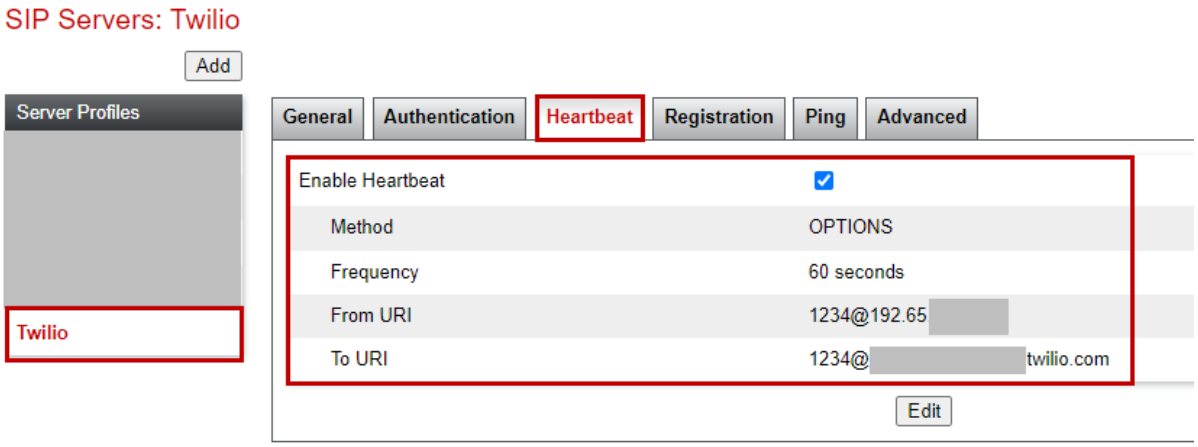



*Figure 57 SIP Server for Twilio Continuation*

- Navigate to **Ping** tab
- *Enable Ping*: **Enabled**
- *Ping Interval:* **60 seconds**
- *Response Timeout:* **30 Seconds**

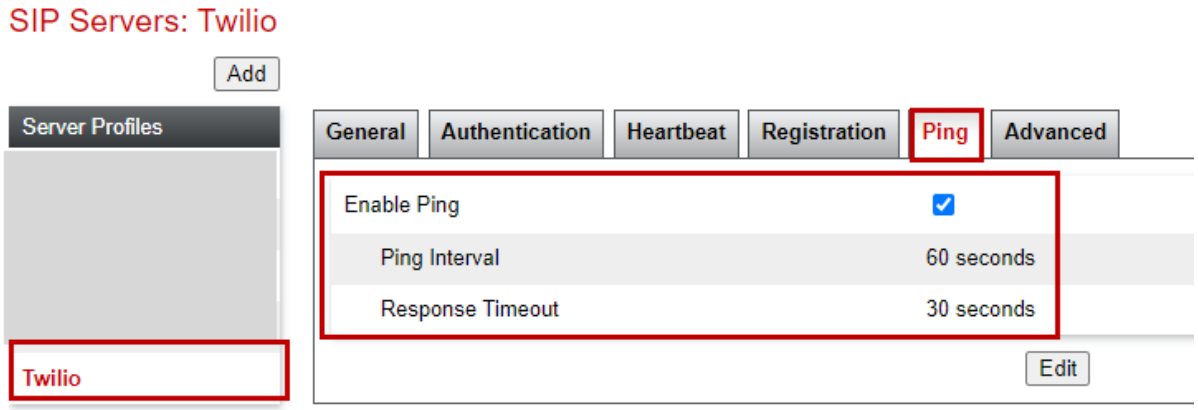



*Figure 58 SIP Server for Twilio Continuation*

- Navigate to **Advanced** tab

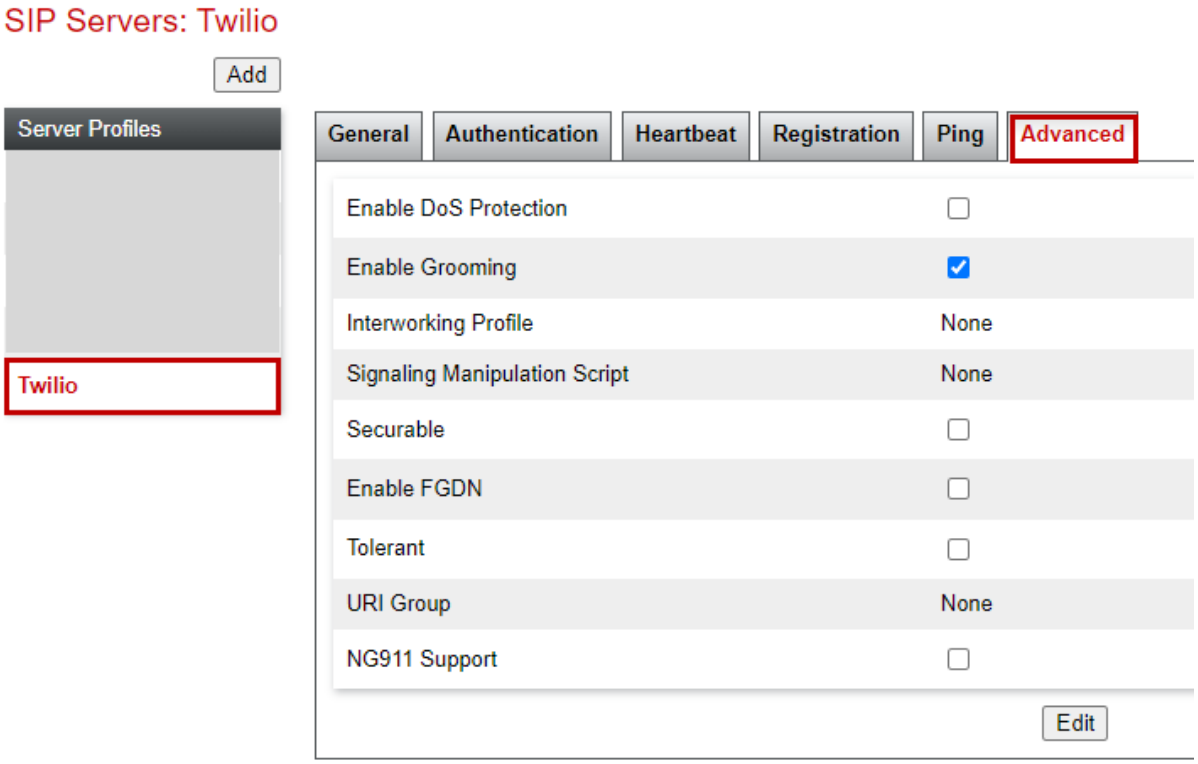



*Figure 59 SIP Server for Twilio Continuation*

### 1.1.4 Topology Hiding

**Topology Hiding profile for Twilio**

- Set *Header*: **Request-Line, To** are selected
- Set *Replace Action*: **Overwrite**
- Set *Overwrite Value*: Enter **Twilio Host name**

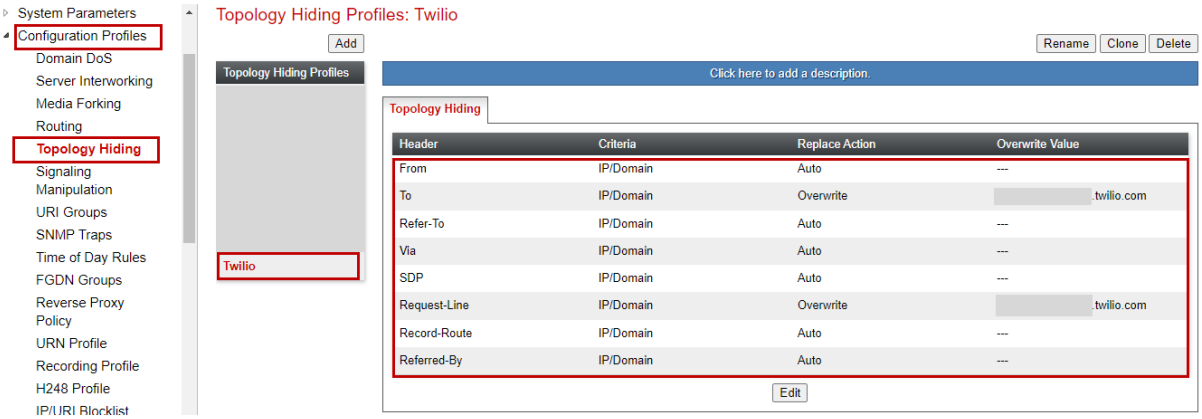



*Figure 60 Topology Hiding Profile for Twilio*

### 1.1.5 Routing

**Routing for Avaya SM**

- Navigate to **Configuration Profiles > Routing**
- Create Routing Profile named **Avaya_SM**
- At Routing Profile Window, Click **Add**
- Set *SIP Server Profile*: Avaya (configured in section 6.2.3)
- The **Server IP**, **Port** and **Transport Protocol** populates automatically

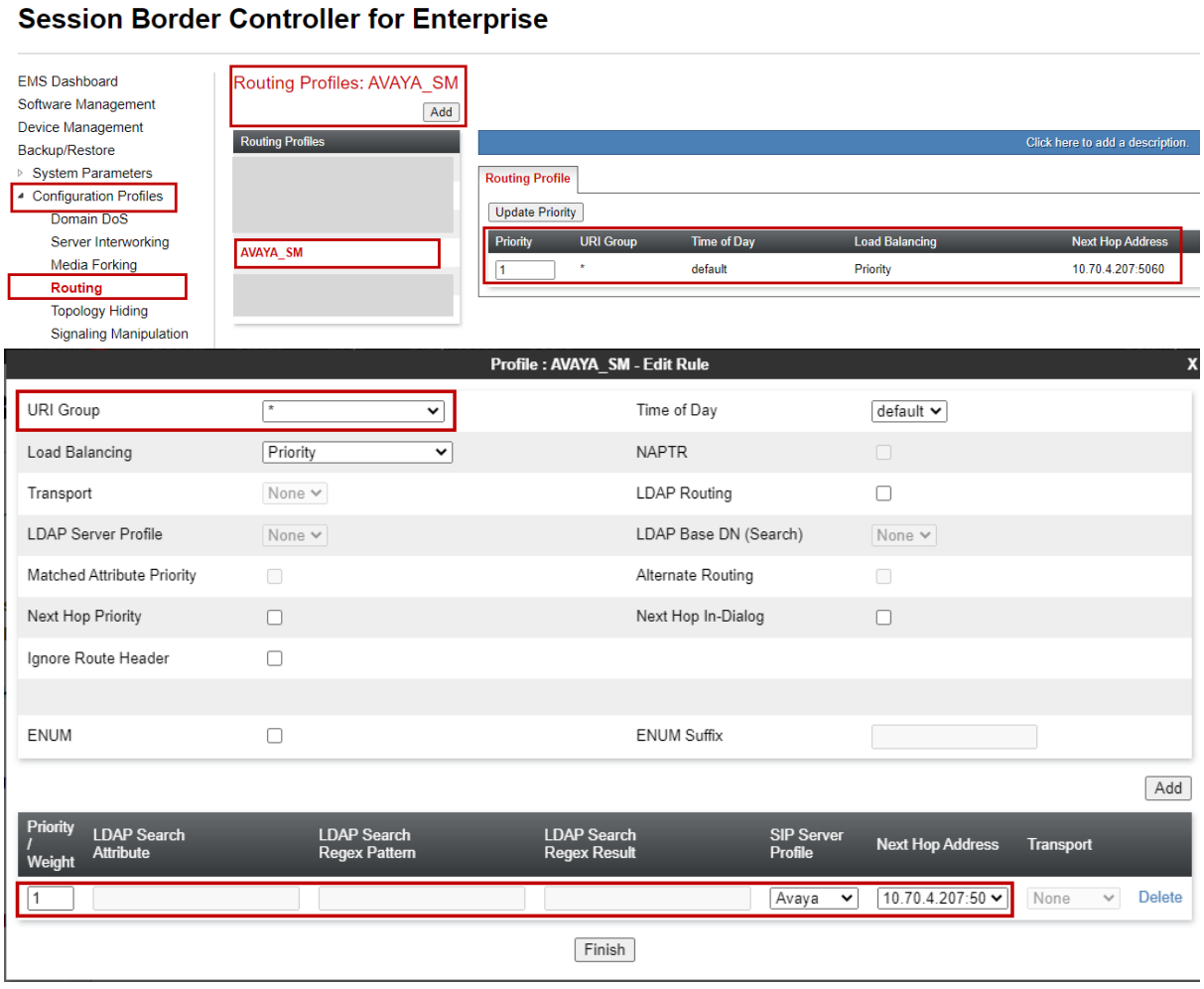



*Figure 61 Routing for Avaya SM*

### Routing for Twilio

- Set *SIP Server Profile*: **Twilio** (configured in section 6.2.3)
- The **Server IP**, **Port** and **Transport Protocol** populates automatically

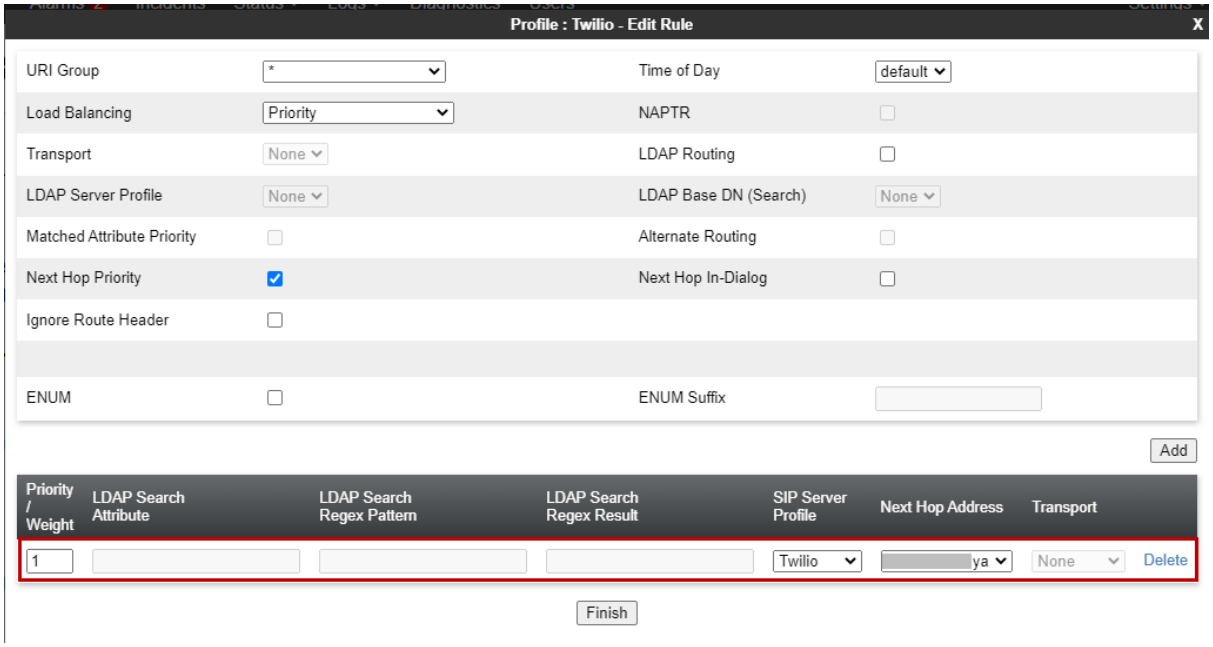



*Figure 62 Routing for Twilio*

### 1.1.6 Media Rules

**Media Rule for Twilio**

- Navigate to **Domain Policies > Media Rules**
- Create Media Rules named **Twilio**

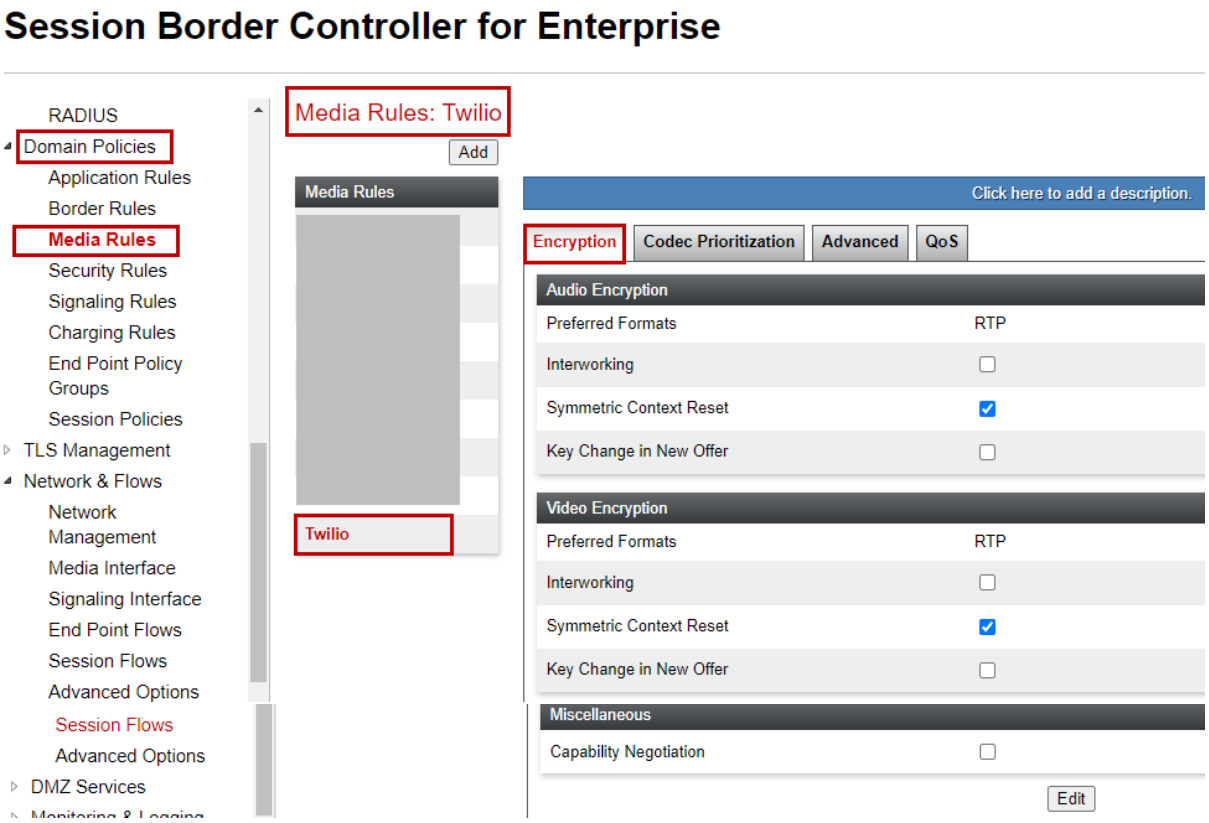



*Figure 63 Media Rules for Twilio*

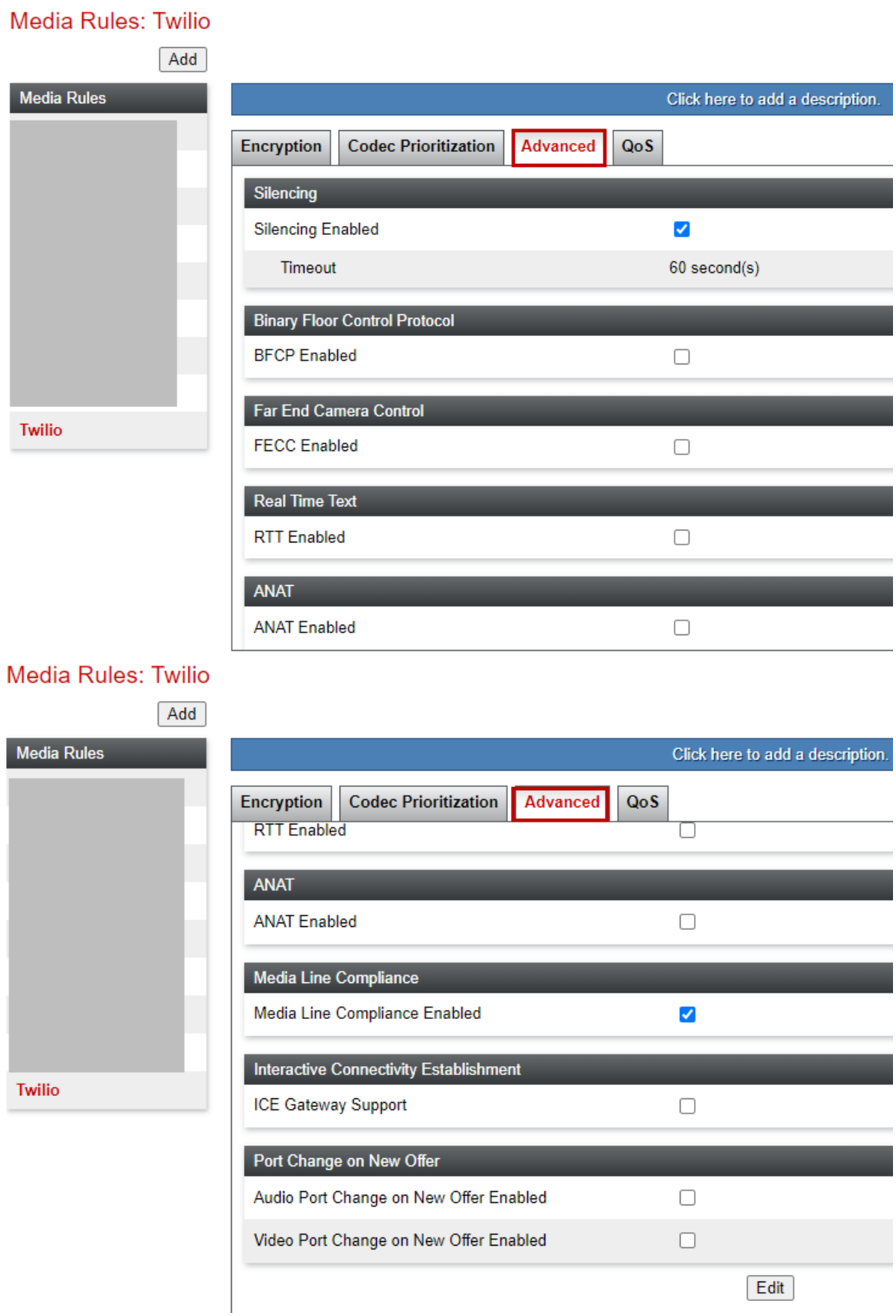



*Figure 64 Media Rules for Twilio Continuation*

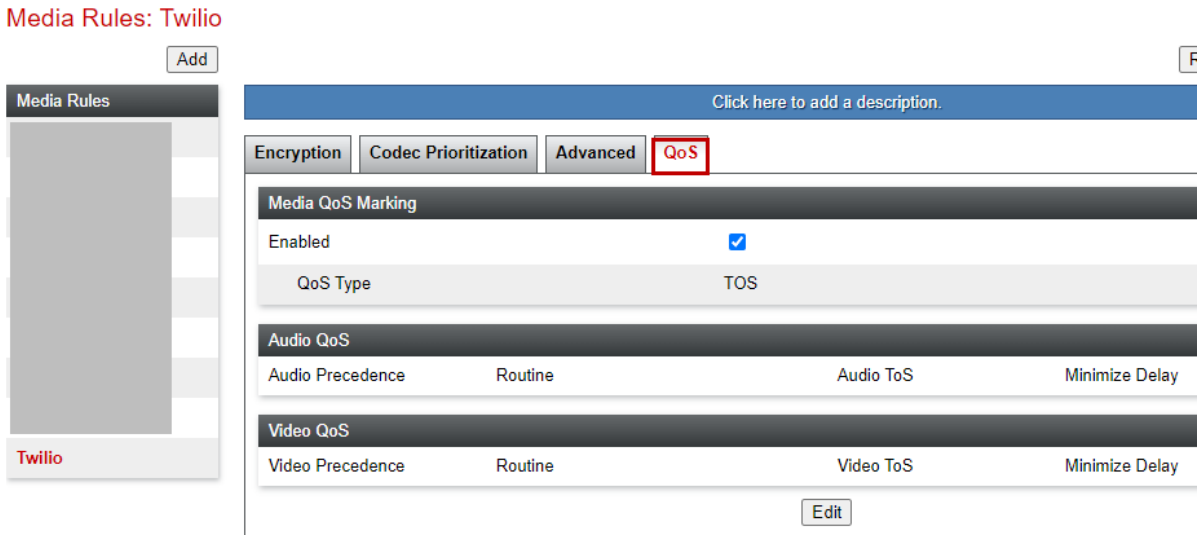



*Figure 65 Media Rules for Twilio Continuation*

### Media Rule for Avaya SM

Media Rule for Avaya SM is same as Twilio. It is named as **AvayaSM.** (No screenshot of AvayaSM Media rule is shown)

### 1.1.7 Signaling Rules

**Signaling Rules for Avaya SM**

- Navigate to **Domain Policies > Signaling Rules**
- Create Signaling Rules named **Avaya SM**

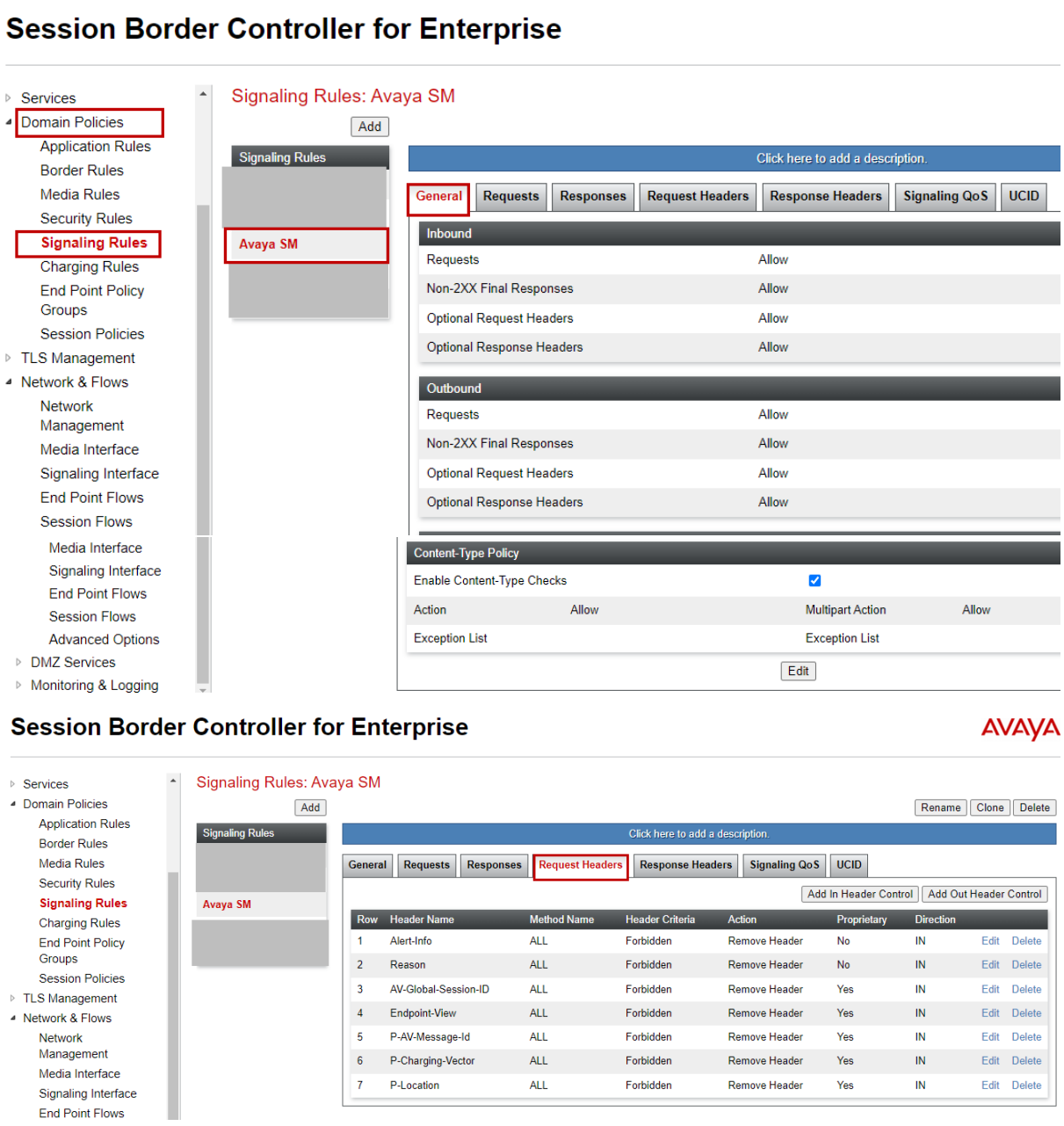






*Figure 66 Signaling Rules for Avaya SM*

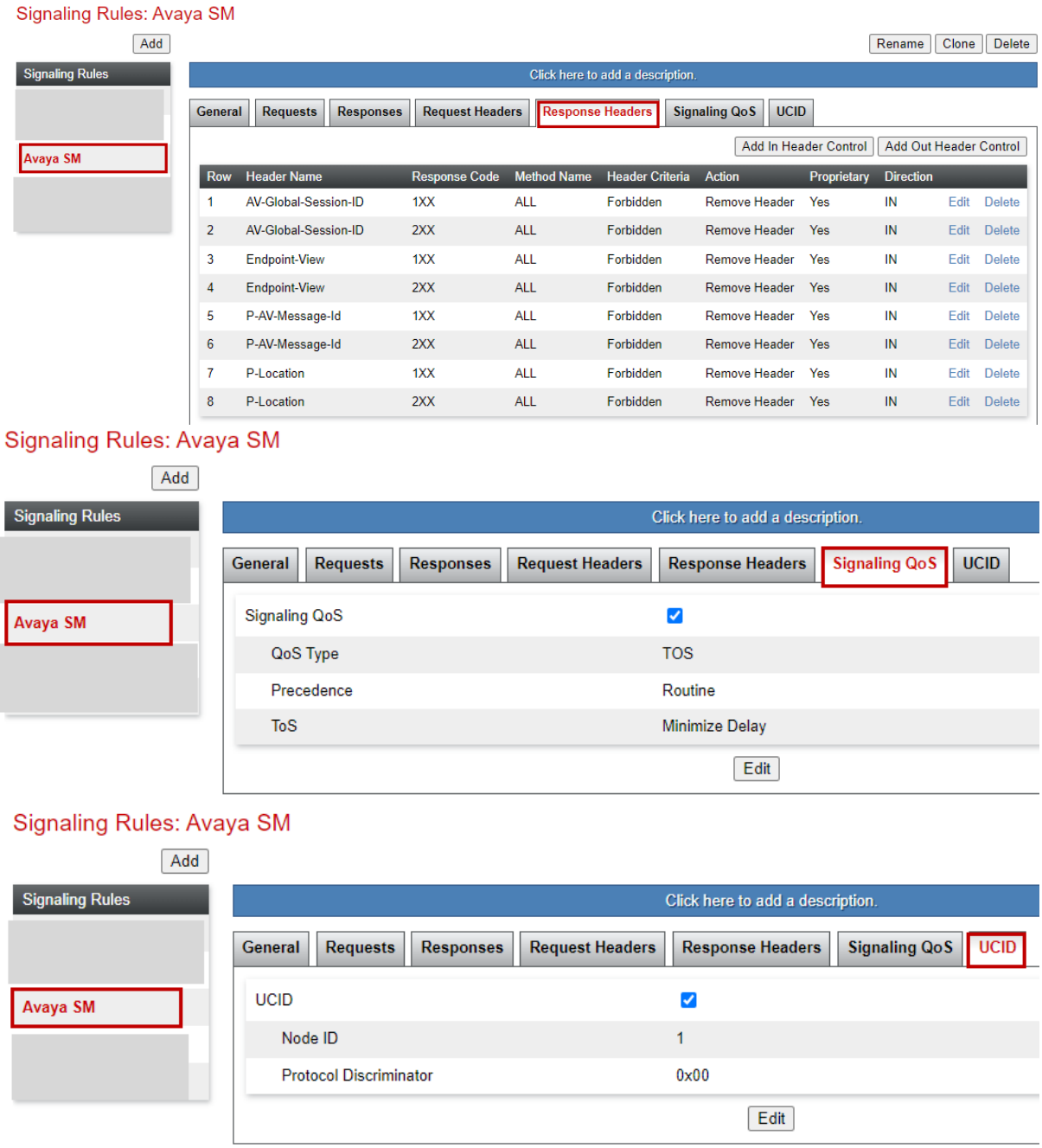



*Figure 67 Signaling Rules for Avaya SM Continuation*

### 1.1.8 End Point Policy Groups

**End Point Policy Group for Twilio**

- Create End Point Policy Groups named **Twilio**
- Set *Media Rule*: **Twilio** (Created in Section 6.2.6)

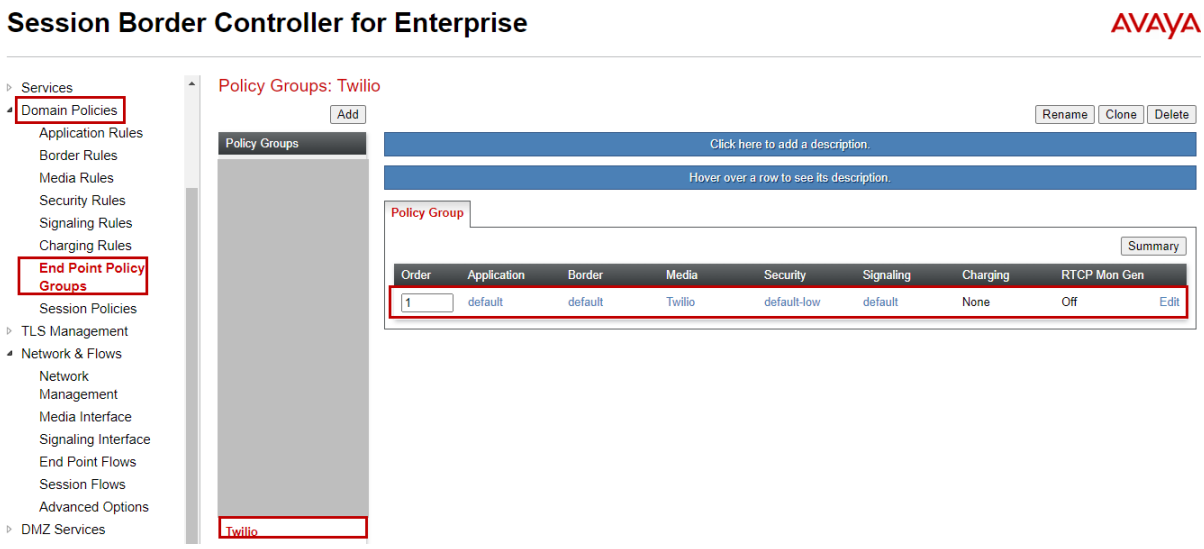



*Figure 68 End Point Policy Group for Twilio*

**End Point Policy Group for Avaya SM**

- Create End Point Policy Groups named **Avaya SM**
- Set *Media Rule*: **AvayaSM** (Created in Section 6.2.6)
- Set *Signaling Rule*: **Avaya SM** (Created in Section 6.2.7)


*Figure 69 End Point Policy Group for Avaya SM*

### 1.1.9 Network Management

**Network Interface for Avaya SM**

- Navigate to **Network & Flows > Network Management**
- Create Network Interface for Avaya SM named **LAN**
- Set *Default Gateway*: Default Gateway of **Avaya SM IP address**
- Set *Network Prefix or Subnet Mask*: Enter the **Subnet Mask**
- *Interface*: Select **A1**
- *IP Address*: Enter IP address of **Avaya SM**

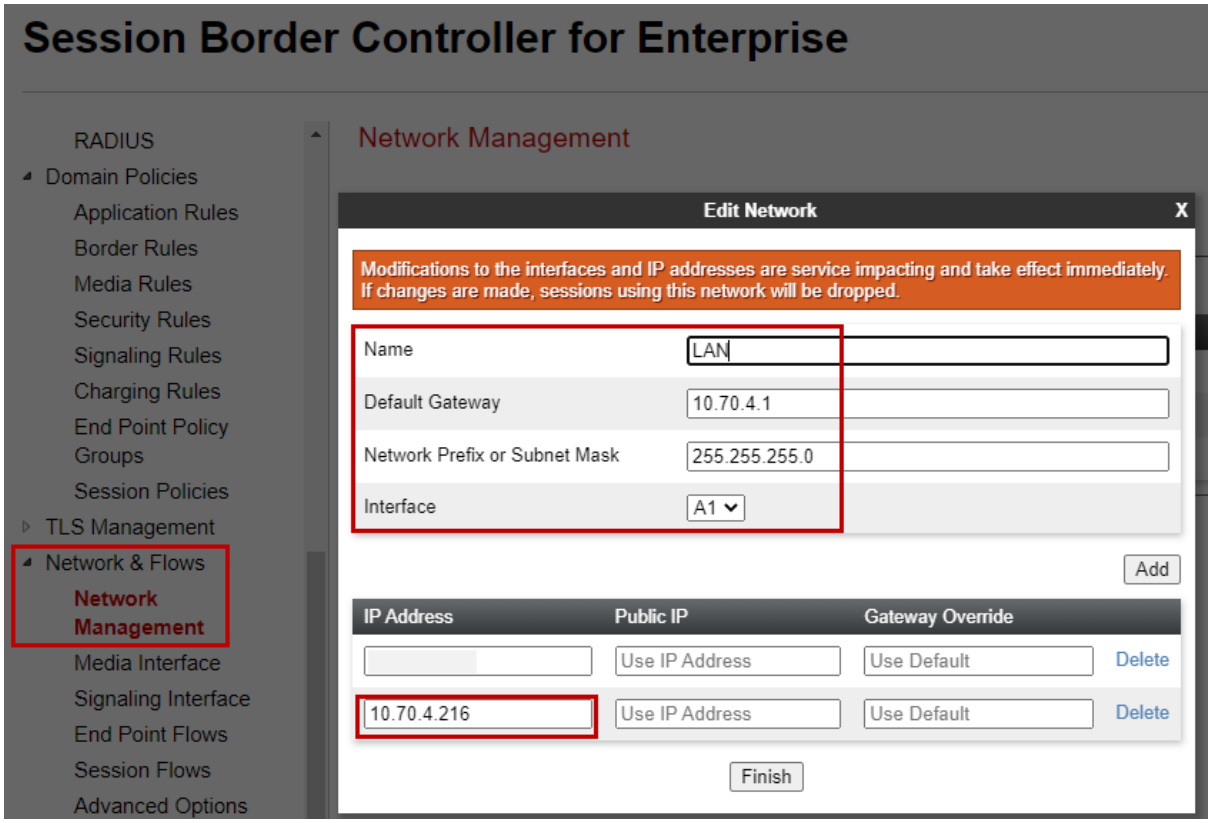



*Figure 70 Network Interface for Avaya SM*

**Network Interface for Twilio**

- Create Network Interface for Twilio named **WAN**
- Set *Default Gateway*: Default Gateway for **Twilio** interface
- Set *Network Prefix or Subnet Mask*: Enter the **Subnet Mask**
- *Interface*: Select **B1**
- *IP Address*: Enter interface IP address for **Twilio**

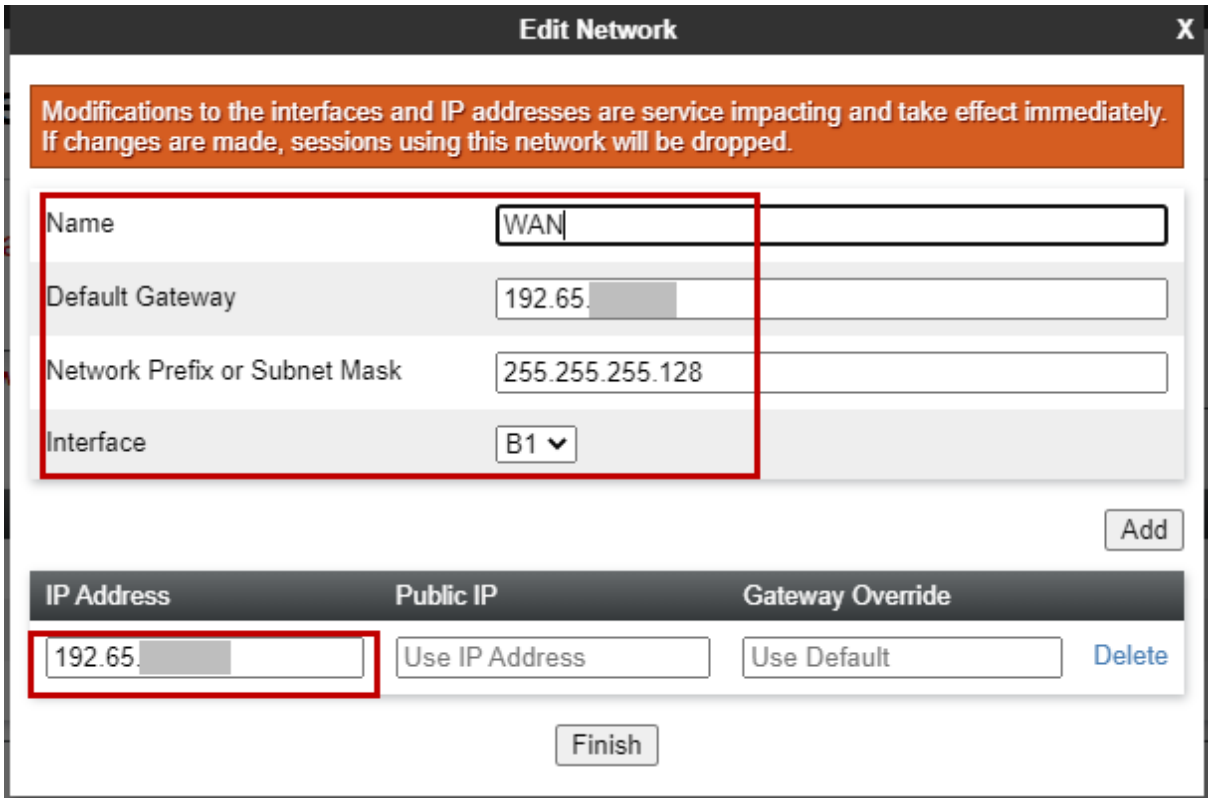



*Figure 71 Network Interface for Twilio*

### 1.1.10 Media Interface

**Media Interface facing Twilio**

- Navigate to **Network & Flows > Media Interface**
- Create **Media Interface** named **MI_WAN**
- Set *IP Address*: Select **WAN (B1, VLAN 0)** from the drop down and the **IP address** populates automatically. The IP address for Interface facing Twilio is 192.65.X.X

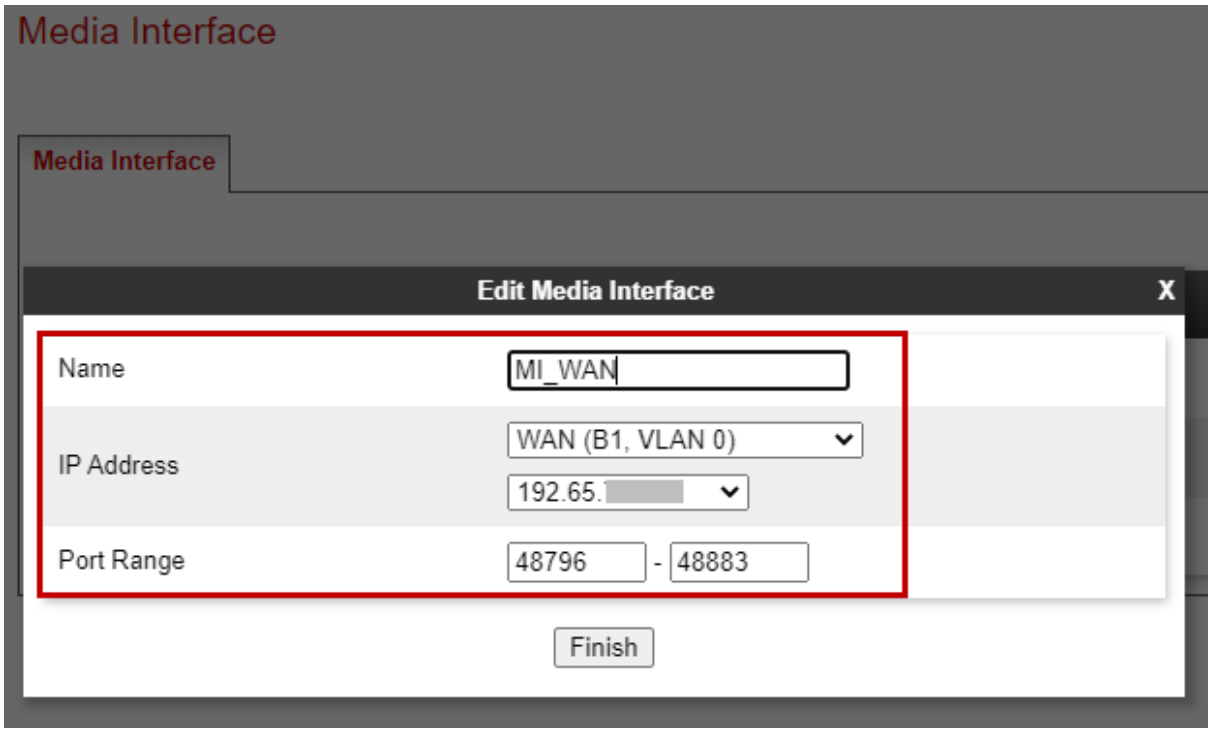



*Figure 72 Media Interface for Twilio*

**Media Interface facing Avaya SM**

- Create **Media Interface** named **MI_LAN**
- Set *IP Address*: Select **LAN (A1, VLAN 0)** from the drop down and the **IP address** populates automatically. The IP address for Interface facing Avaya SM is 10.70.4.216

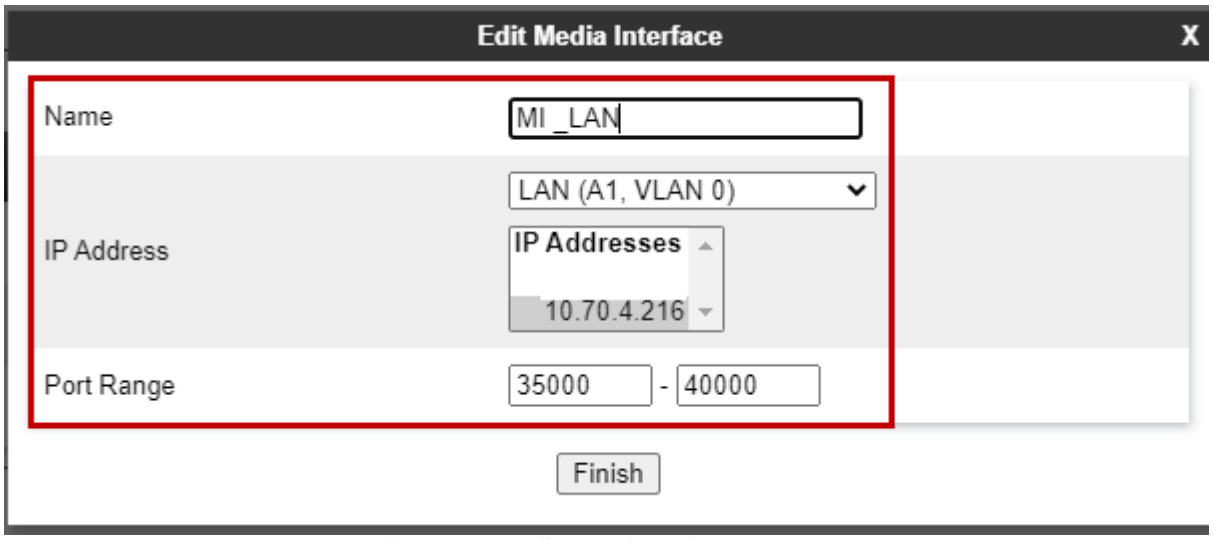



*Figure 73 Media Interface for Avaya SM*

### 1.1.11 Signaling Interface

**Signaling Interface for Avaya SM**

- Navigate to: **Network & Flows > Signaling Interface**
- Set *Name*: **SI_LAN** is given for the interface facing Avaya Aura SM
- Set *IP Address*: Select **LAN (A1, VLAN 0)**
- Set *UDP Port*: **5060**

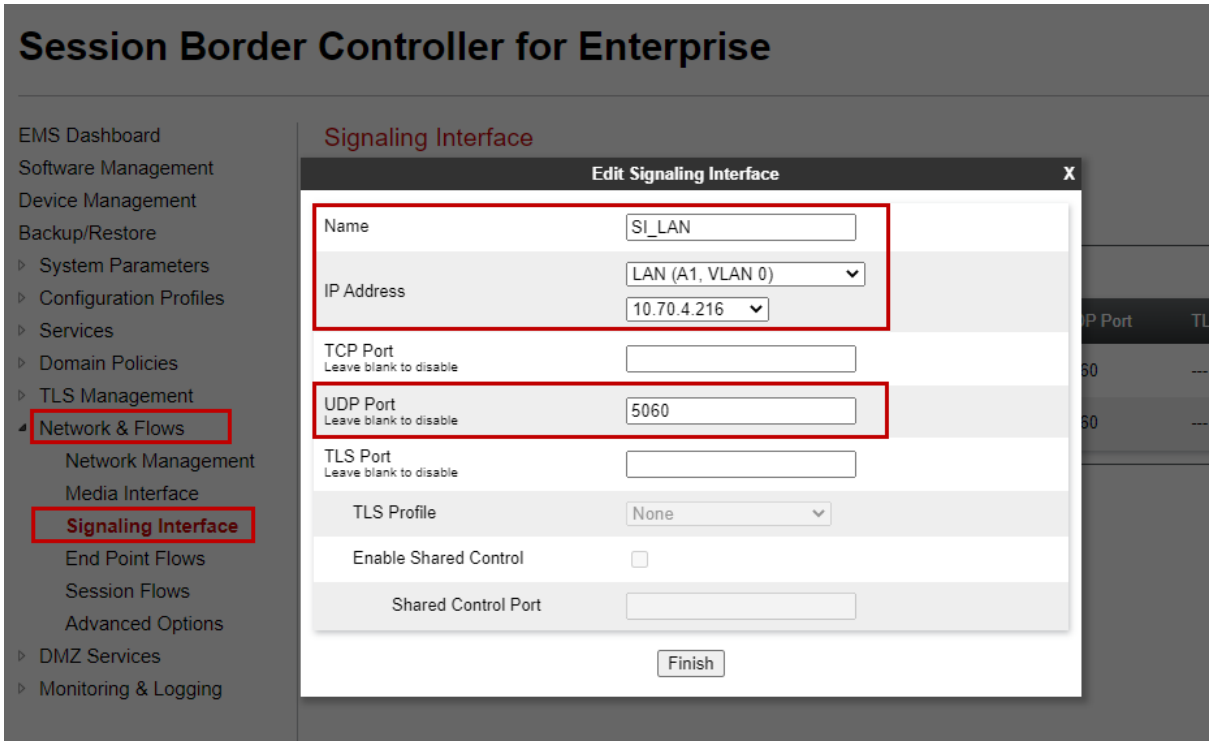



*Figure 74 Signaling Interface for Avaya SM*

### Signaling Interface for Twilio

- Navigate to: **Network & Flows > Signaling Interface**
- Set *Name*: **SI_WAN** is given for the interface facing Avaya SM
- Set *IP Address*: Select **WAN (B1, VLAN 0)**
- Set *UDP Port*: **5060**

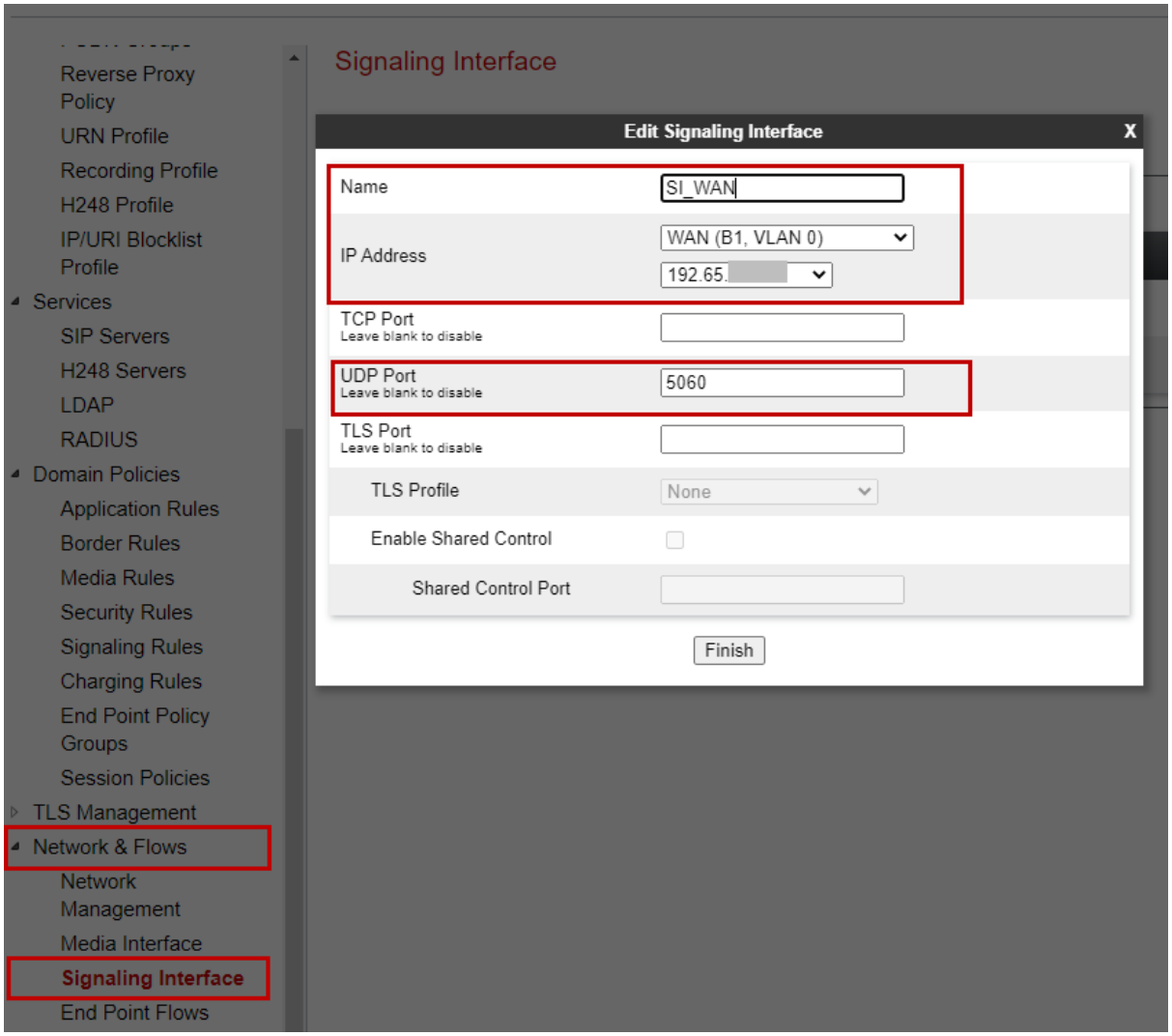



*Figure 75 Signaling Interface for Twilio*

### 1.1.12 End Point Flows

- Navigate to **Network & Flows > End Point Flows > Server Flows**
- Below are the Server Flows created for Avaya SM and Twilio

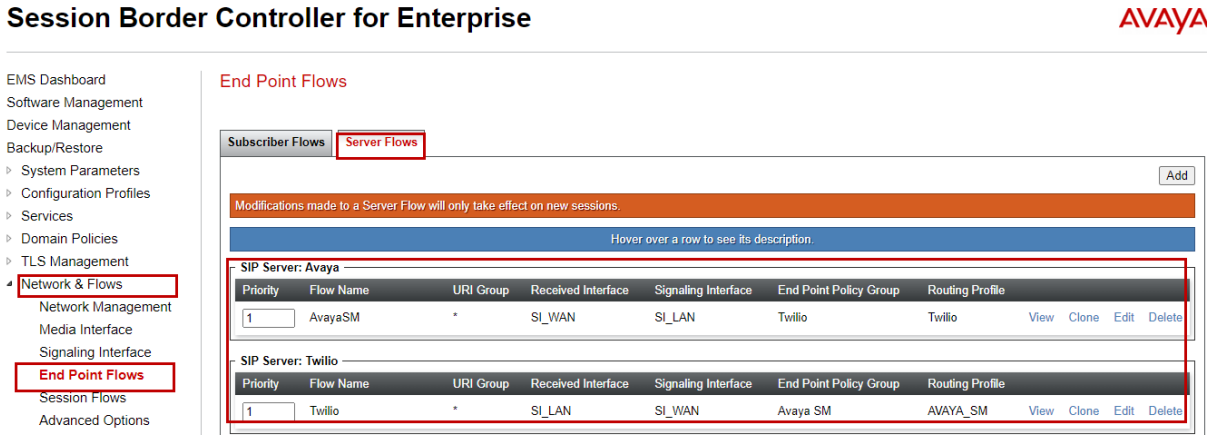



*Figure 76 End Point Flows – Server Flows*

**Server Flow for Avaya SM**

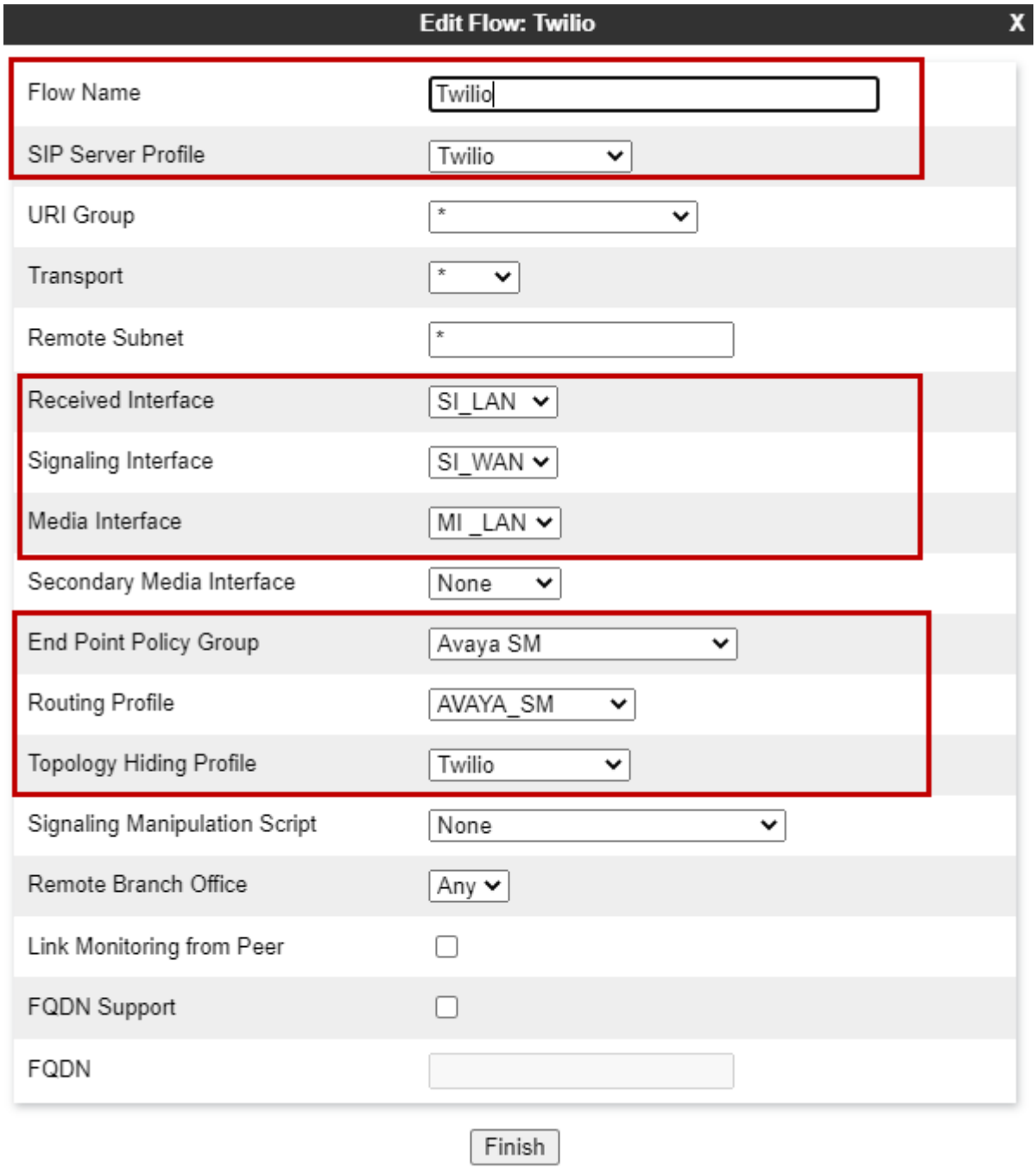



*Figure 77 End Point Flow for Avaya SM*

**Server Flow for Twilio**

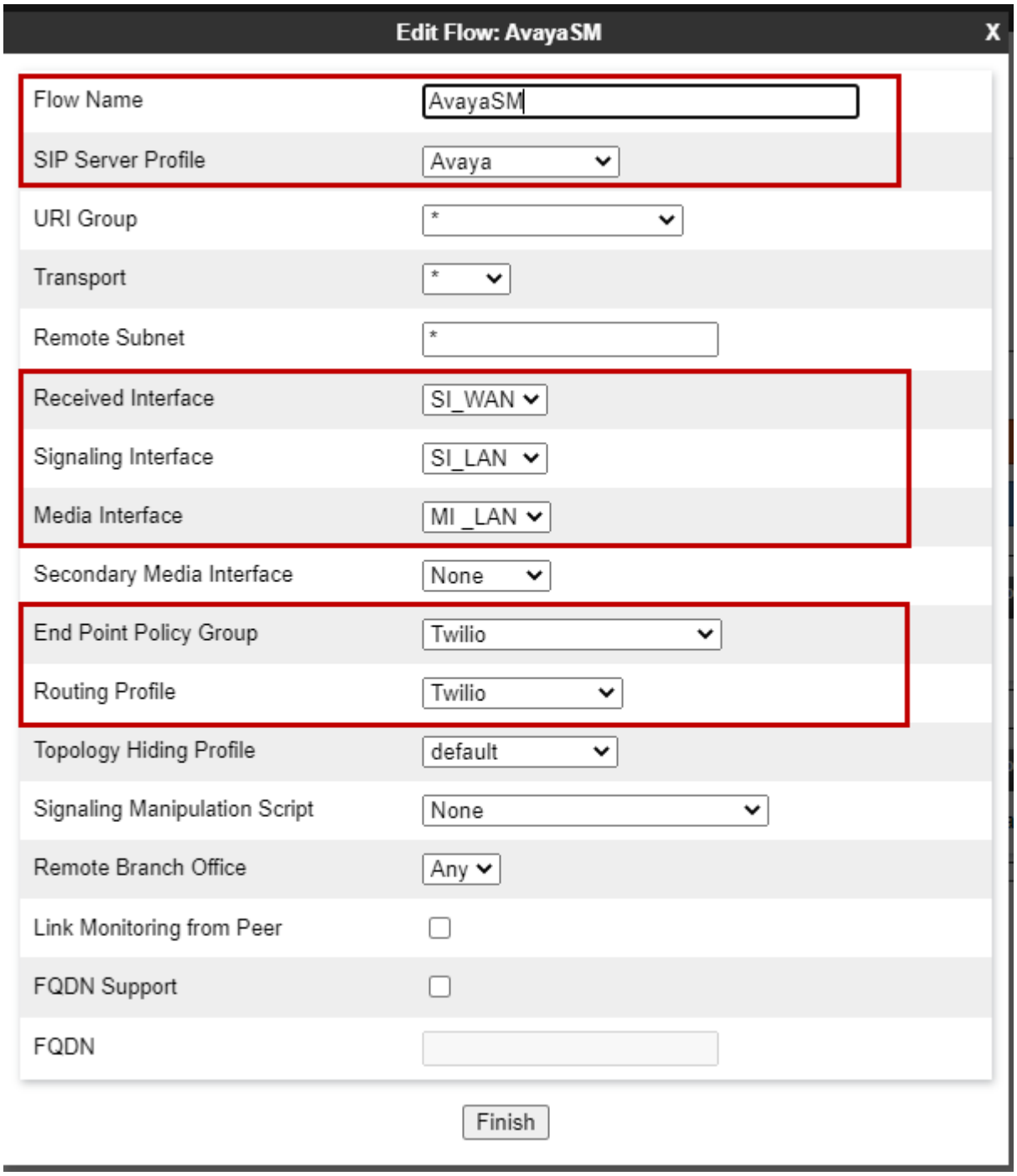



*Figure 78 End Point Flow for Twilio*

### 1.1.13 Signaling Manipulation

Signaling Manipulation script **AvayaSM** is created towards Avaya SM. This script assigns the X-parentCall value to User-to-User header and sent towards Avaya SM

Navigate to **Configuration Profiles > Signaling Manipulation**

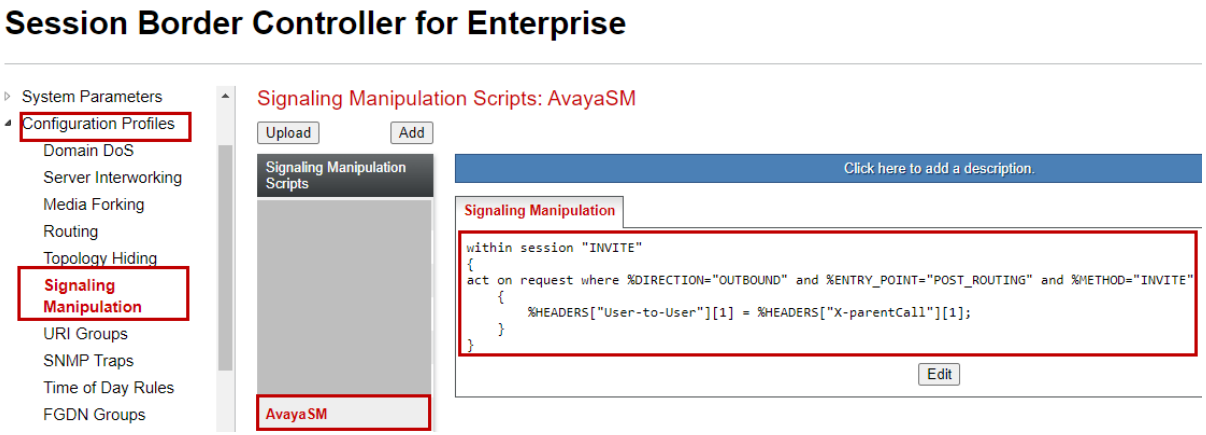



*Figure 79 Sigma script towards Avaya SM*

## 1.2 Avaya Aura Session Manager Configuration

### 1.2.1 Avaya Aura SM Login

- Avaya Aura SM Configuration is accomplished through the Avaya Aura System Manager.
- Access Avaya Aura System Manager Web login screen via **https://<IP Address/FQDN>**
- Use admin as **User ID** and associated **Password**
- Click **Log On**

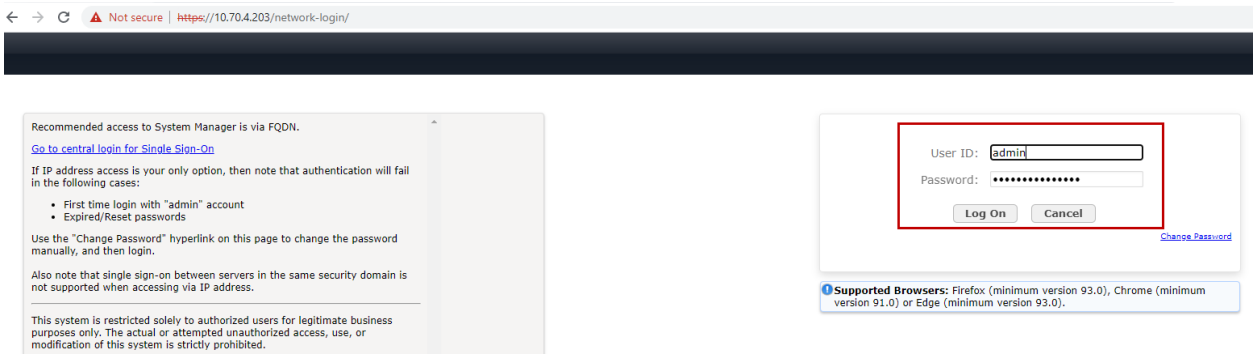



*Figure 80 Avaya SM Login*

### 1.2.2 Domain

- Navigate to **Elements > Routing**

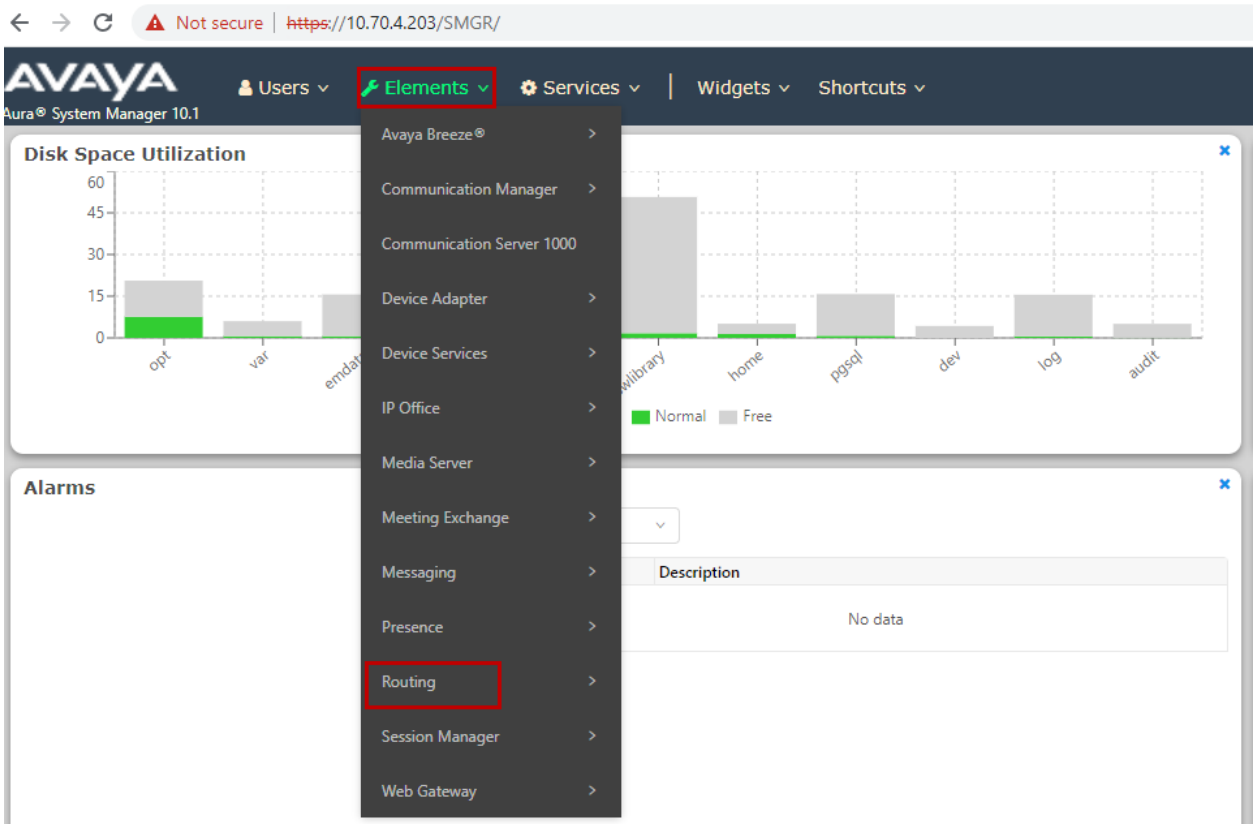



*Figure 81 Routing*

- Navigate to **Routing > Domains**
- Click **New**

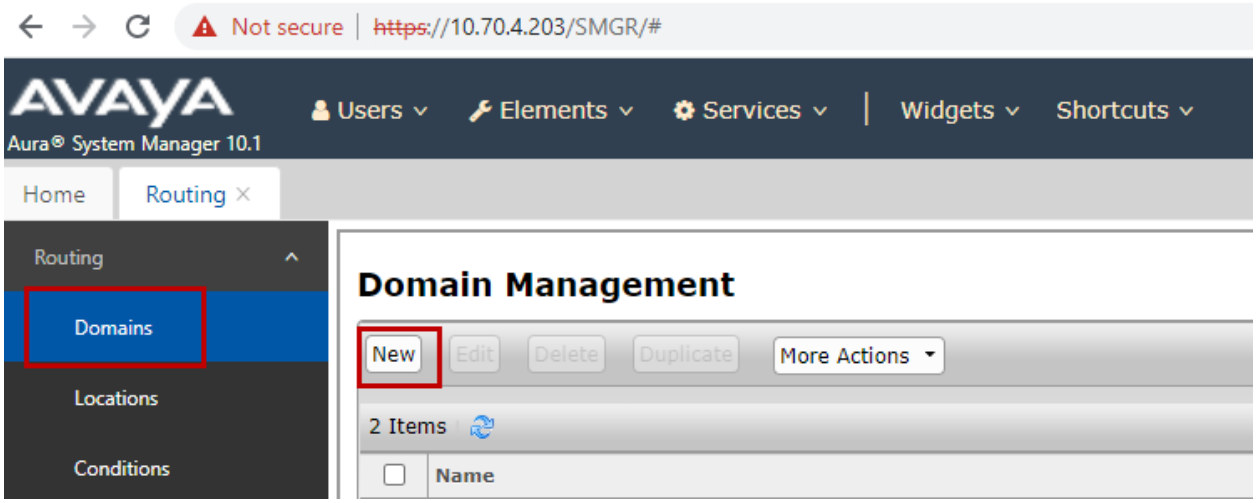



*Figure 82 Add Domain*

- Set *Name*: Enter the domain name of Avaya Aura SM, **xxxx.com**
- Set *Type*: **sip**

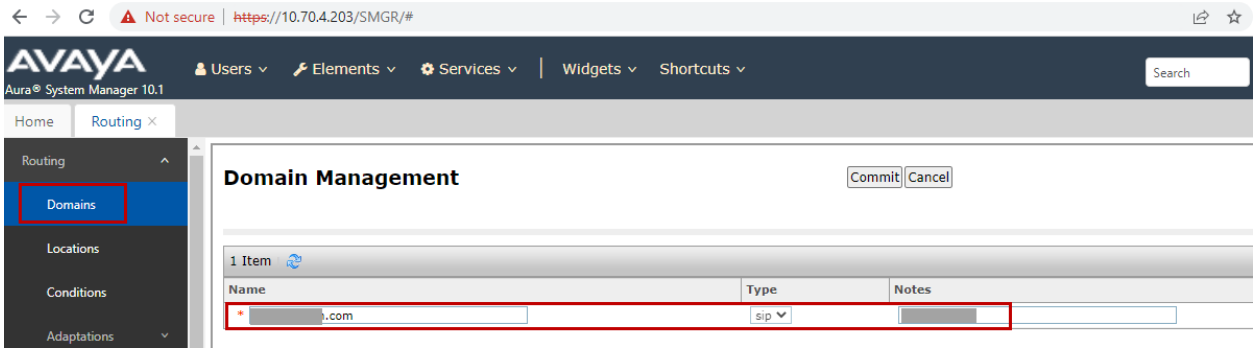



*Figure 83 Domain*

### 1.2.3 Locations

- Navigate to **Routing > Locations**
- Select **New**

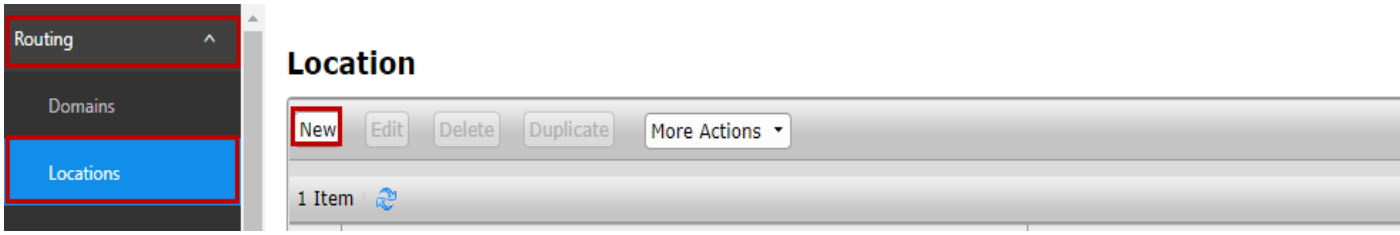



*Figure 84 Locations*

- Set *Name*: **Plano**

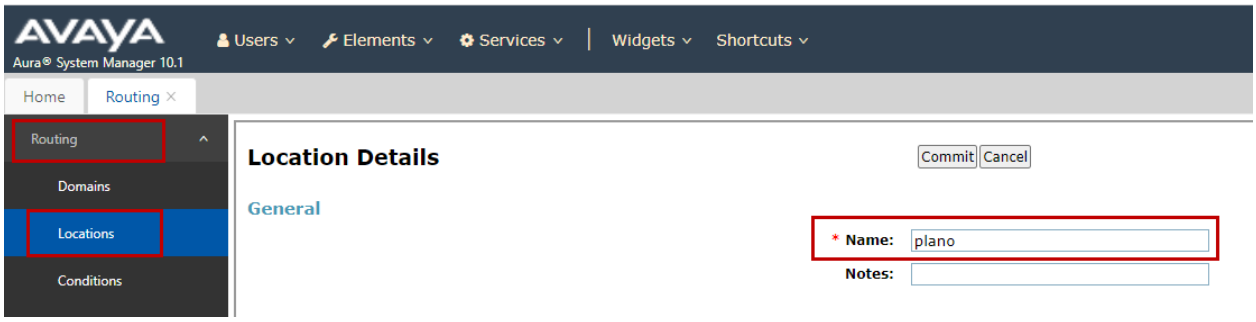



*Figure 85 Locations Continuation*

- Under *Location Pattern*, select **Add** to add **IP Address** Patterns for different networks that communicates within the location
- Set *IP Address Pattern*: **10.80.33.x**
- Leave all other fields to default values
- Click **Commit**

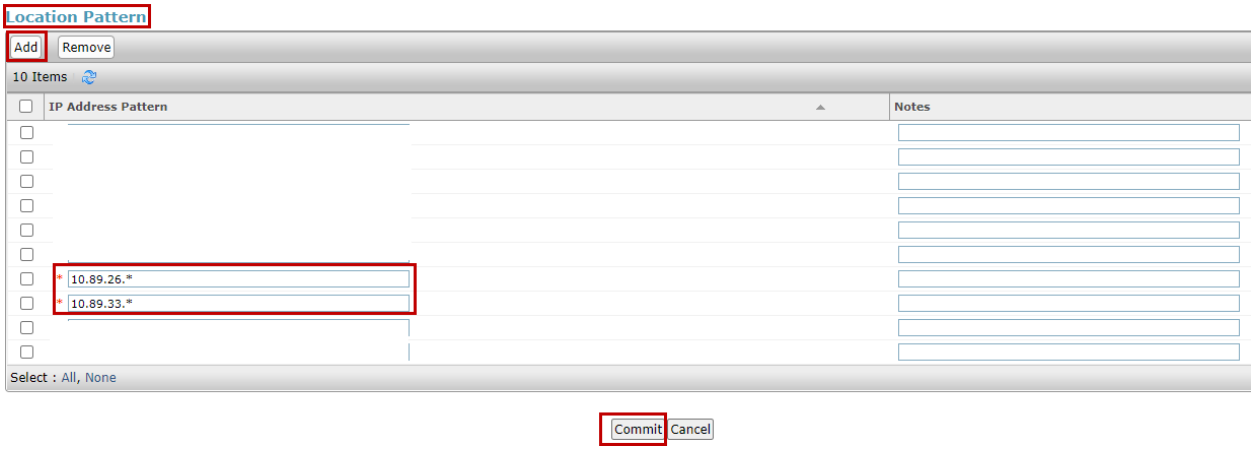



*Figure 86 Locations continuation*

### 1.2.4 Adaptations

**Adaptation for Avaya Aura Contact Center**

- Navigate to **Routing > Adaptations**. Click **New**
- Set *Adaptation Name*: **Adaptation for AvayaCC**
- Set *Module Name*: **DigitConversionAdapter**
- Set *Module Parameter Type*: **Name-Value Parameter** is selected from the drop down, Click **Add**
- Set *Name/Value*: **fromto/true**
- Set *Name/Value*: **osrcd/10.70.4.207** (Avaya Aura SM IP is entered)
- Set *Name/Value*: **odstd/xxxx.com** (Avaya Aura Contact Center Domain name is entered)
- Under **Digit Conversion for Outgoing Calls from SM**, add an entry to delete the incoming number pattern and send/insert CDN number (Pilot number e.g. 7500) of the Avaya Aura Contact Center in the destination address
- Click **Commit**

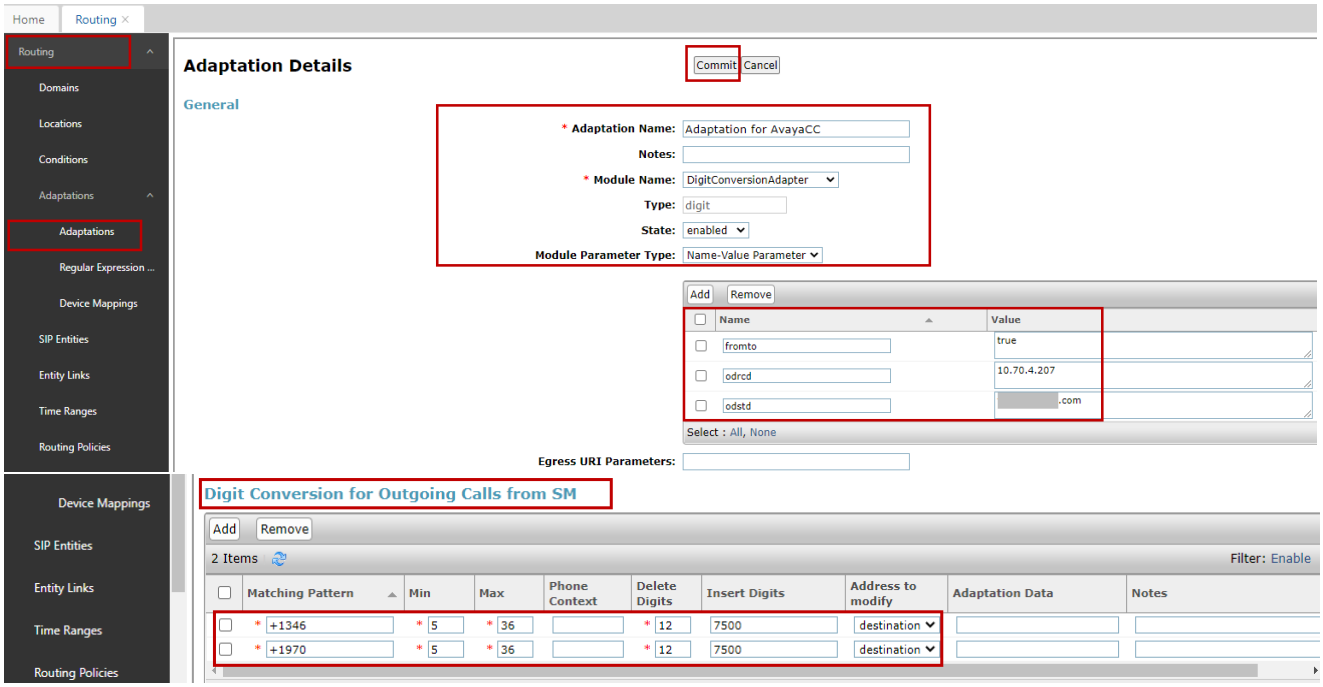



*Figure 87 Adaptation for Avaya Aura Contact Center*

### Regular Expression for Avaya Aura Contact Center

This regular expression towards Avaya Aura Contact Center (CC) copies the User-to-User header value (i.e. X-parentCall value) received from Avaya SBCE and assigns it to User-Agent header in the Avaya Aura SM

Avaya Aura SM sends User-Agent header with X-parentCall value towards Avaya Aura CC

- Navigate to **Routing > Adaptations > Regular Expression**. Click **New**
- Set Name: **test**

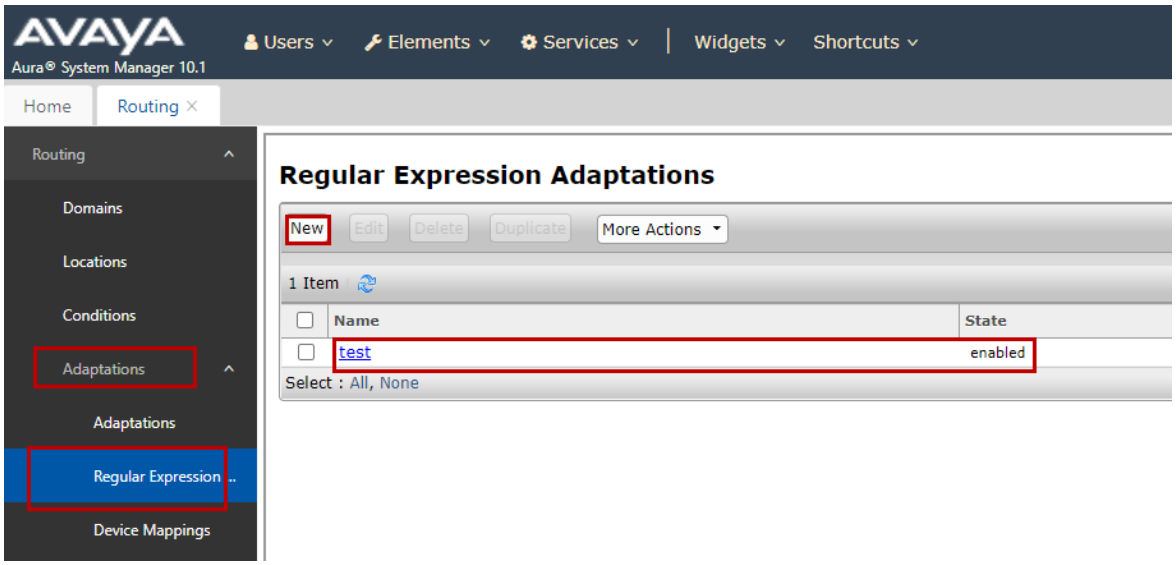



*Figure 88 Regular Expression for Avaya Aura Contact Center*

- Navigate to **Outgoing Adaptation Rules**. Click **Add**
- Create *Rule Name*: **removeuseragent**

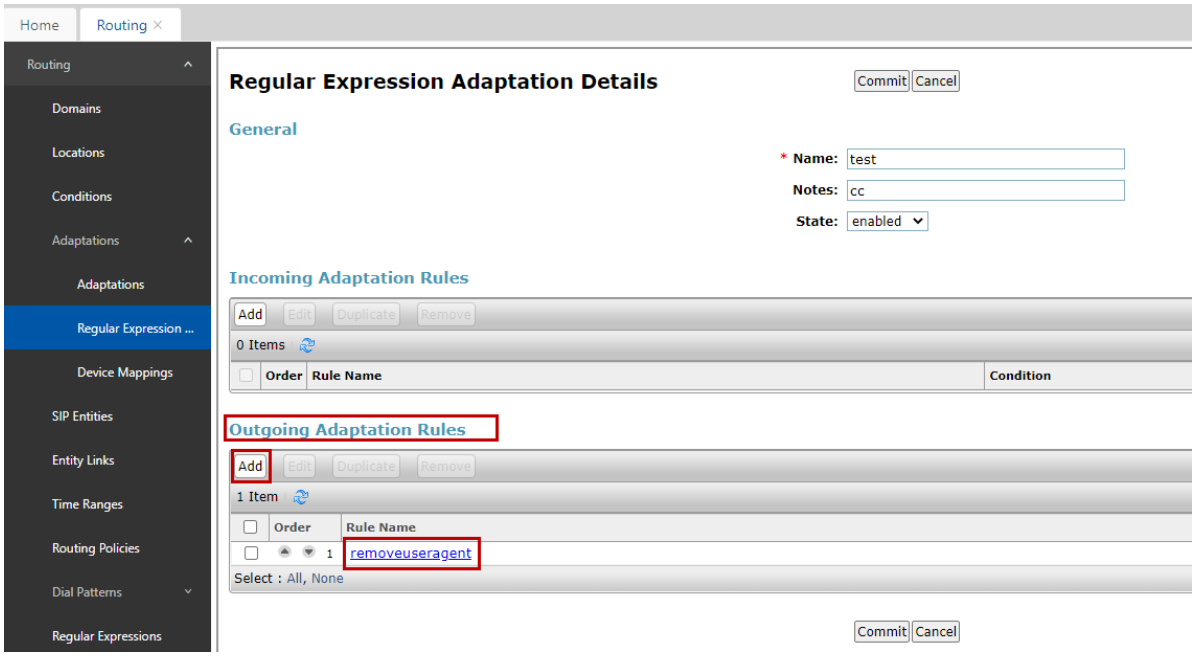



*Figure 89 Regular Expression for Avaya Aura Contact Center Continuation*

- Under **Regular Expression Adaptation Rule Details**,
  - *Rule Name*: **removeuseragent**
  - *Direction*: **Outgoing**
- Navigate to **Rule Variables**. Click **Add**
  - Set *Variable Name*: **AvayaSM**
  - Set *Source Type*: **Header**
  - Set *Source*: **User-to-User**
  - Set *Instance*: **top**
  - Set *Match Expression*: **.***
- Navigate to **Rule Actions**. Click **Add**
  - Set *Source Type*: **Header**
  - Set *Source*: **User-Agent**
  - Set *Instance*: **any**
  - Set *Operation*: **modify**
  - Set *Match Expression***: .***
  - Set *Replace/Add Expression*: **${AvayaSM}**

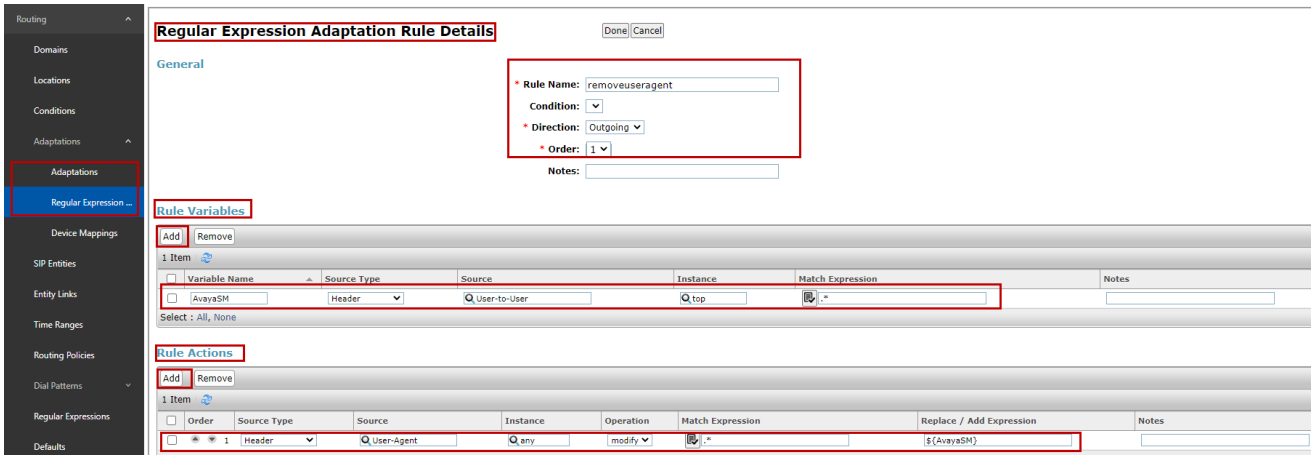



*Figure 90 Regular Expression for Avaya Aura Contact Center Continuation*

### Adaptation for Avaya Aura Communication Manager (CM)

This adaptation changes the host part of FROM and TO header with Avaya Aura CM domain name

- Set *Adaptation Name*: **Adaptation_ For_cm**
- Set *Module Name*: **DigitConversionAdapter**
- Set *Module Parameter Type*: **Name-Value Parameter** is selected from the drop down, Click **Add**
- Set *Name/Value*: **fromto/true**.
- Set *Name/Value*: **odstd/xxxx.com** (Avaya Aura CM Domain name is entered)
- Set *Name/Value*: **osrcd/xxxx.com** (Avaya Aura CM Domain name is entered)

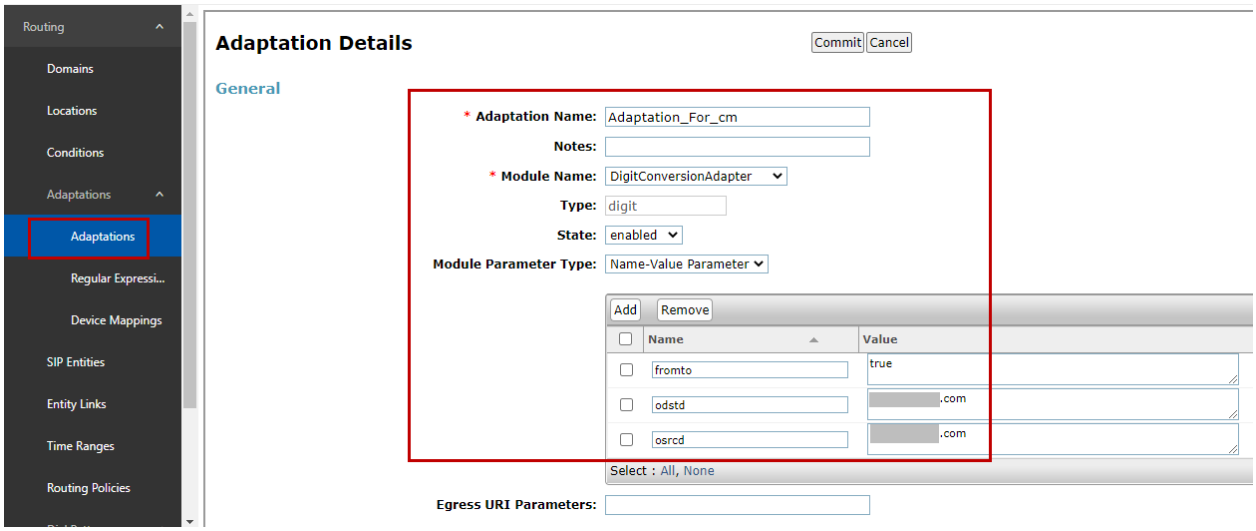



*Figure 91 Adaptation for Avaya Aura CM*

### 1.2.5 SIP Entities and Entity Links

**SIP Entity for Avaya Aura Session Manager**

- Navigate to: **Routing > SIP Entities**
- Click **New**
- Set *Name*: Enter name of the host, **AASM10**
- Set *FQDN or IP Address*: Enter the **SIP address** of the **Session Manager**
- Set *Type*: **Session Manager** is selected from the drop down
- Set *Location*: Select the **location** (Created in Section 6.3.3)
- Under *Listen Ports*
    - Set *TCP/TLS Failover Port*: **5060/5061**
- Click **Add** to assign Domain **xxxxx.com** (Default Domain name)for the following Ports and Protocols
- Port **5060/5062** and Protocol **TCP/UDP**
- Click **Commit**

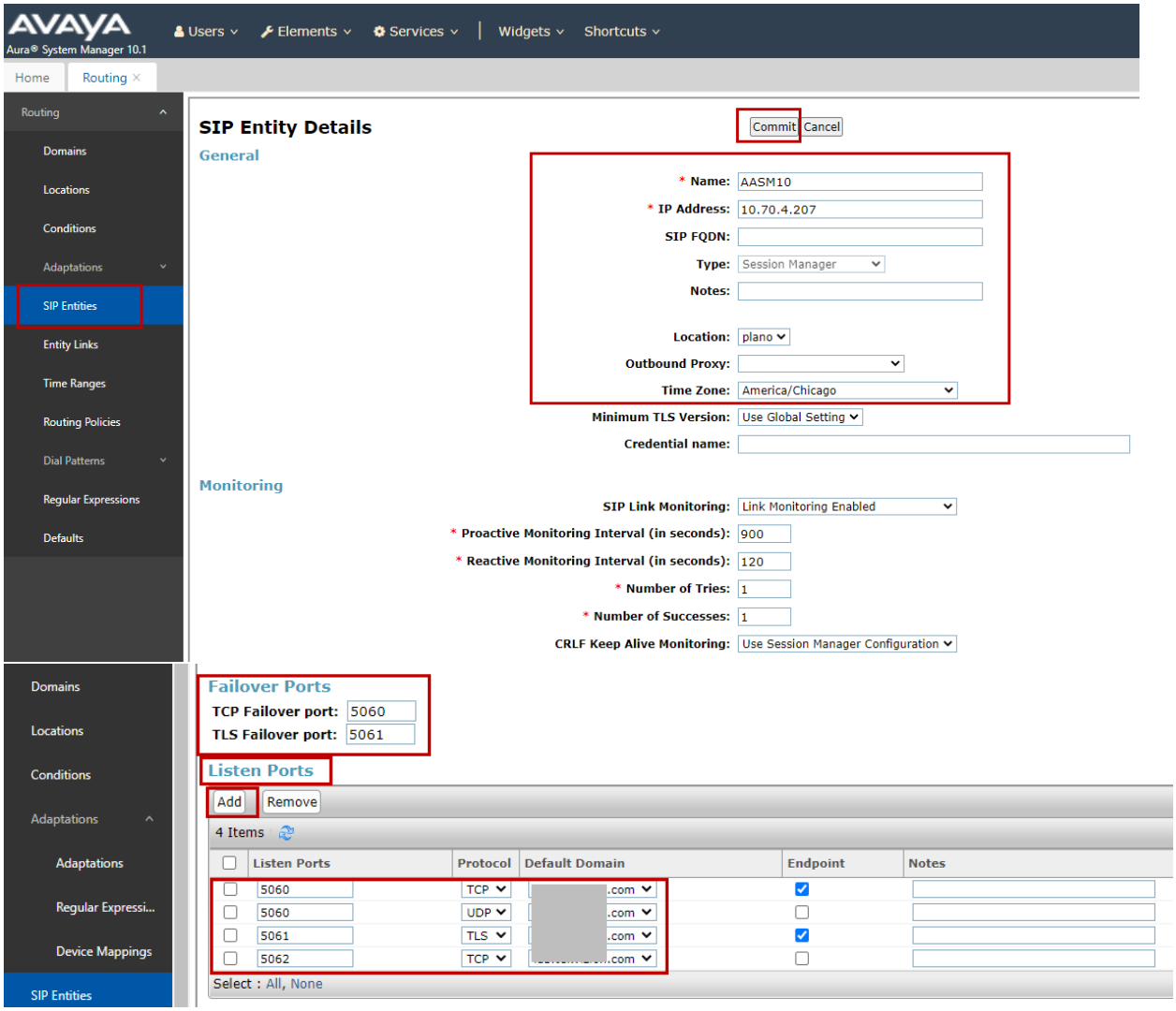



*Figure 92 SIP Entity for Avaya SM*

**SIP Entity and Entity Links for Avaya Aura Communication Manager**

- Set *Name*: **AACM10**
- Set *FQDN or IP Address*: Enter the **IP address** of **Avaya Aura Communication Manager**
- Set *Type*: **CM**
- Set *Location*: Select the **location** (Created in Section 6.3.3)

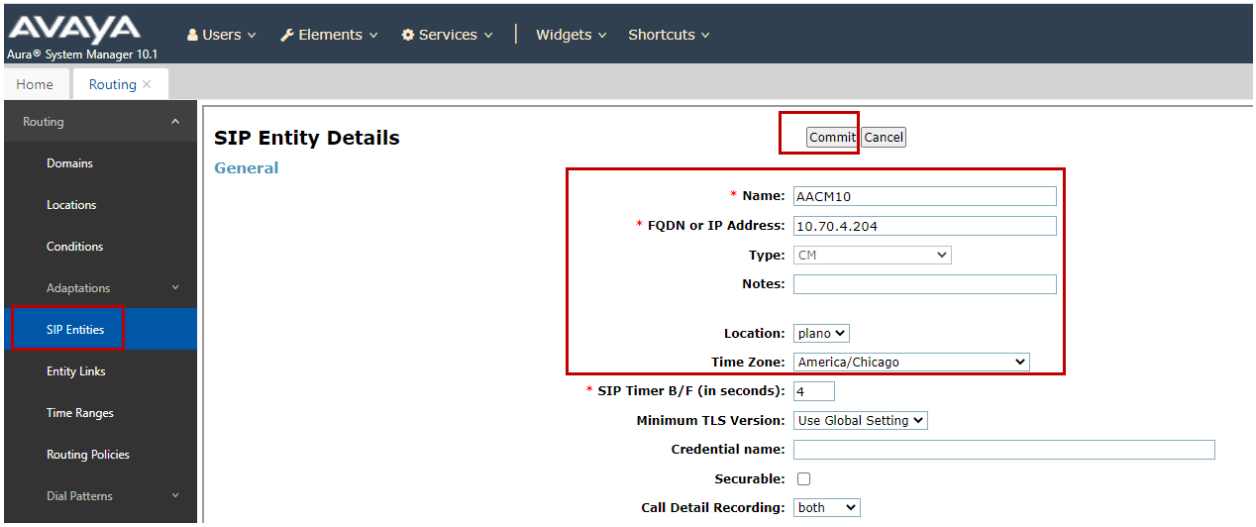



*Figure 93 SIP Entity and Entity Link for Avaya CM*

- Set *Adaptation*: Select the **Adaptation** for Avaya Aura CM configured in Section 6.3.4
- Under **Entity Links**, Click **Add**
    - Set *Name*: **SM10_CM_SIP Trunk**
    - Set *SIP Entity 1*: Select the SIP entity **AASM10**
    - Set *SIP Entity 2*: **AACM10**
    - Set *Protocol*: **TCP**
    - Set *Port*: **5060**
    - Set *Connection Policy*: **trusted**

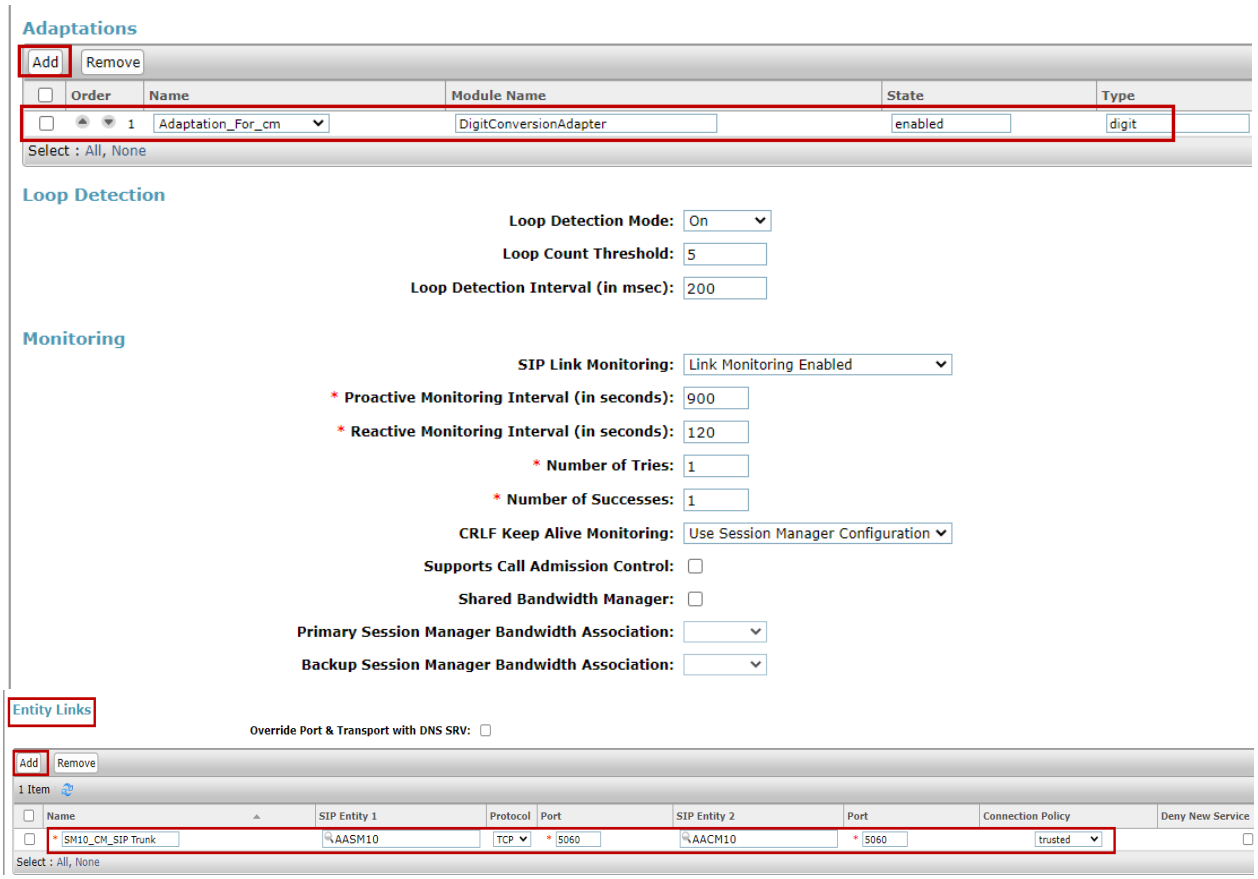



*Figure 94 SIP Entity and Entity Link for Avaya CM Continuation*

**SIP Entity and Entity Links for Twilio**

- Set *Name*: **Twilio**
- Set *FQDN or IP Address*: Enter the **IP address** of **Avaya SBCE LAN IP**
- Set *Type*: **SIP Trunk**
- Set *Location*: Select the **location** (Created in Section 6.3.3)

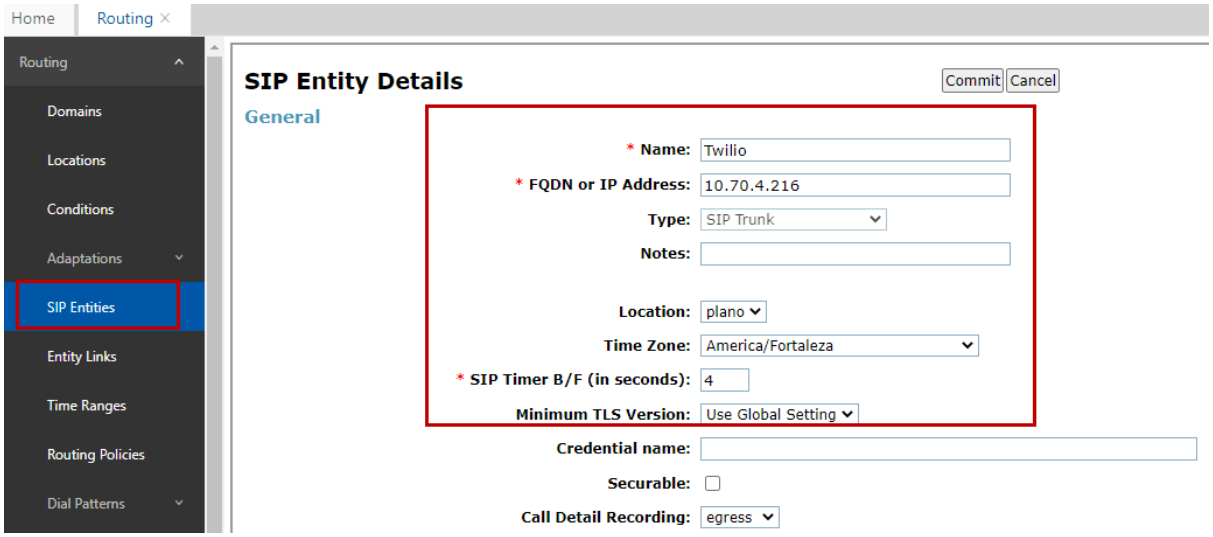



*Figure 95 SIP Entity and Entity Link for Twilio*

- Under **Entity Links**, Click **Add**
  - Set *Name*: **SMtoTwilio**
  - Set *SIP Entity 1*: Select the SIP entity **AASM10**
  - Set *SIP Entity 2*: **Twilio**
  - Set *Protocol*: **UDP**
  - Set *Port*: **5060**
  - Set *Connection Policy*: **trusted**

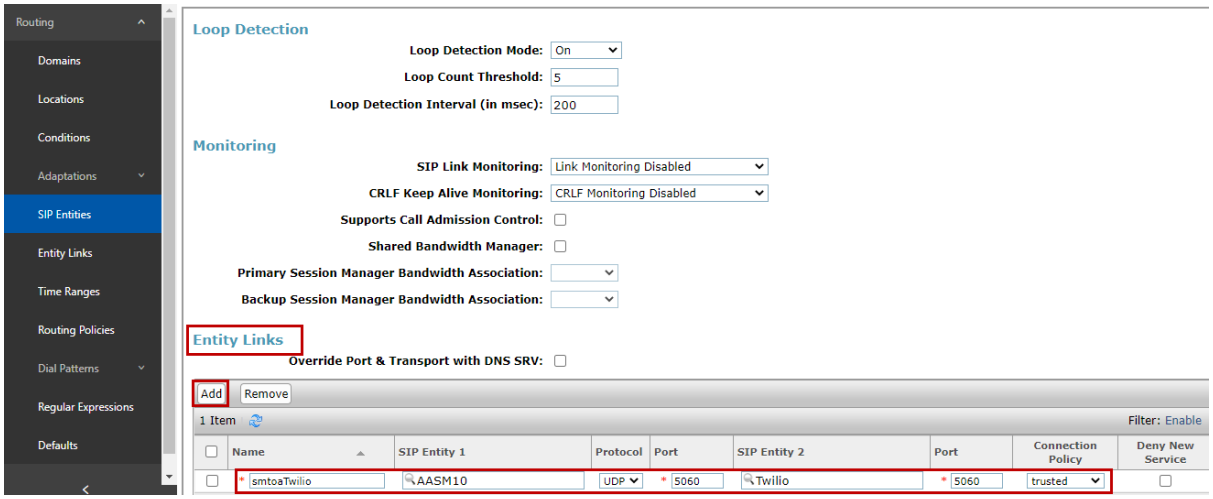



*Figure 96 SIP Entity and Entity Link for Twilio Continuation*

**SIP Entity and Entity Links for Avaya Aura CC**

- Set *Name*: **Avaya CC Manager Server**
- Set *FQDN or IP Address*: Enter the **IP address** of **Avaya Aura Contact Center IP**
- Set *Type*: **Other**
- Set *Location*: Select the **location** (Created in Section 6.3.3)
- *Adaptations*: Select the **Adaptation** and the **Regular Expression** for Avaya Aura CC configured in Section 6.3.4
- Under **Entity Links**, Click **Add**
    - Set *Name*: **SMtoAvayaCCMS**
    - Set *SIP Entity 1*: Select the SIP entity **AASM10**
    - Set *Port*: **5062**
    - Set *SIP Entity 2*: **Avaya CC Manager Server**
    - Set *Protocol*: **TCP**
    - Set *Port*: **5060**
    - Set *Connection Policy*: **trusted**

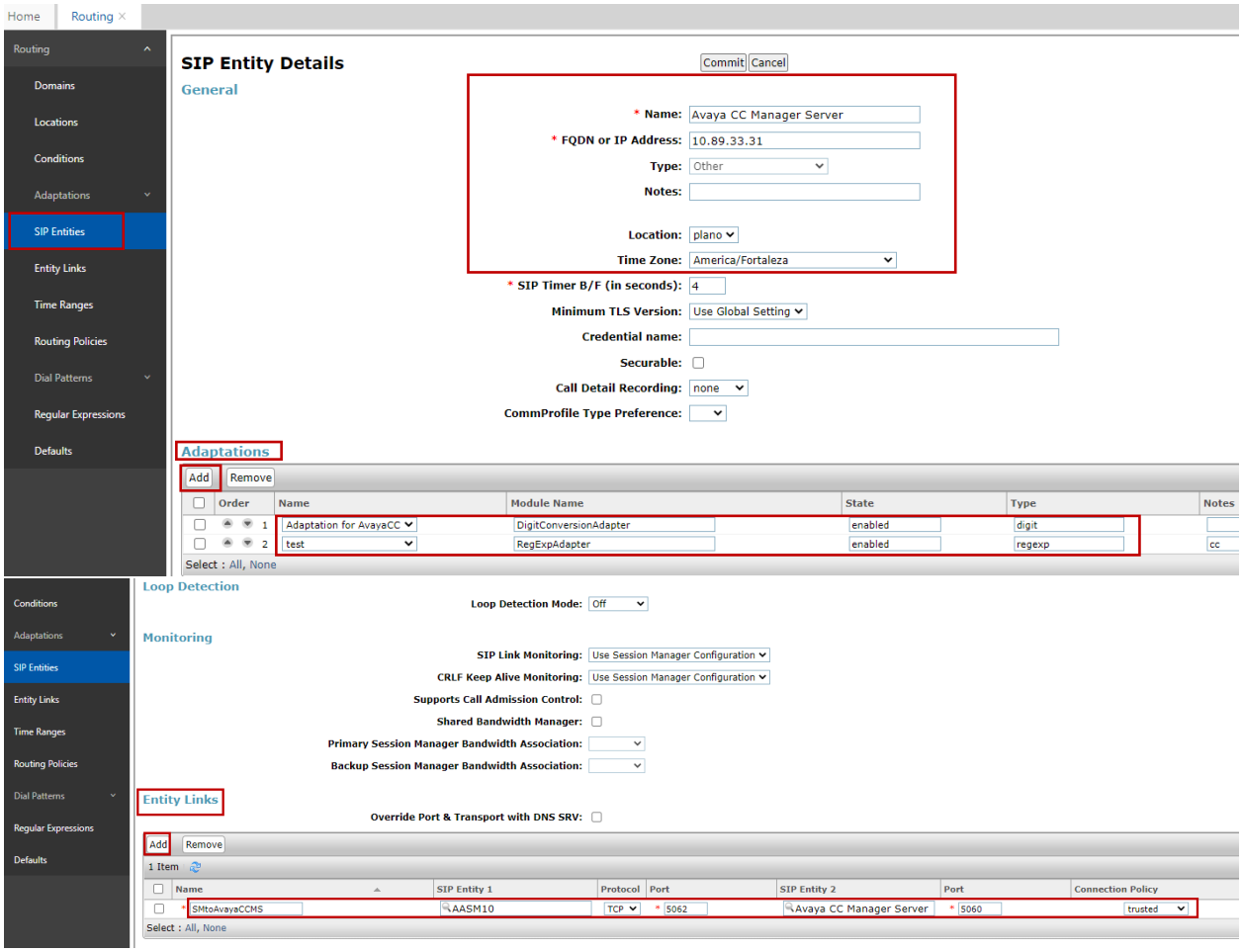



*Figure 97 SIP Entity and Entity Link for Avaya Aura CC*

### 1.2.6 Routing Policies

**Routing policy to Avaya Aura CM**

- Navigate to: **Routing > Routing Policies**
- Click **New**
- Set *Name*: **SM to CM**
- Click **Select** under **SIP Entity as Destination** and the **SIP Entities** window is displayed
- Check the radio button beside **AACM10** as destination SIP Entity (Configured in Section 6.3.5)
- Click **Select** and return back to **Routing Policy Details** page
- Click **Commit**

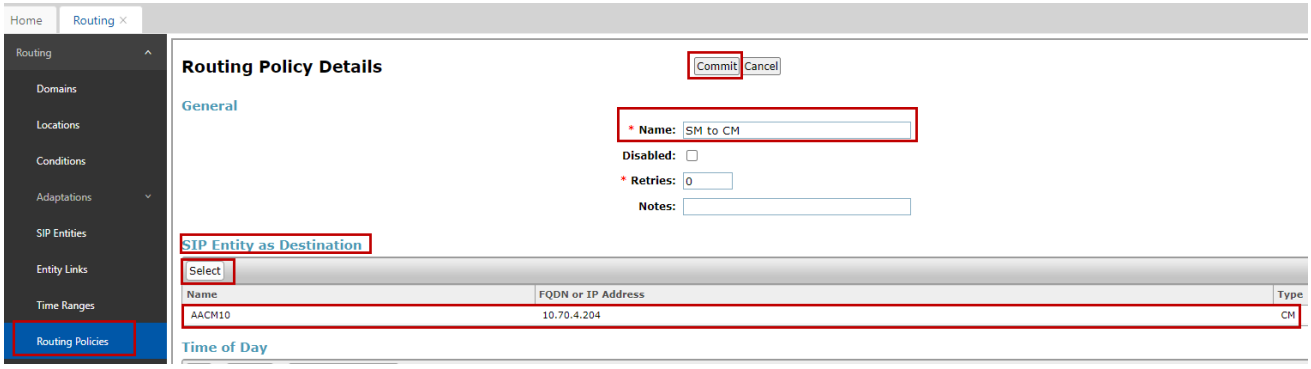



*Figure 98 Routing Policy for Avaya Aura CM*

**Routing policy to Avaya Aura Contact Center (CC)**

- Set *Name*: AvayaCCMS
- Click **Select** under **SIP Entity as Destination** and the **SIP Entities** window is displayed
- Check the radio button beside **Avaya CC Manager Server** as destination SIP Entity (Configured in Section 6.3.5)
- Click **Select** and return back to **Routing Policy Details** page
- Click **Commit**

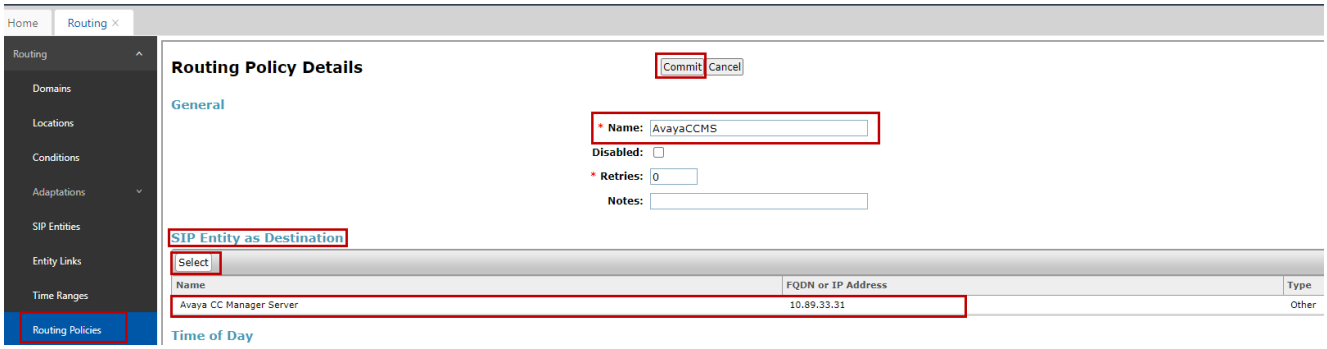



*Figure 99 Routing Policy for Avaya Aura CC*

### 1.2.7 Dial Patterns

**Dial Pattern for Avaya Aura CM**

- Navigate to: **Routing > Dial Patterns**
- Click **New**
- Set *Pattern*: **3001**
- Set *Min*: **4**
- Set *Max*: **4**
- SIP *Domain*: **xxxx.com** (Domain name of Avaya Aura SM)
- Under **Originating Locations** and **Routing Policies**, Click **Add**, at the new window
    - *Originating Location*: Select **Plano** (Created in Section 6.3.3)
    - *Routing Policies*: Select **SM to CM** under Routing Policies
    - Click **Select** to return to **Dial Pattern Details** page
    - Click **Commit**

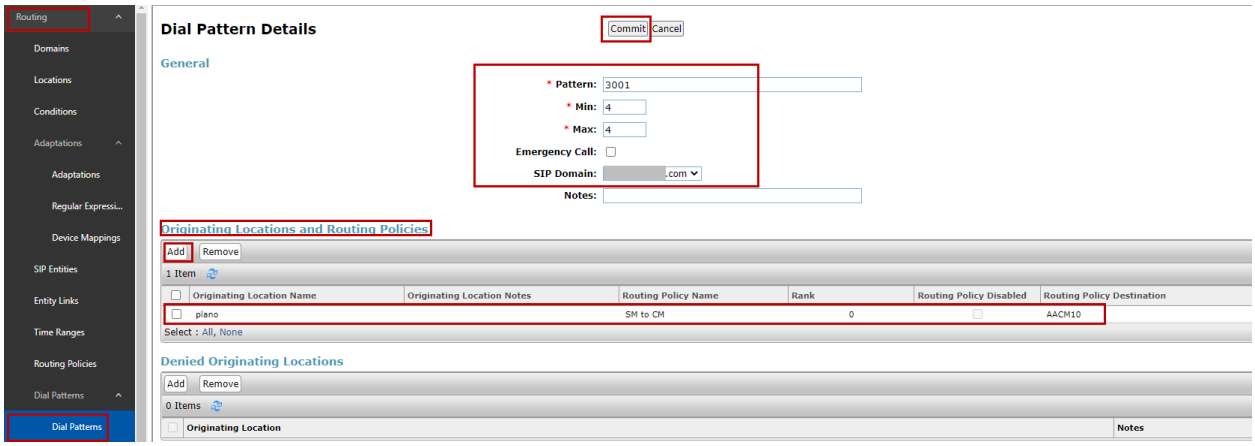



*Figure 100 Dial Pattern for Avaya Aura CM*

### Dial Pattern for Avaya Aura CC

- Navigate to: **Routing > Dial Patterns**
- Click **New**
- Set *Pattern*: **7500**
- Set *Min*: **4**
- Set *Max*: **36**
- SIP *Domain*: **xxxx.com** (Domain name of Avaya Aura SM)
- Under **Originating Locations** and **Routing Policies**, Click **Add**, at the new window
    - *Originating Location:* Select ***Plano*** (Created in Section 6.3.3)
    - *Routing Policies:* Select **AvayaCCMS** under Routing Policies
    - Click **Select** to return to **Dial Pattern Details** page
    - Click **Commit**

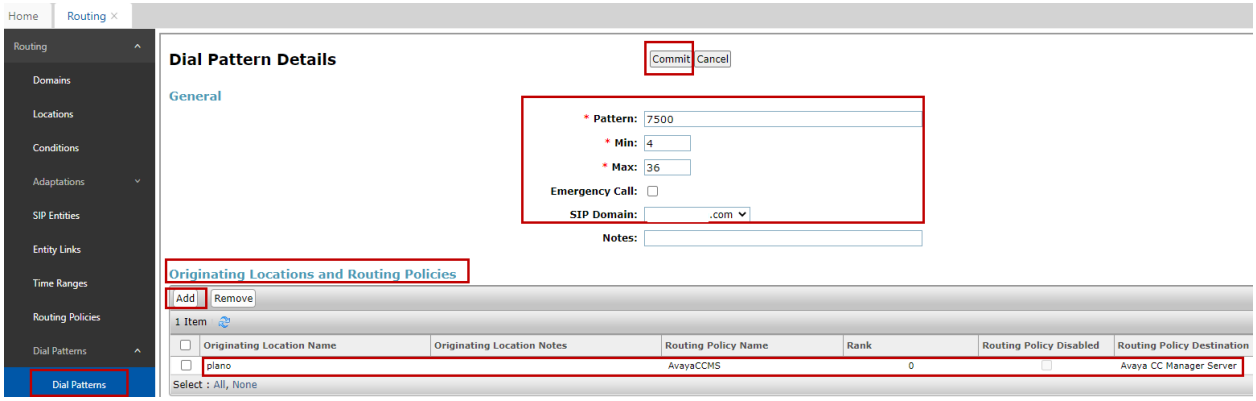



*Figure 101 Dial Pattern for Avaya Aura CC*

## 1.3 Avaya Aura Communication Manager Configuration

This section with screen shots taken from Avaya Aura CM gives a general overview of the Avaya Aura CM configuration.

### 1.3.1 Avaya Aura CM Login

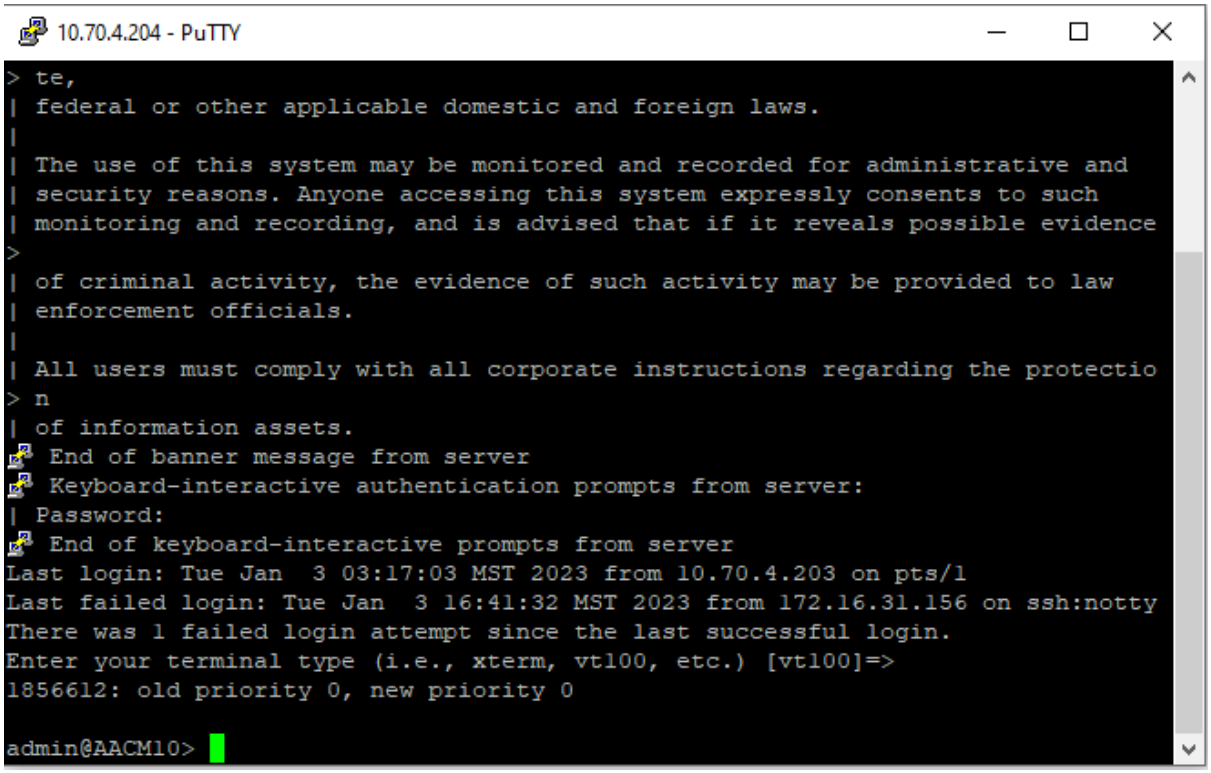



*Figure 102 Avaya Aura CM Login*

### 1.3.2 IP Node Name

Use the **list node-names ip** command to verify that node names are defined for Avaya Aura CM (procr), Session Manager **(AASM10).** The node names are needed for configuring the Signaling Group

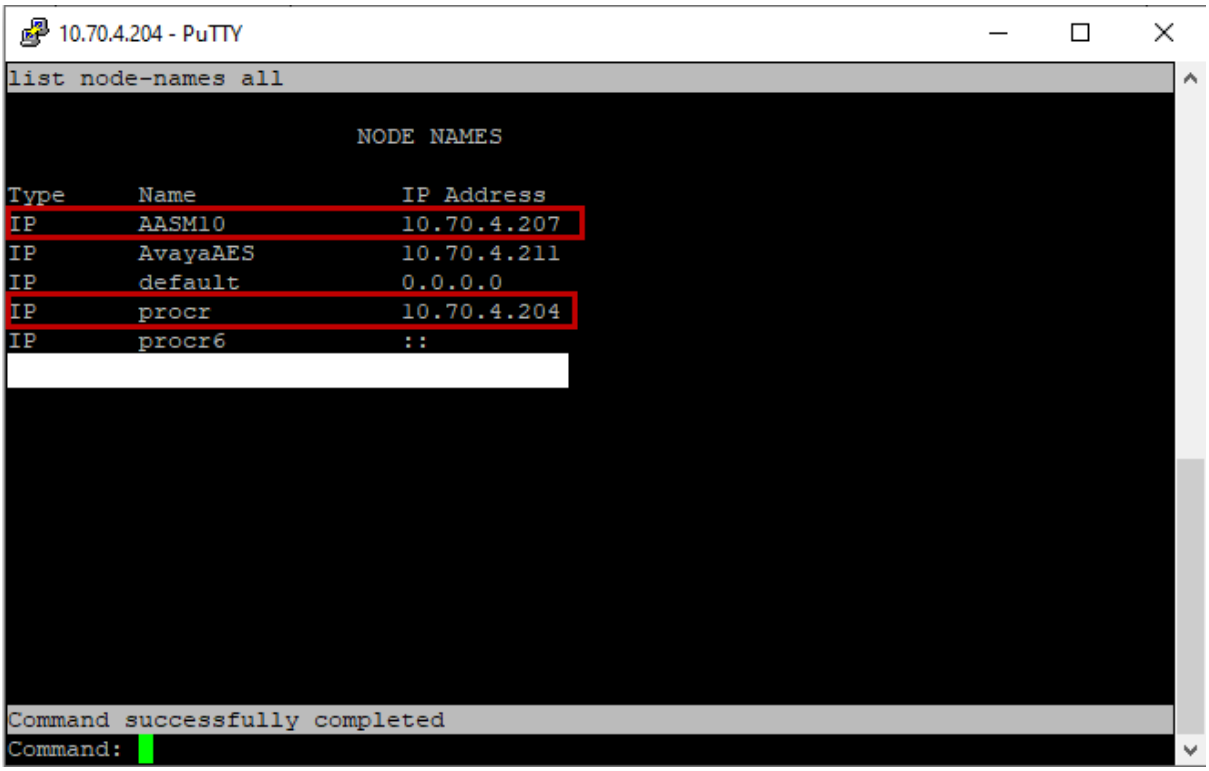



*Figure 103 IP Node Name*

### 1.3.3 IP Codec Set

Use **change ip-codec-set 1** to define list of codecs for calls between Avaya Aura CM and SM.

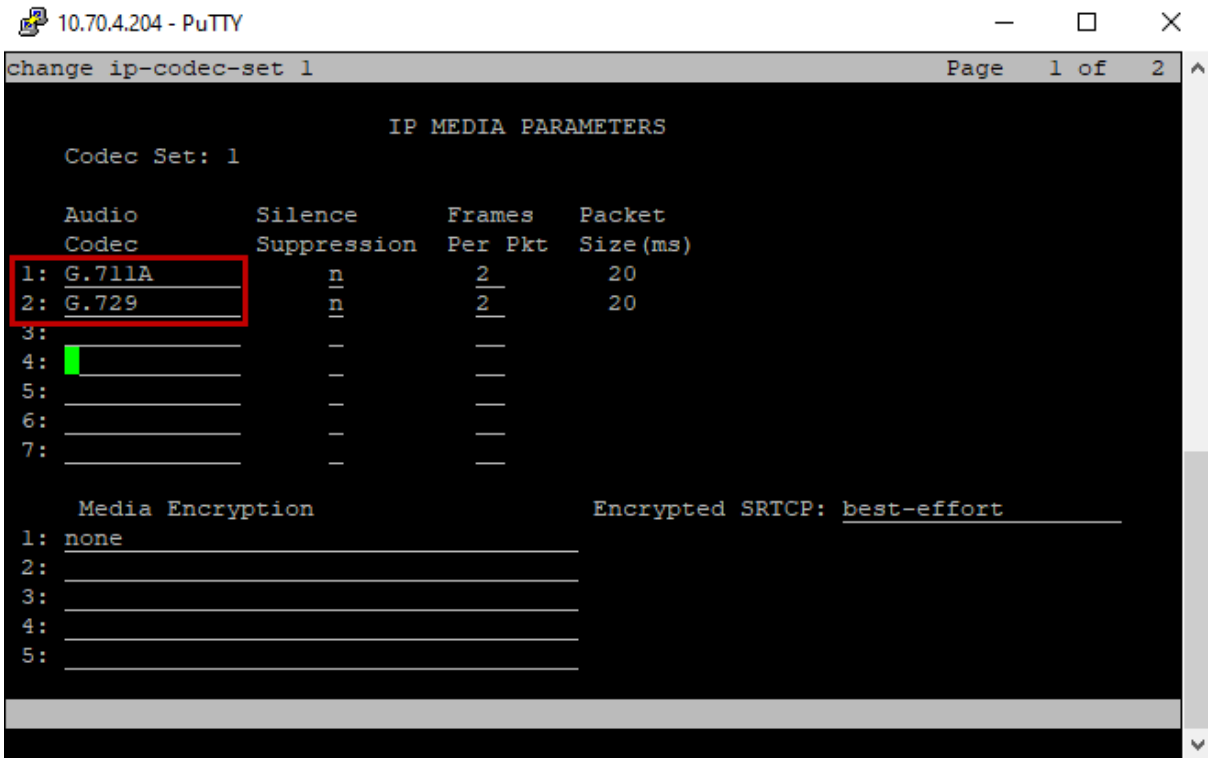



*Figure 104 IP Codec Set*

### 1.3.4 IP Network Region

- Use **change ip-network-region 1** to define the network region
- *Authoritative Domain*: Domain name **xxxxx.com**
- *Codec Set*: Enter codec set **1** created in Section 6.4.3
- *Intra-region IP-IP Direct Audio*: **yes**
- *Intra-region IP-IP Direct Audio*: **yes**

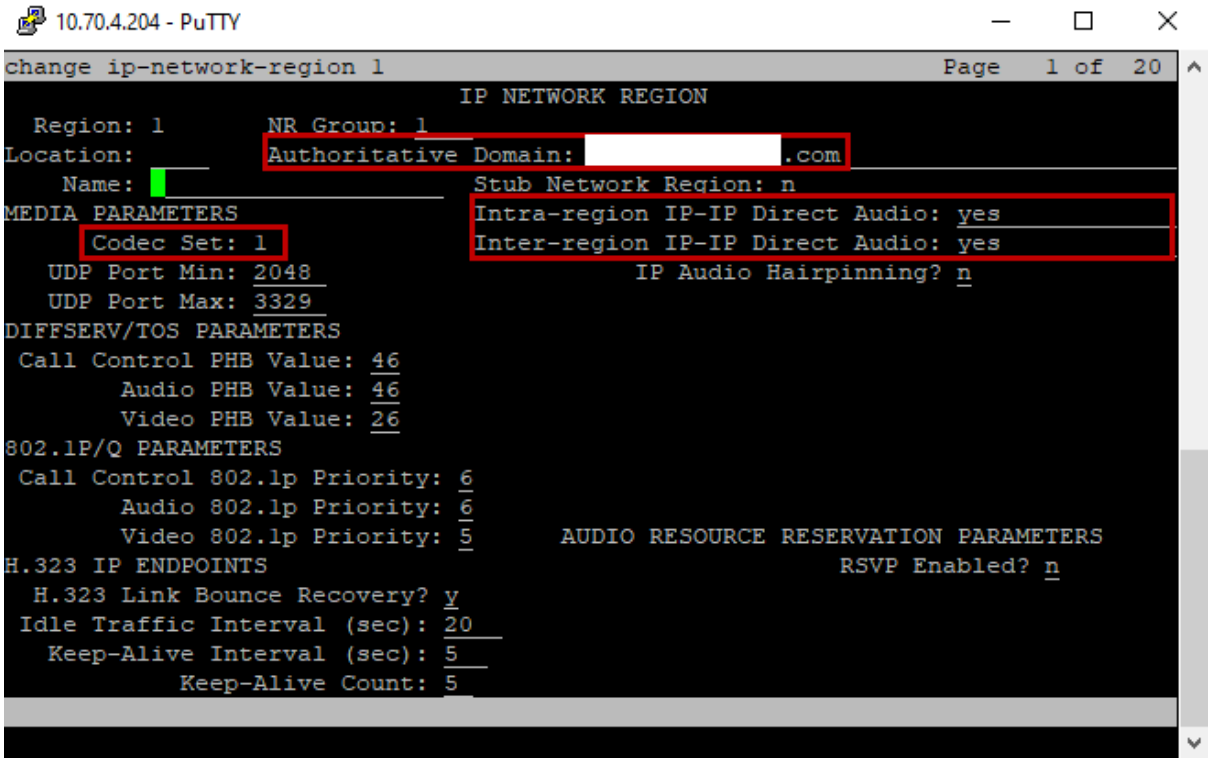



*Figure 105 IP Network Region*

### 1.3.5 Signaling Group

- Command **add signaling group 1** is used to create Signaling Group. Use **change signaling group 1** to modify existing signaling group.
- Set *Group Type*: **sip**
- Set *Transport Method*: **tcp**
- Set *Near-end Node Name*: **procr**
- Set *Near-end Listen Port*: **5060**
- Set *Far-end Node Name*: **AASM10**
- Set *Far-end Listen Port*: **5060**
- Set *Far-end Network Region*: **1**
- Set *Far-end Domain*: **xxxxx.com (Avaya Aura SM Domain name)**

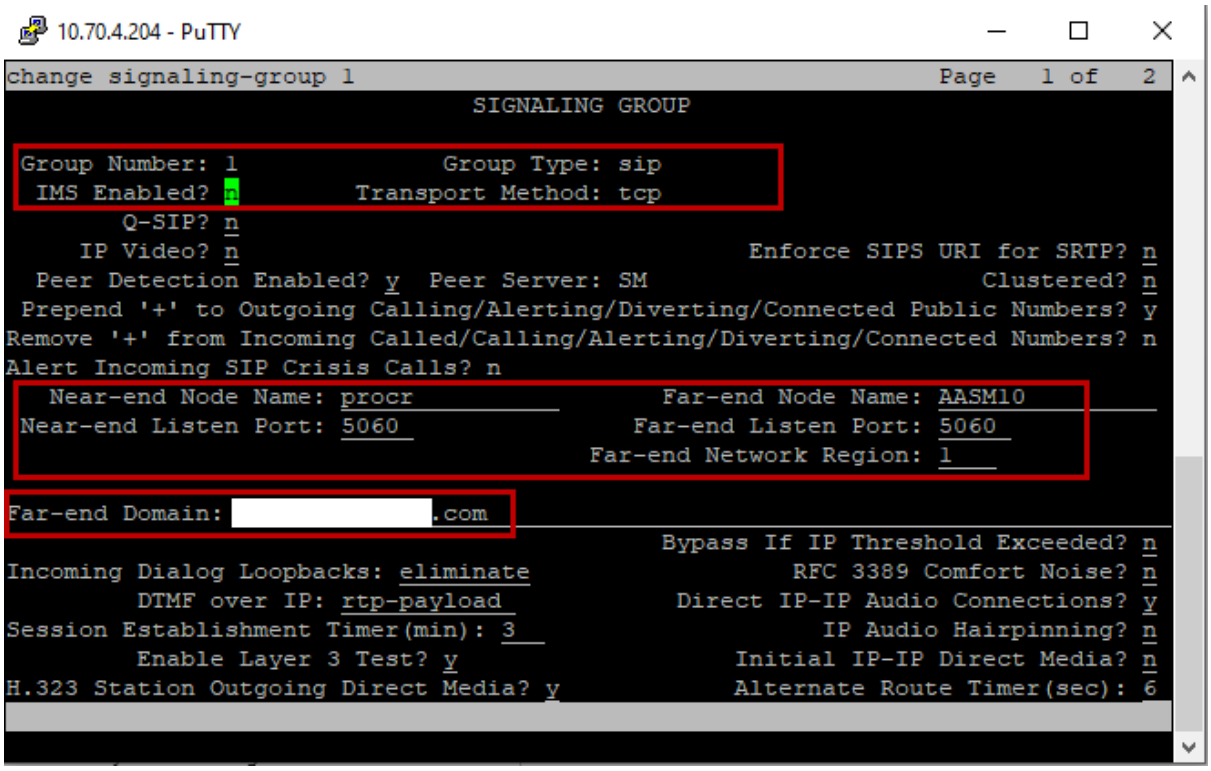



*Figure 106 Signaling Group*

### 1.3.6 Trunk Group

- Trunk group **1** is used for trunk to Avaya SM. Command **add trunk group 1** is used to create Trunk Group. Use **change trunk group 1** to modify existing trunk group.
- Set *Group Type*: **sip**
- Set *Group Name*: **SIP Trunk**
- Set *TAC*: **#001**
- Set *Direction*: **two-way**
- Set *Service Type*: **tie**
- Set *Member Assignment Method*: **auto**
- Set *Signaling Group*: **1** (created in Section 6.4.5)
- Set *Number of Members*: **10**

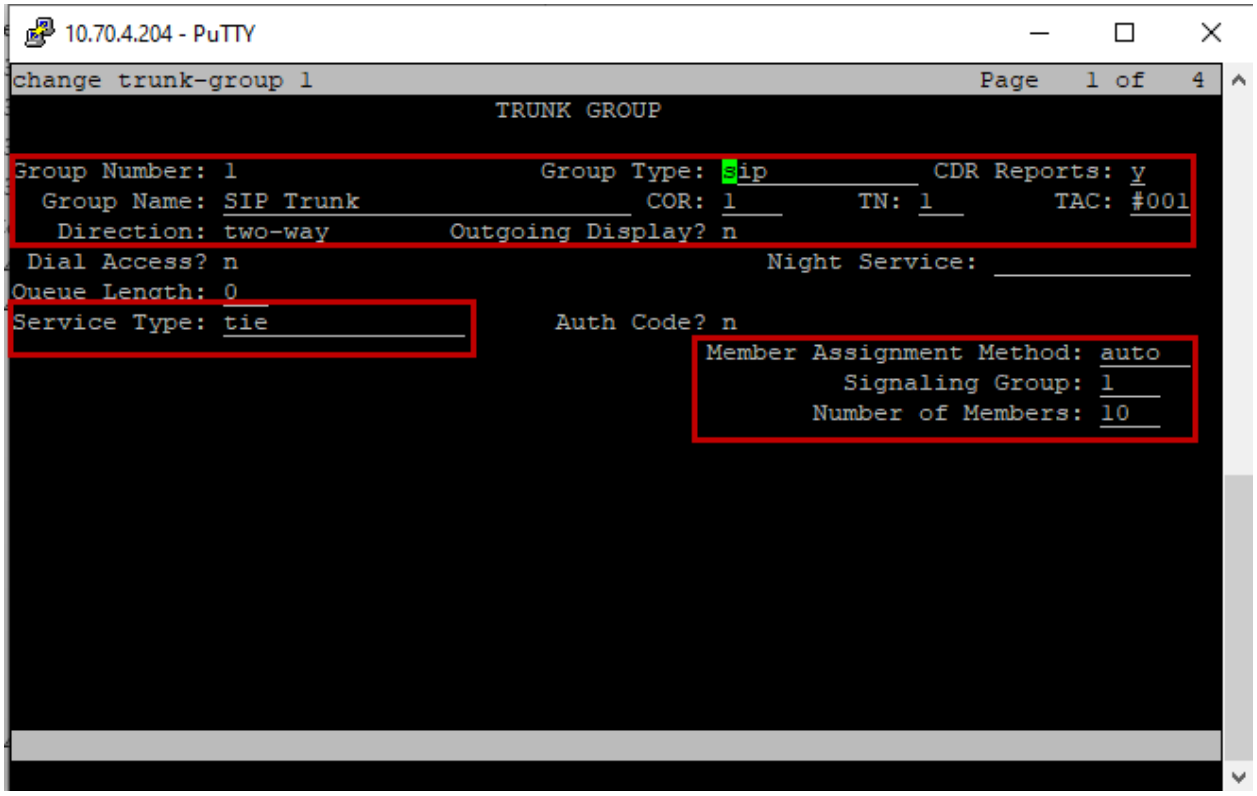



*Figure 107 Trunk Group*

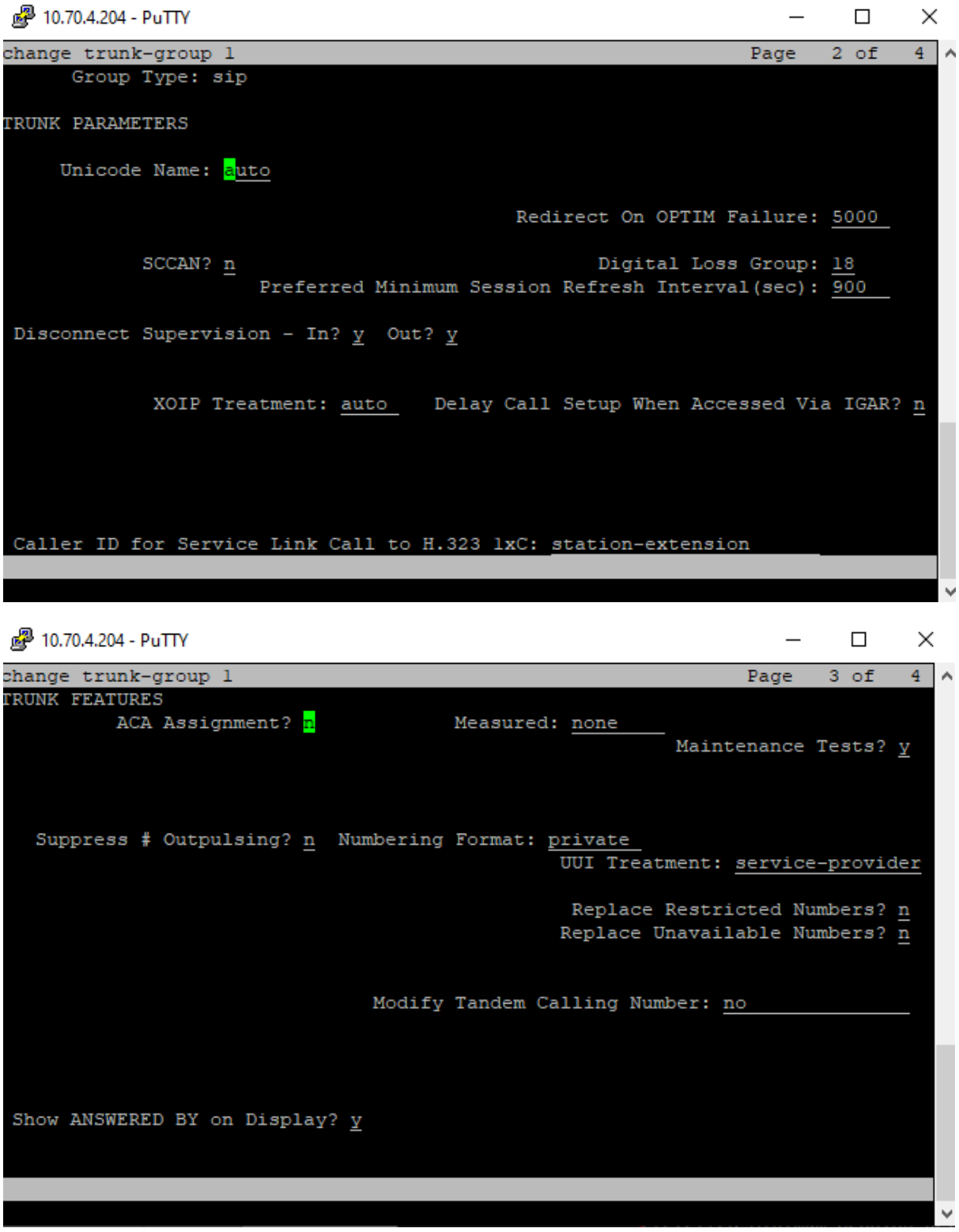



*Figure 108 Trunk Group Continuation*

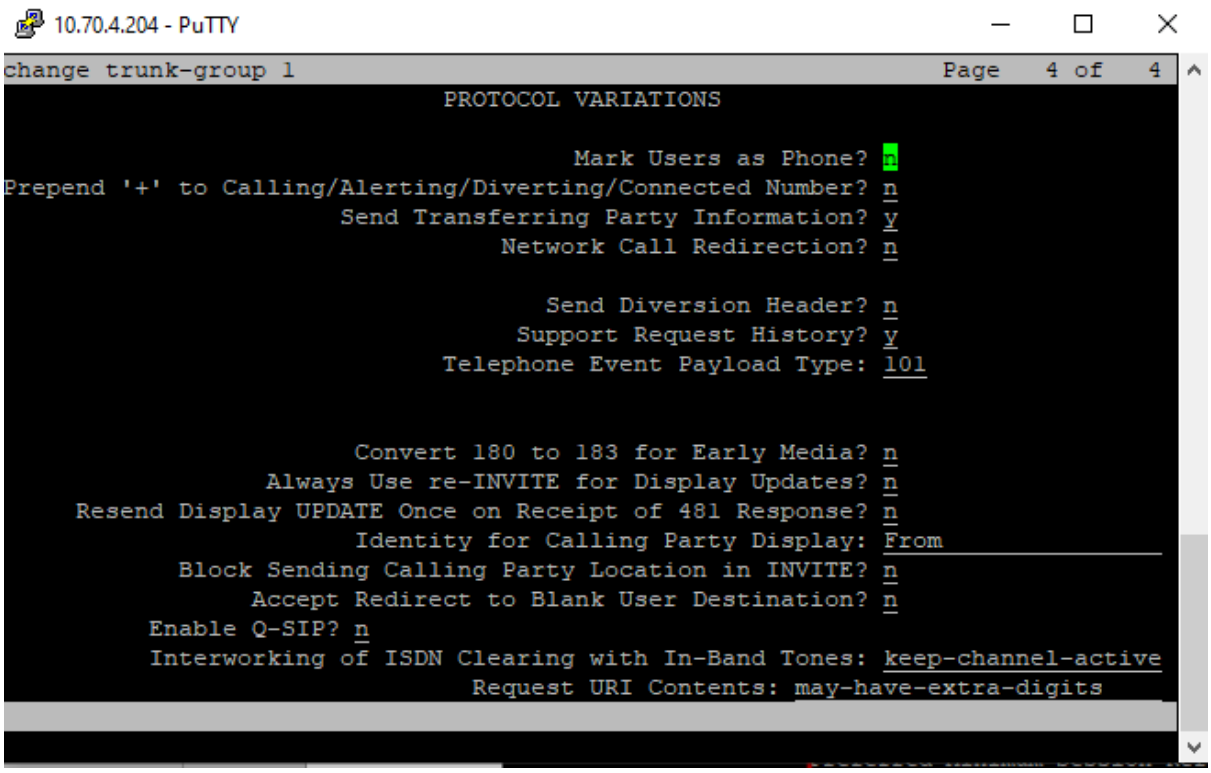



*Figure 109 Trunk Group Continuation*

### 1.3.7 Station

- The command **add station 3001** is used to add an extension
- *Type*: **9620**
- *Name*: **Avaya Deskphone**
- *Security Code*: Enter a **Password**
- *IP Softphone*: **y**

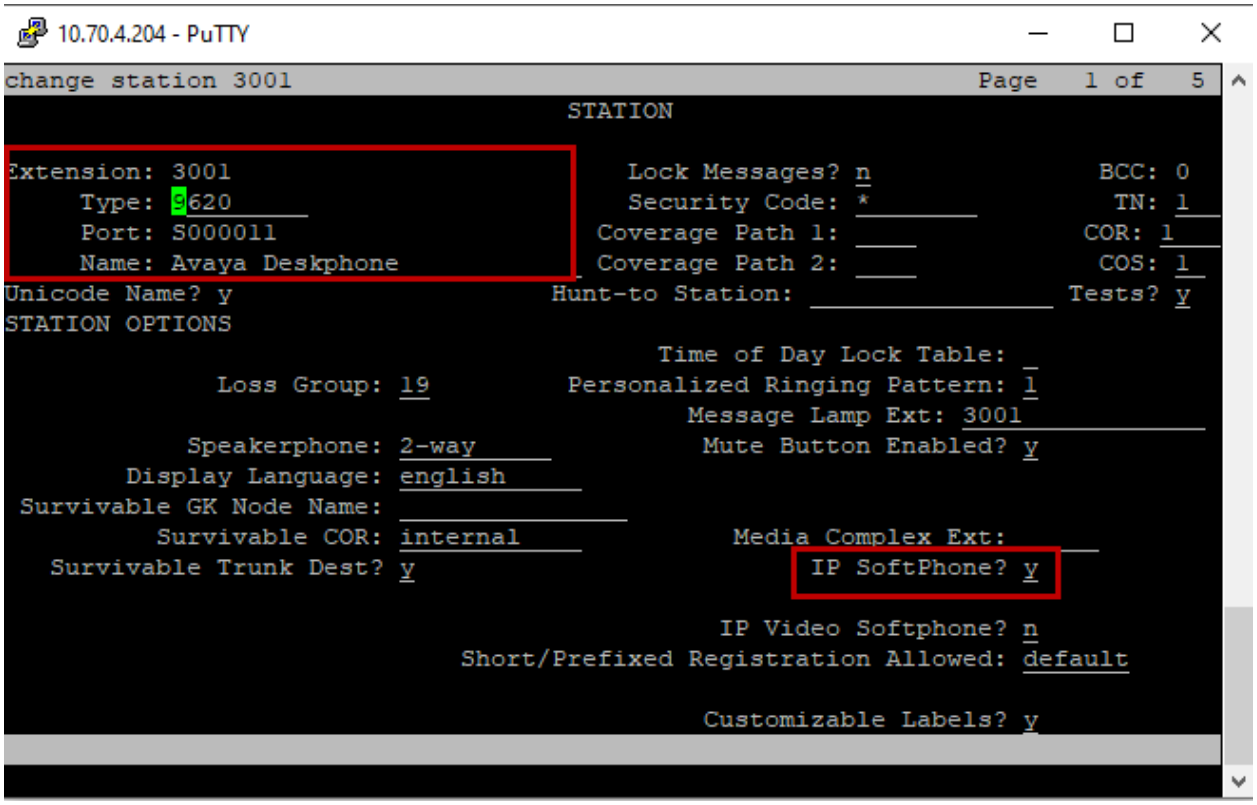



*Figure 110 Add Station*

### 1.3.8 Application Enablement Services (AES) Configuration

**List Node Names**

Type **list node-names all** and verify if AES server IP is configured

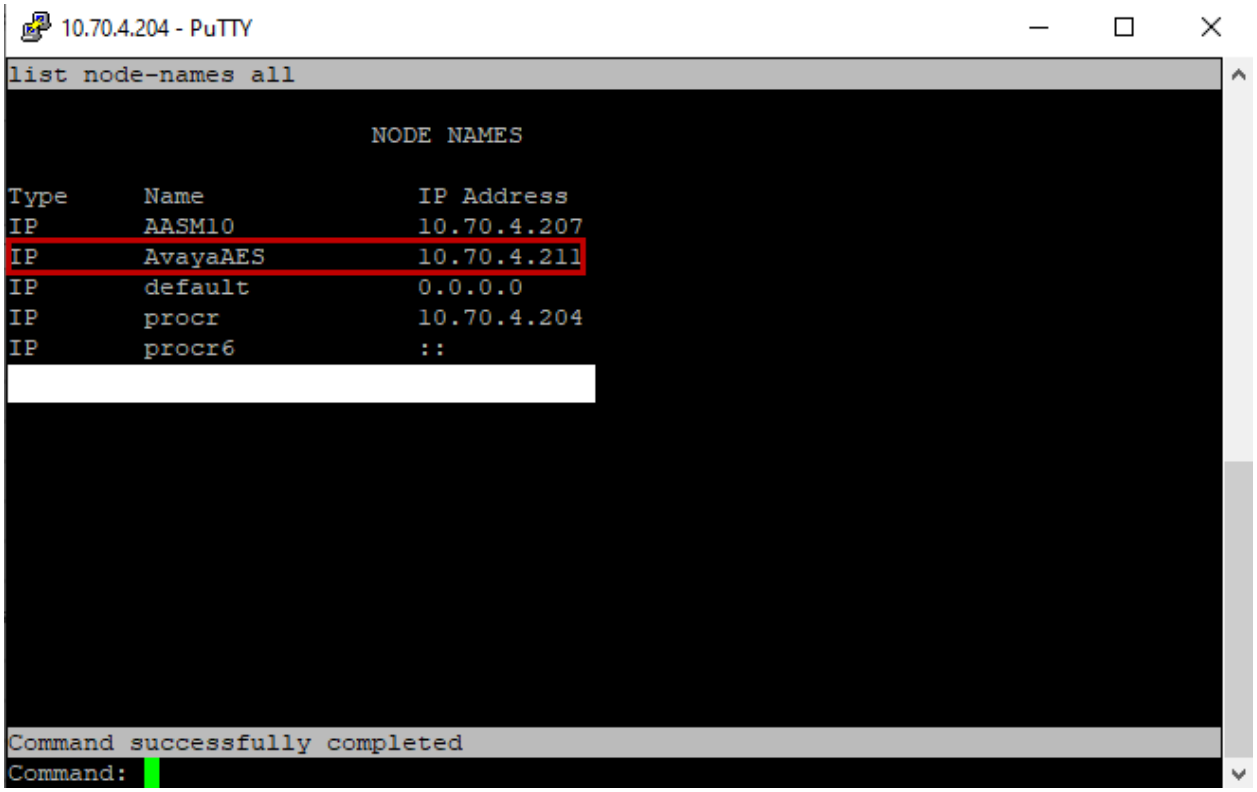



*Figure 111 List Node Names - AES IP Address*

**Configuration of IP services for AES transport link**

- The Command **change ip-services** is used to configure IP services
- Set *Type*: **AESVCS**
- Set *Enabled*: **Y**
- Set *Local Node*: **procr**
- Set *Local Port*: **8765**

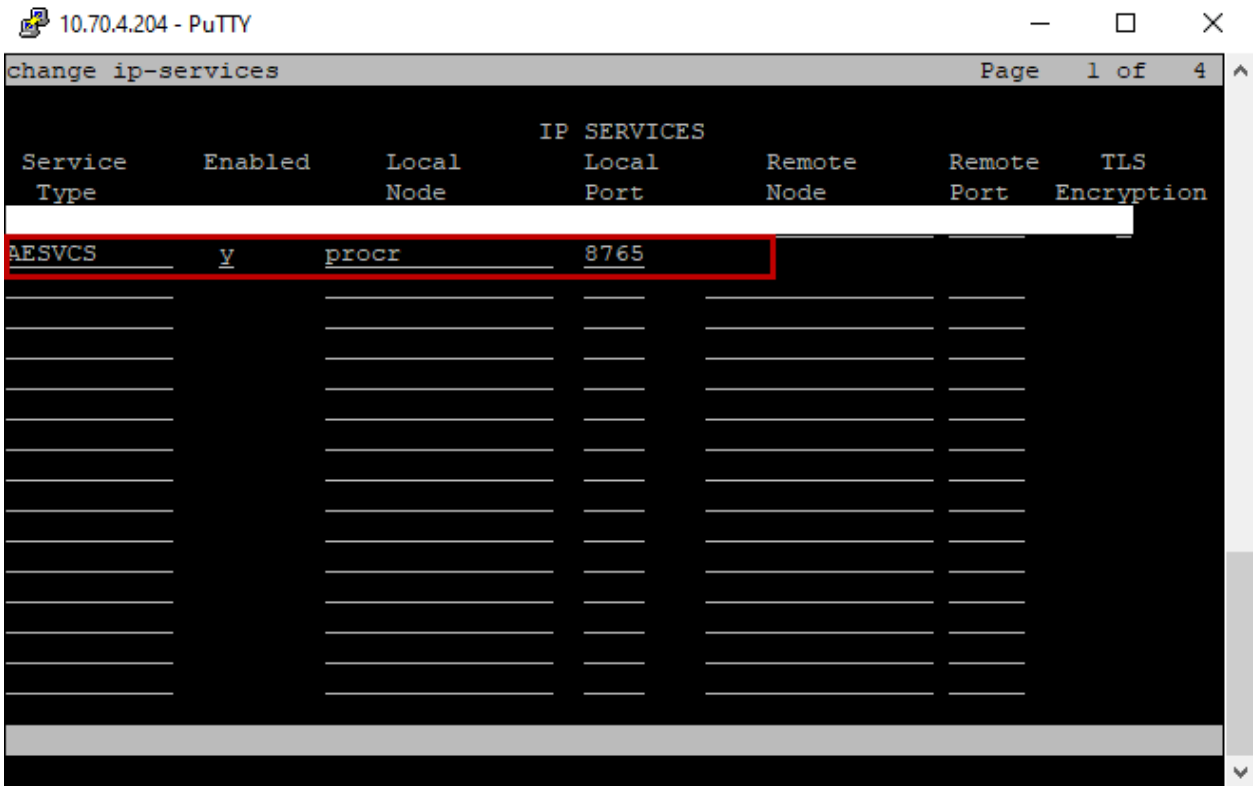



*Figure 112 AES IP Services*

- *AE Services Server*: **aaaes10 (host name of AES server)**
- Set *Password*: **Enter a password (This must matches the name and password of the AES server)**
- Set *Enabled*: **Y**

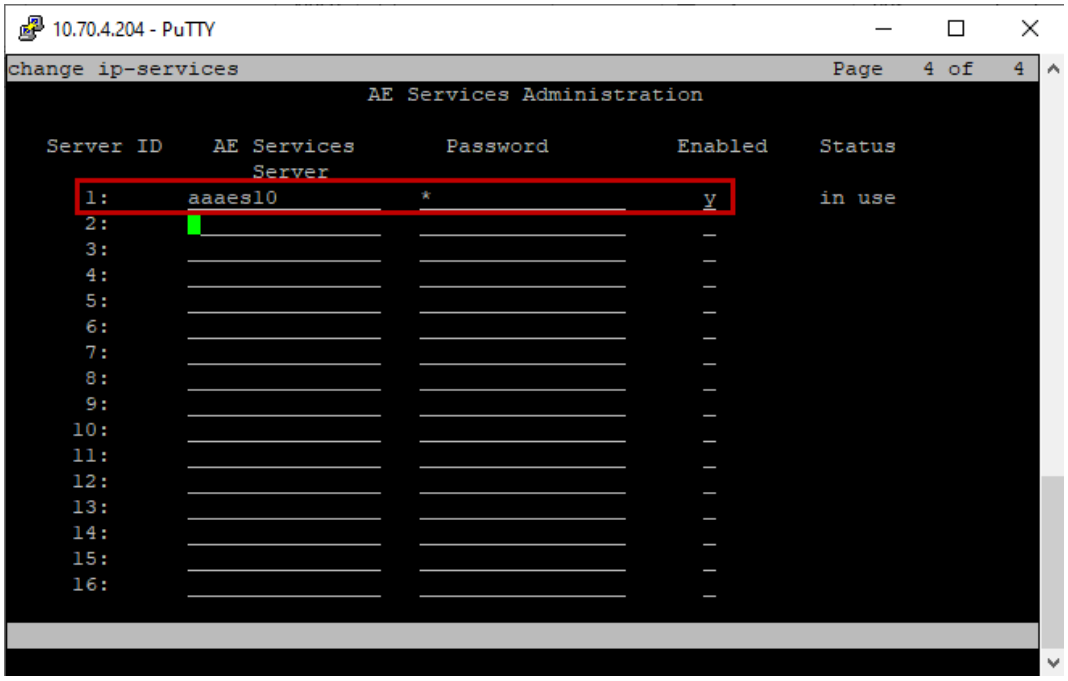



*Figure 113 AES IP Services Continuation*

**Configuration of CTI Link for AES**

- The Command **add cti-link** is used to configure CTI link to AES server
- Set *Extension*: **An available Extension number**
- Set *Type*: **ADJ-IP**
- Set *Name*: **CTI to AES**

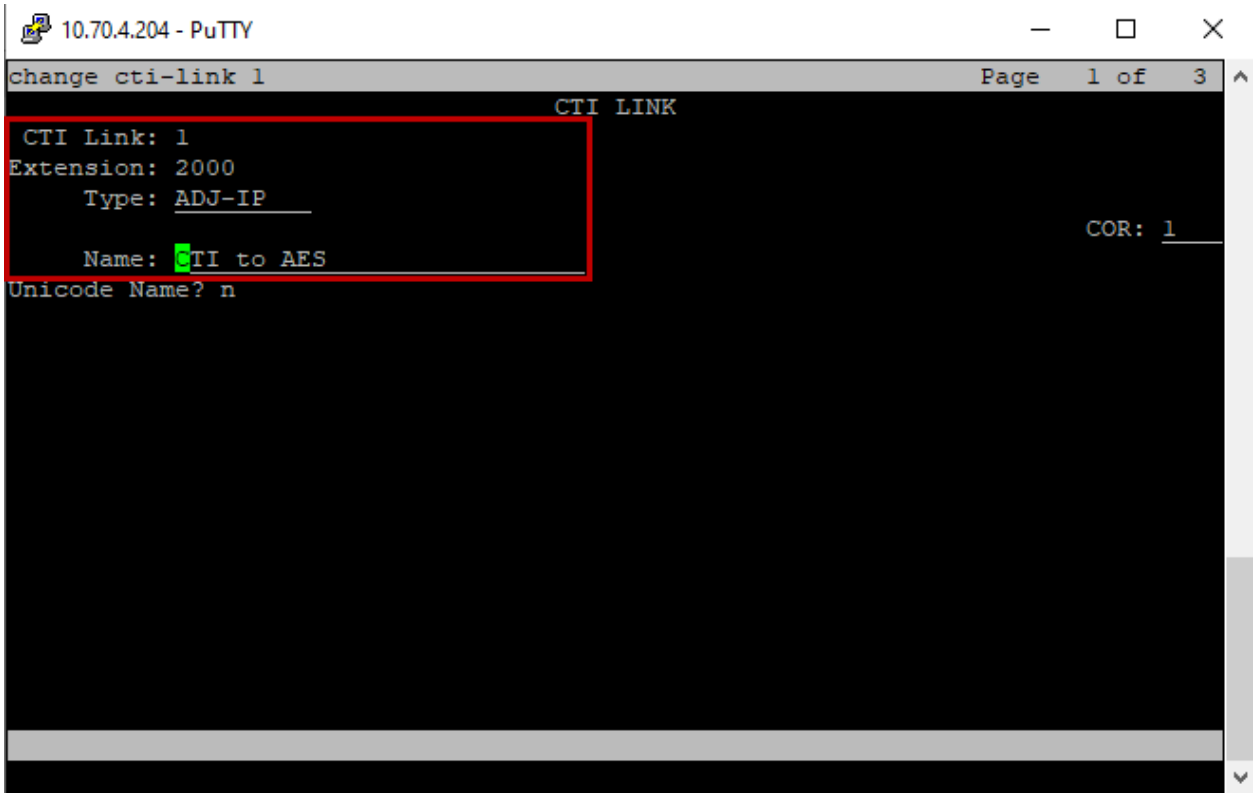



*Figure 114 AES CTI Link*

## 1.4 Application Enablement Services (AES) Configuration

Avaya Aura Application Enablement Services (AES) is a set of enhanced telephony APIs, protocols, and Web services. These applications support access to the call processing, media, and administrative features available in Communication Manager

### 1.4.1 AES Login

- Access AES Web login screen via **https://<AES IP Address>**
- Use *cust* as **User ID** and associated **Password**
- Click **Login**

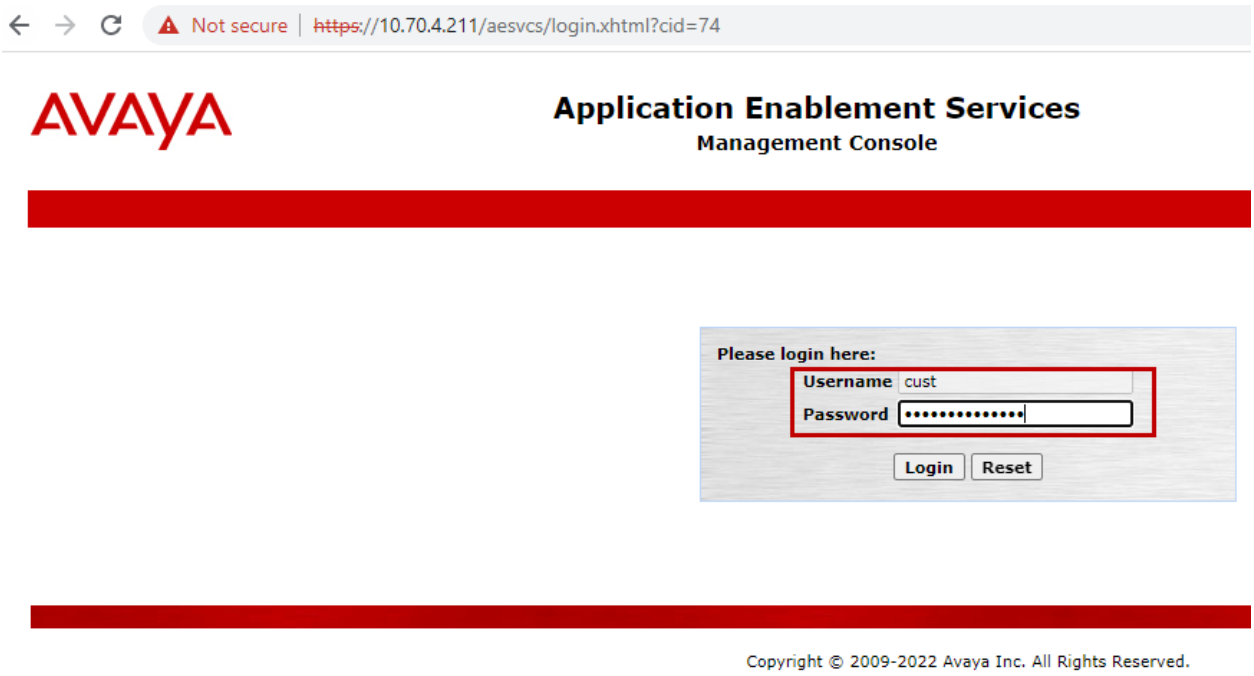



*Figure 115 AES Login*

### 1.4.2 Communication Manager Switch Connections

- Navigate to **Communication Manager Interface > Switch Connections**
- Under *Switch Connections*, type the host name of **Avaya Aura CM** (e.g. AACM10)
- Click **Add Connection**

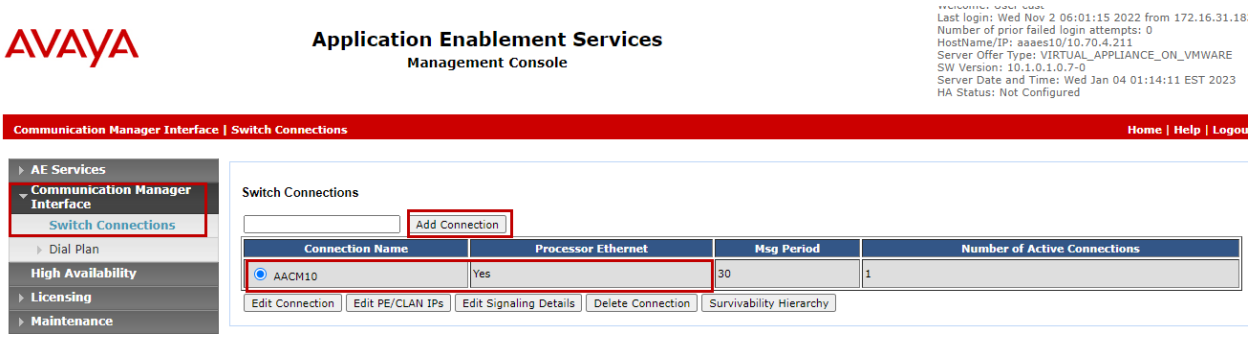



*Figure 116 AES Avaya CM Switch Connections*

- *Switch Password*: Type the Avaya CM **AES IP Service password** (This password must match the password entered when configuring IP Services (Section 6.4.8)
- *Confirm Switch Password*: **Type the above password again**
- *Processor Ethernet*: **Enabled**
- Click **Apply**

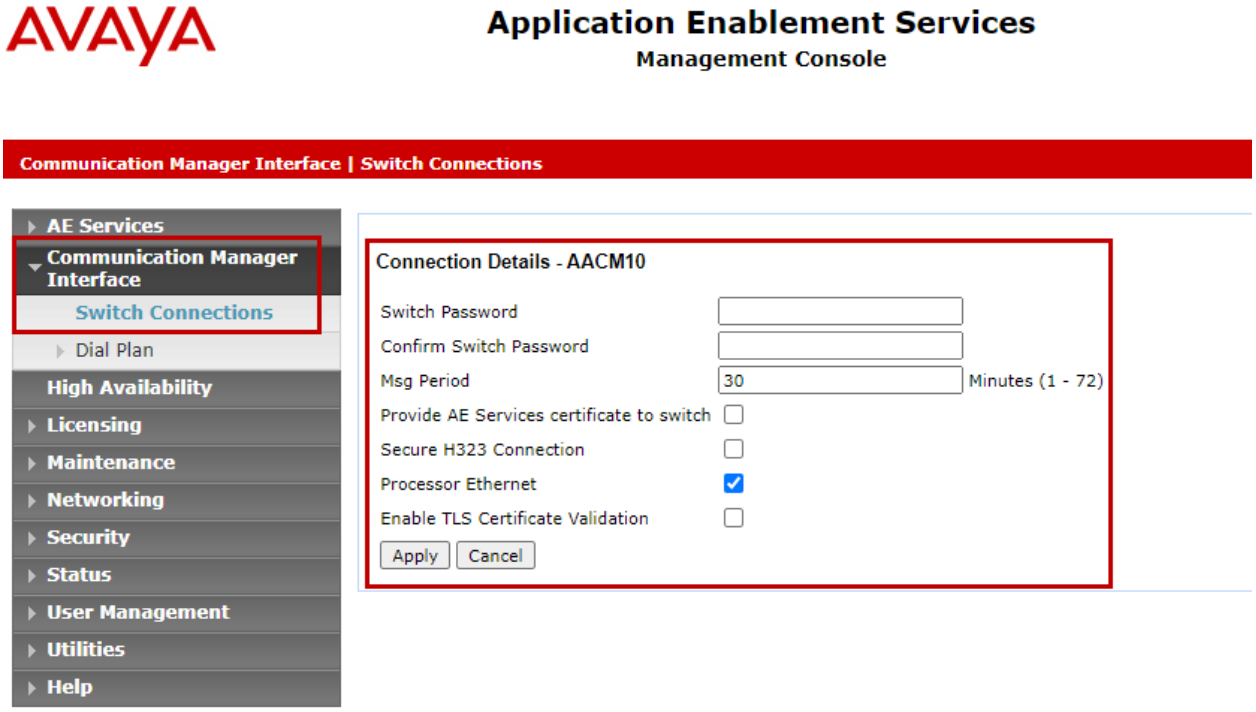



*Figure 117 AES Avaya CM Switch Connections Continuation*

### 1.4.3 Communication Manager Switch Connections CLAN IP

- Navigate to **Communication Manager Interface > Switch Connections**
- Under *Switch Connections*, Select the Connection Name **AACM10**
- Click **Edit PE/CLAN IPs**

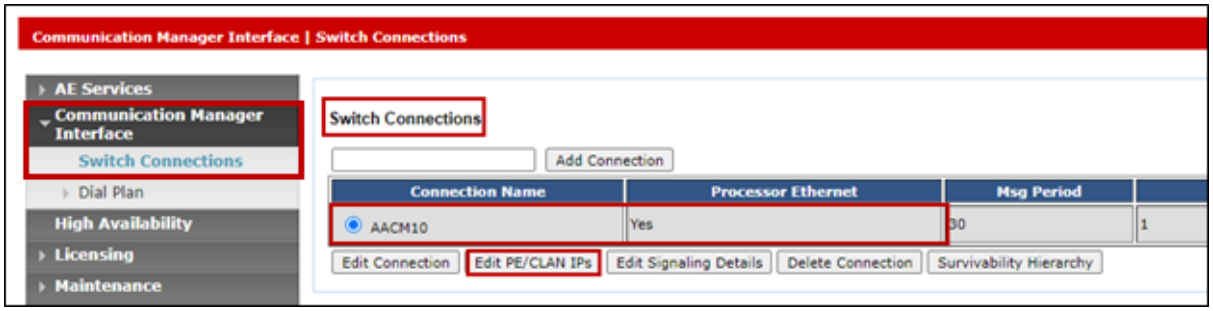



*Figure 118 AES Avaya CM Switch Connections CLAN IP*

- Under *Edit Processor Ethernet IP*, type the **IP address of Avaya Aura CM** (e.g. 10.70.4.204)
- Click **Add/Edit Name or IP**

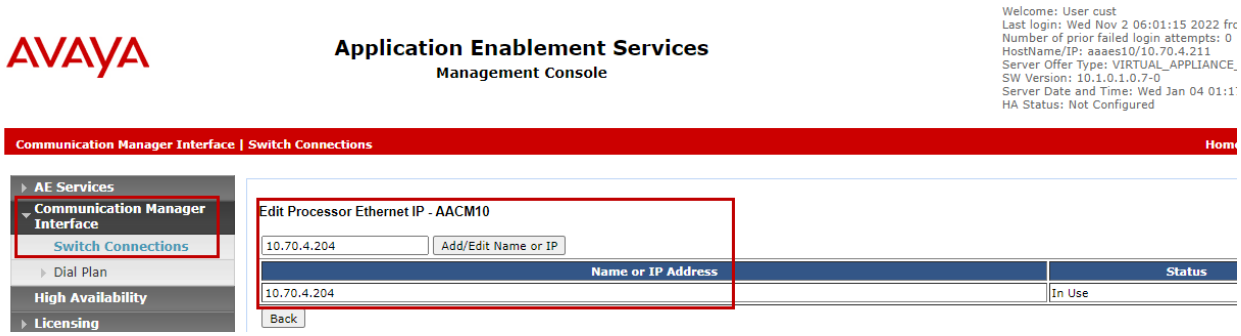



*Figure 119 AES Avaya CM Switch Connections CLAN IP Continuation*

### 1.4.4 CTI link to Communication Manager

- Navigate to **AE Services > TSAPI > TSAPI Links**
- Under *TSAPI Links*, Click **Add Link**
  - *Link:* **1**
  - *Switch Connection:* **AACM10**
  - *Switch CTI Link Number:* **1** (This is the CTI link number which is configured in Avaya CM Section 6.4.8)
- Click **Apply Changes**

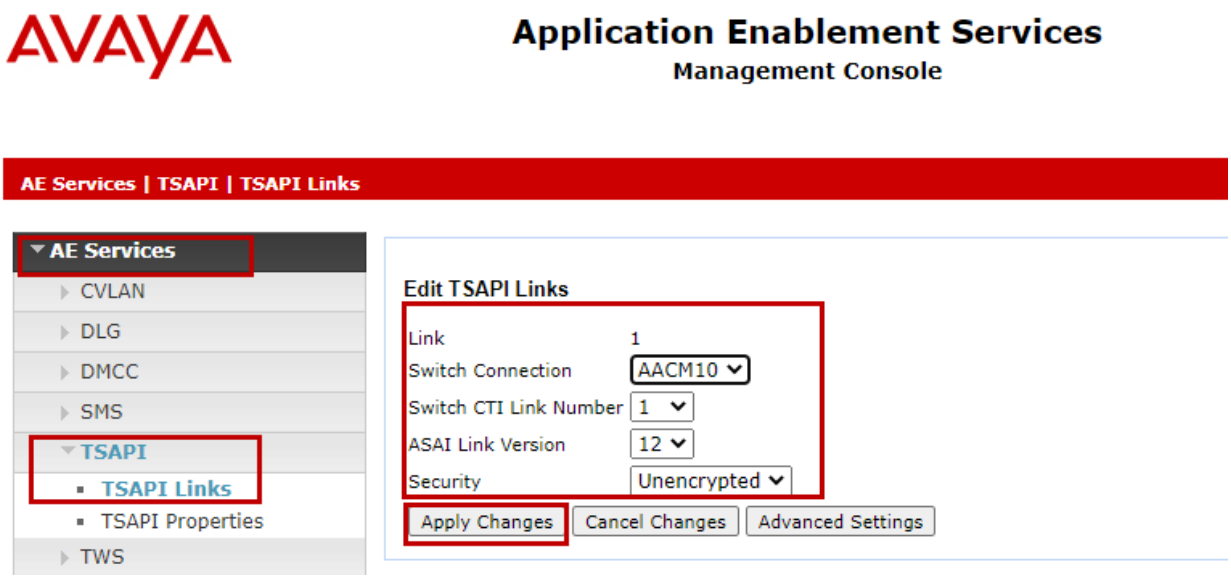



*Figure 120 Adding a CTI Link*

### 1.4.5 TR87 on the AES

- Navigate to **Networking > Ports**
- Under *DMCC Server Ports section*, **Enable** the *TR/87 Port*

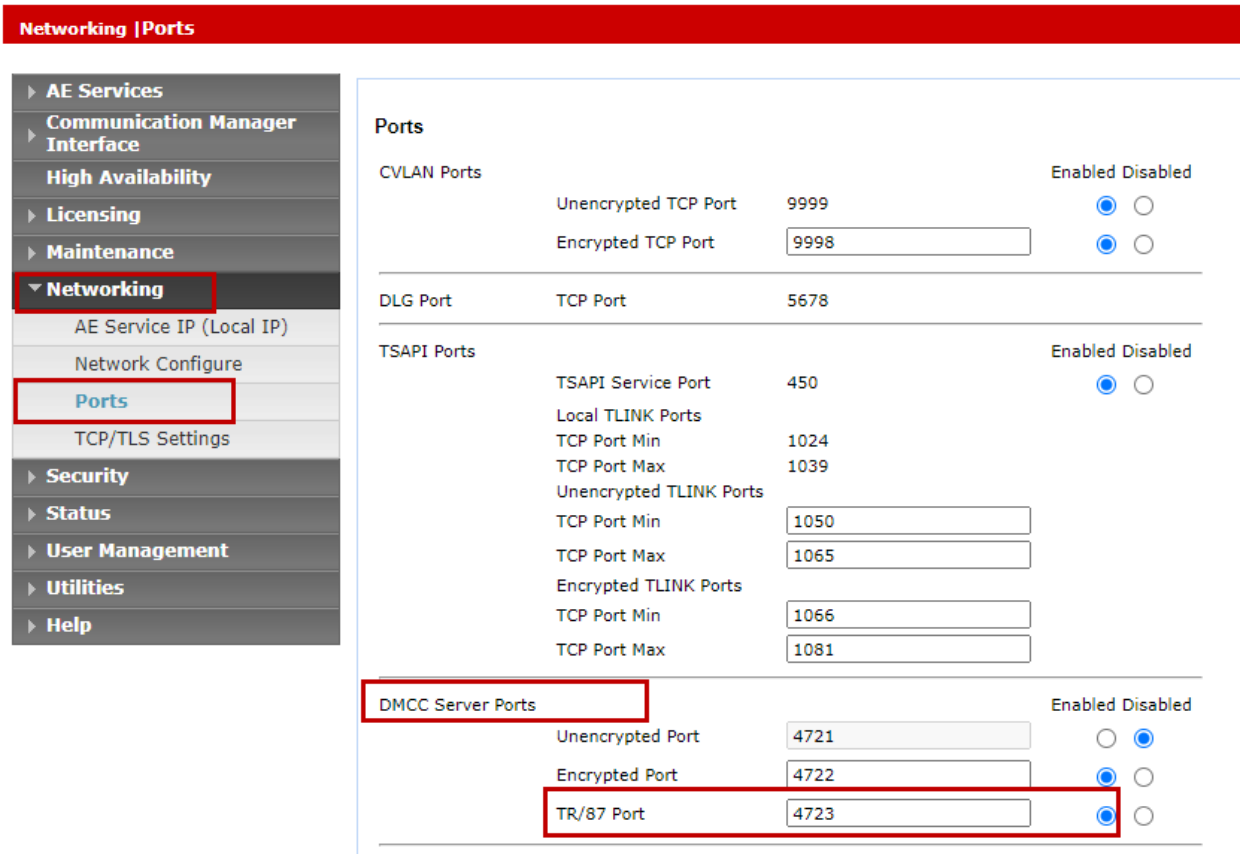



*Figure 121 Enabling TR87 Port*

### 1.4.6 Security on the AES

- Navigate to **Security > Host AA > Service Settings**
- *TR/87*: **Require Trusted Host Entry is enabled**

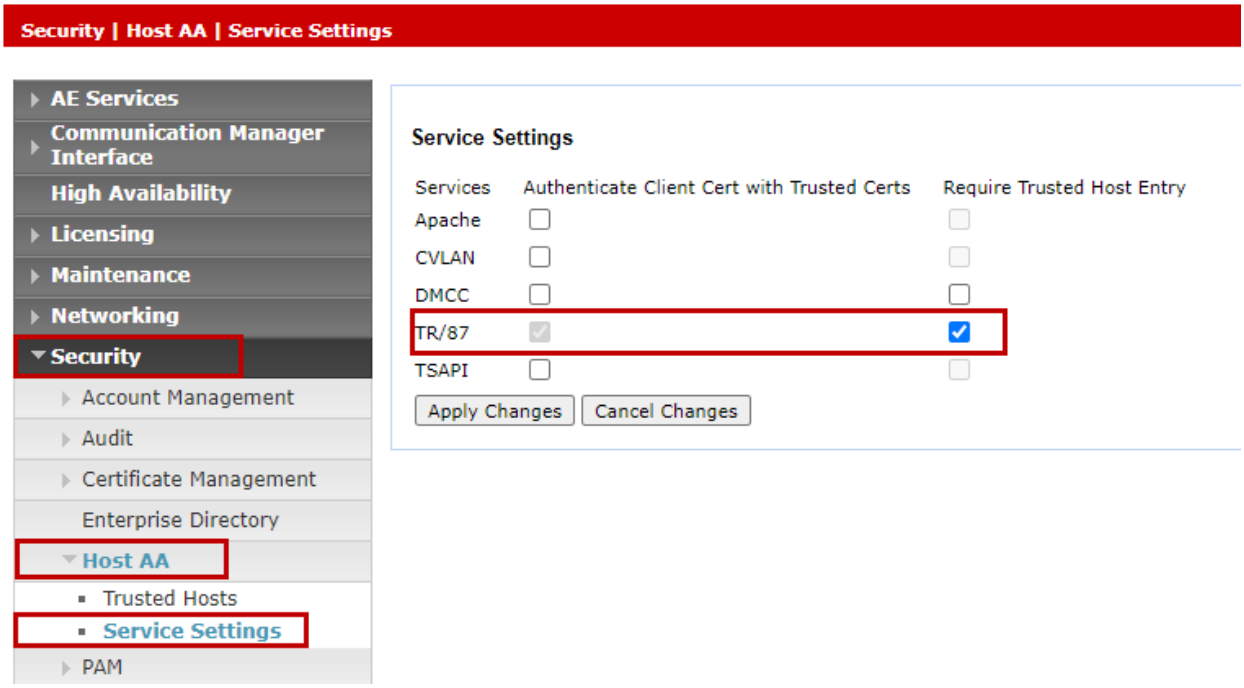



*Figure 122 Configuring Security*

### 1.4.7 Certificate Authority root certificate in to AES

- Navigate to **Security > Certificate Management > CA Trusted Certificates**
- Click **Import**
- Under the **Trusted Certificate Import**,
  - *File Path*: Click Choose File and browse Root CA file **(e.g. RootCA – Copy.pem)**
  - *Certificate Alias:* **RootCA**
- Click **Apply**

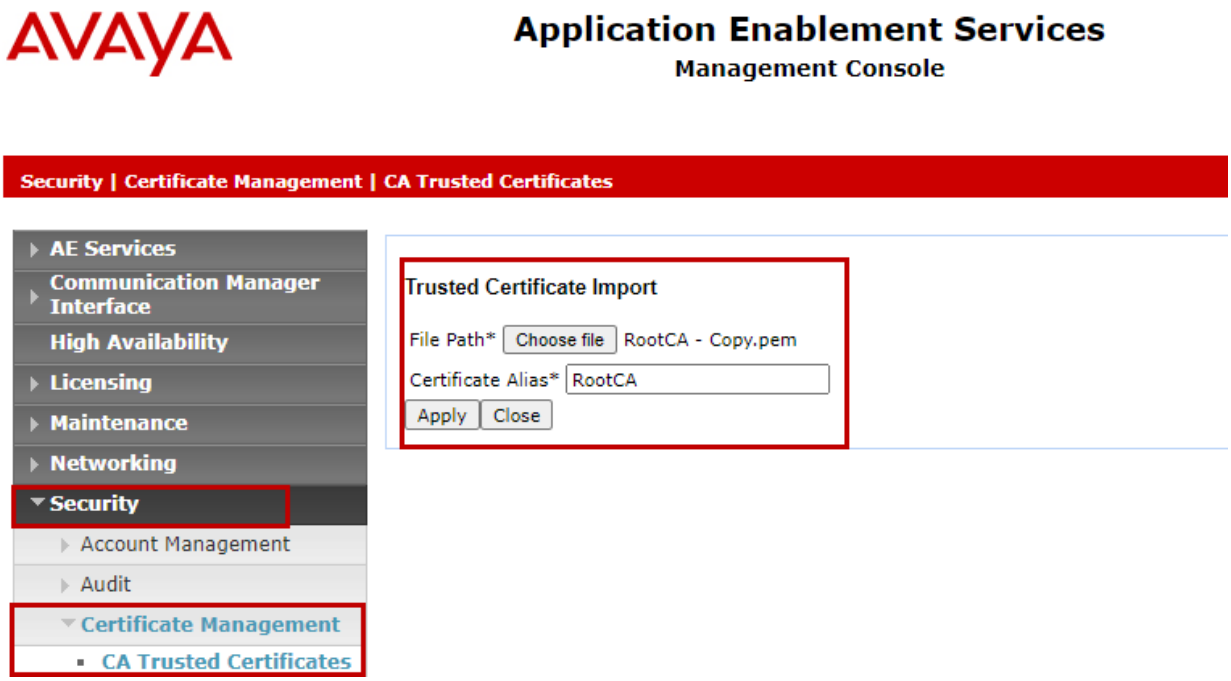



*Figure 123 Importing Root CA in to AES*

- The uploaded **RootCA** is displayed under the CA Trusted Certificates with the status as “Valid’

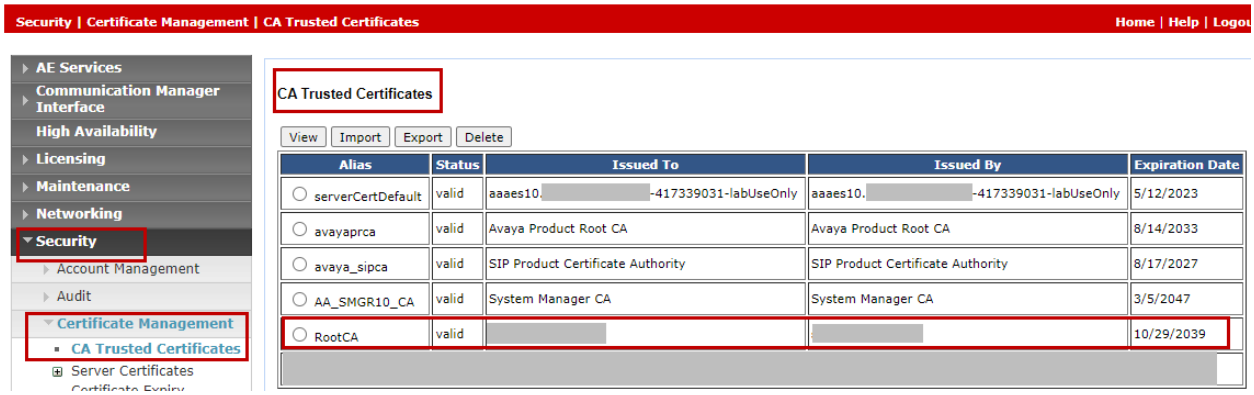



*Figure 124 Importing Root CA in to AES Continuation*

### 1.4.8 AES Certificate Signing Request

- Navigate to **Security > Certificate Management > Server Certificates**
- Click **Add**
- Under the **Add Server Certificate**,
  - *Certificate Alias*: select **aesservices**
  - *Password*: Type **certificate key password**
  - *Re-enter Password*: **Reenter the above password again**
  - *Key size*: **2048**
  - *Distinguished Name (DN)*:* **cn=aaaes10,dc=xxxx,dc=com (where xxxx=domain name of AES)**
  - *Challenge Password*: Type **certificate request password**
  - *Re-enter Challenge Password*: **Type the above password again**

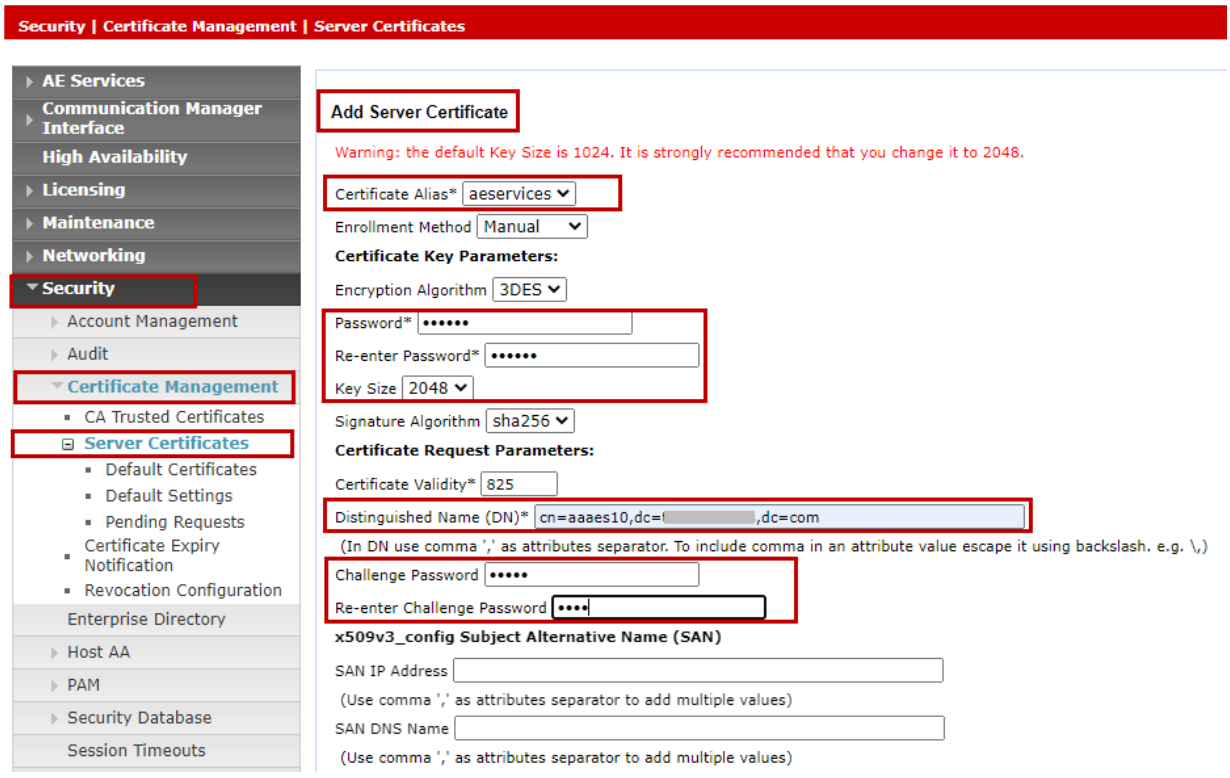



*Figure 125 Generating AES CSR*

- Click **Apply**
- A **Server Certificate Manual Enrollment Request** page appears
- Copy all the text in **the Certificate Request PEM** box in to a text file. This is the Certificate Signaling Request (CSR) text.
- Get the CSR signed from a Certificate Authority

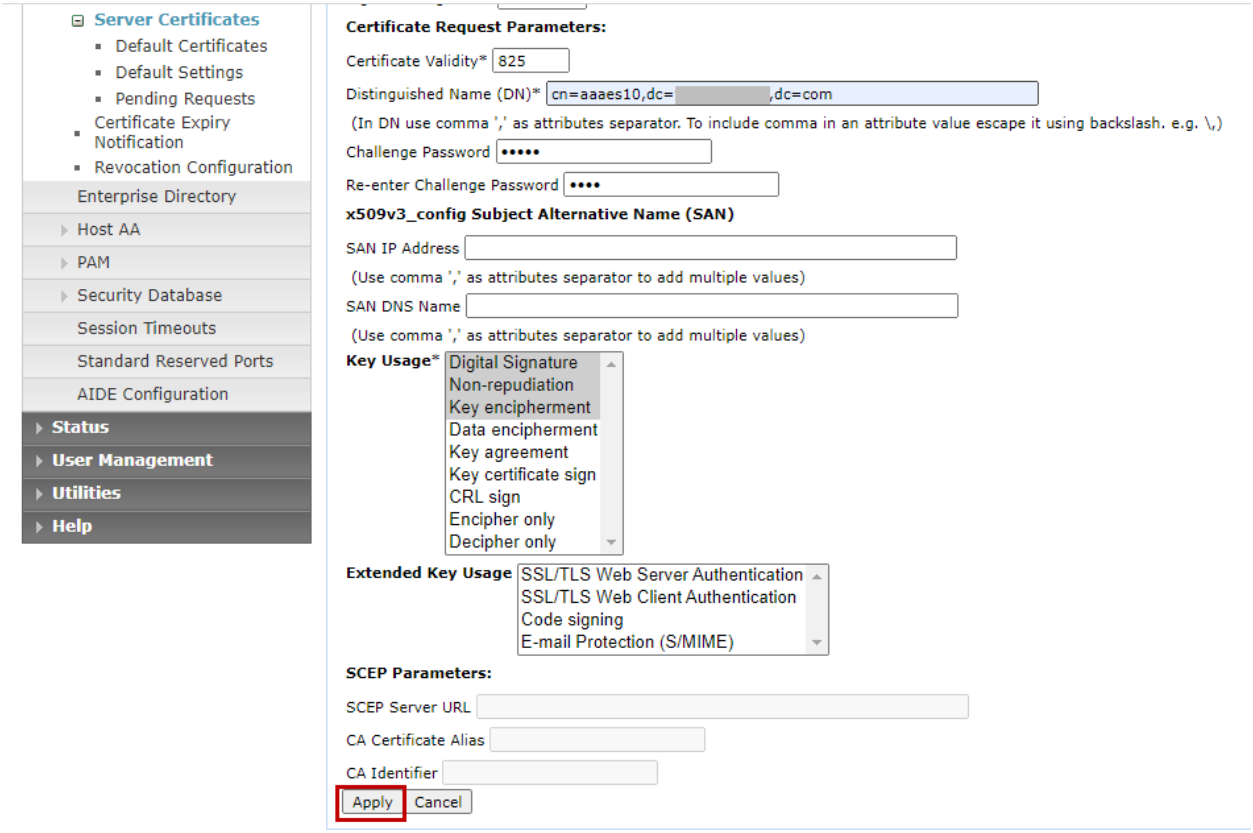



*Figure 126 Generating AES CSR Continuation*

### 1.4.9 Signed Certificate in to AES

- Navigate to **Security > Certificate Management > Server Certificates**
- Click **Import**
- Under the **Server Certificate Import,**
  - *File Path*: Click Choose File and browse Signed Certificate file (e.g. **AESSignedwithclientandserverauth.cer**)
  - *Establish Chain of Trust*: **Enabled**
  - *Certificate Alias*: **aesservices**
- Click **Apply**

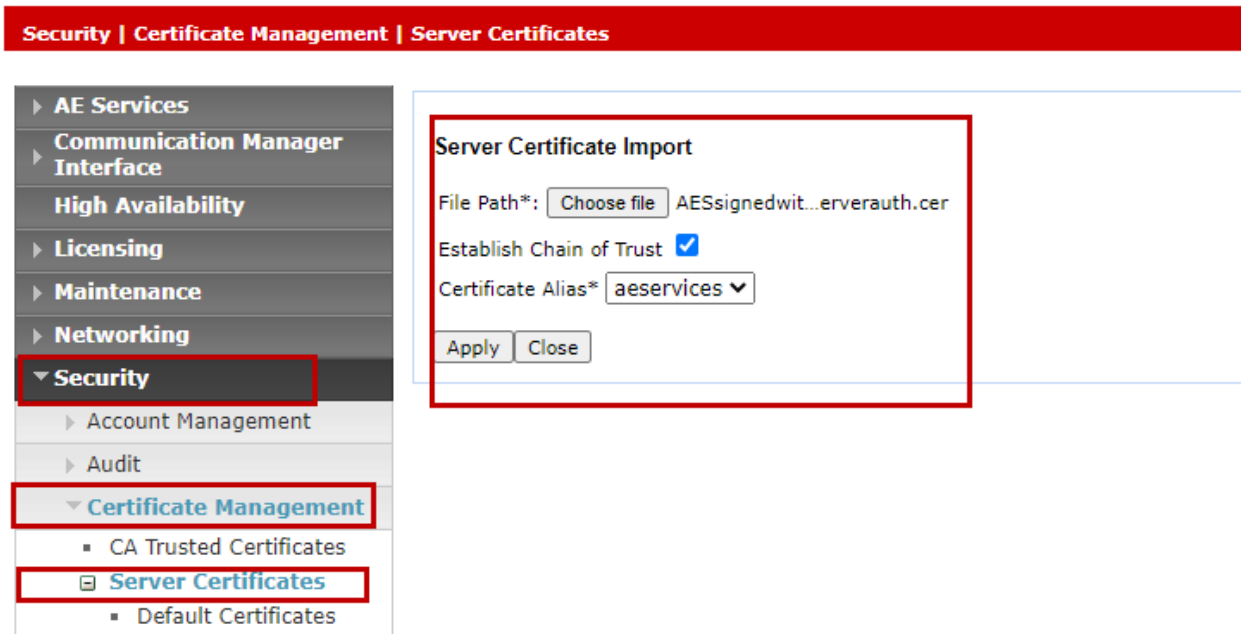



*Figure 127 Importing Signed Certificate in to AES*

- The Server Certificate **aesservices** is imported with the status displayed as “valid”

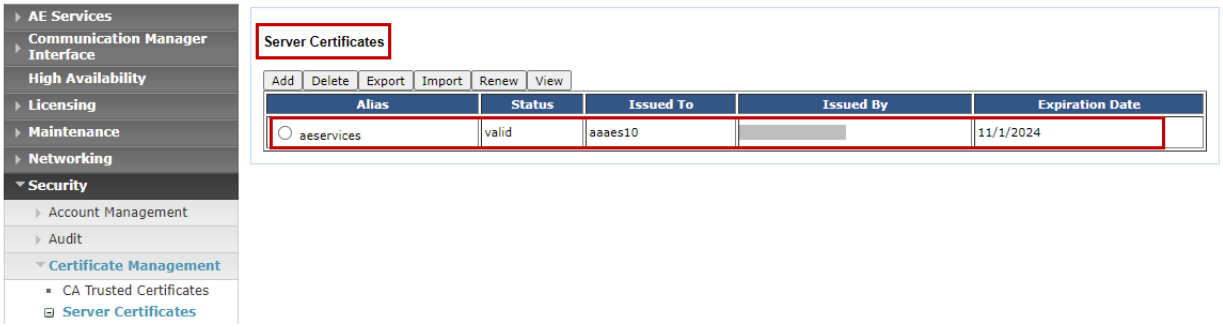



*Figure 128 Importing Signed Certificate in to AES Continuation*

### 1.4.10 Avaya Aura Contact Server as a trusted host on AES

- Navigate to **Security > Host AA > Trusted Hosts**
- Click **Add**
- Under **Add Trusted Host**,
  - *Certificate CN or SubAltName*: **Type FQDN of the Avaya Aura CC**
  - *Service Type*: **TR/87**
- Click **Apply Changes**

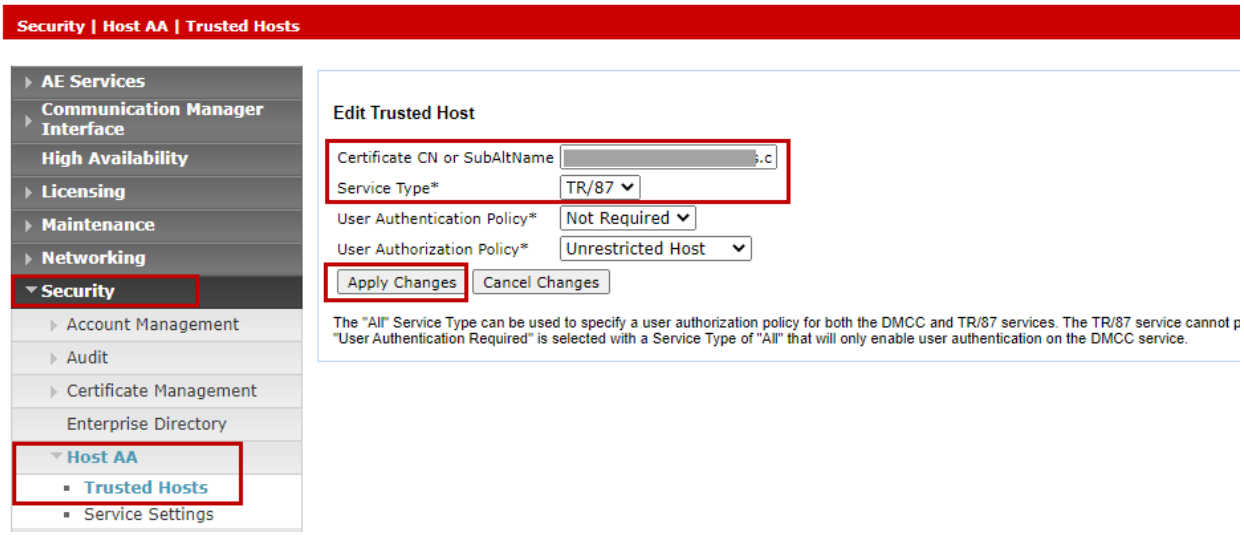



*Figure 129 Adding Avaya Aura CC as a trusted host*

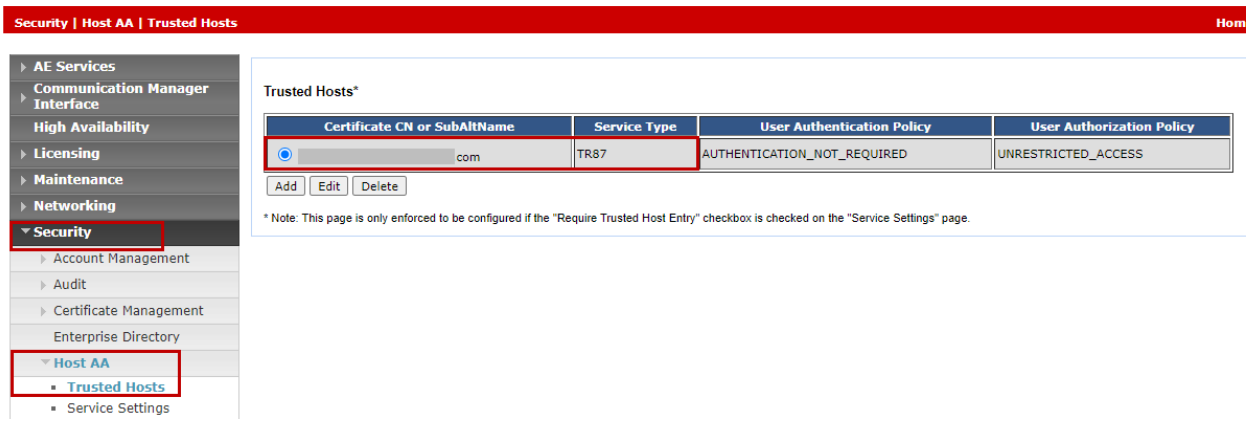



*Figure 130 Adding Avaya Aura CC as a Trusted host Continuation*

### 1.4.11 Certificate Store in Avaya Aura CC

- Login to windows server where Avaya Aura CC is installed
- Search for Security Manager and Click **Security Manager**


*Figure 131 Security Manager*

- *Enter in the security store password*: Type the security store **password**

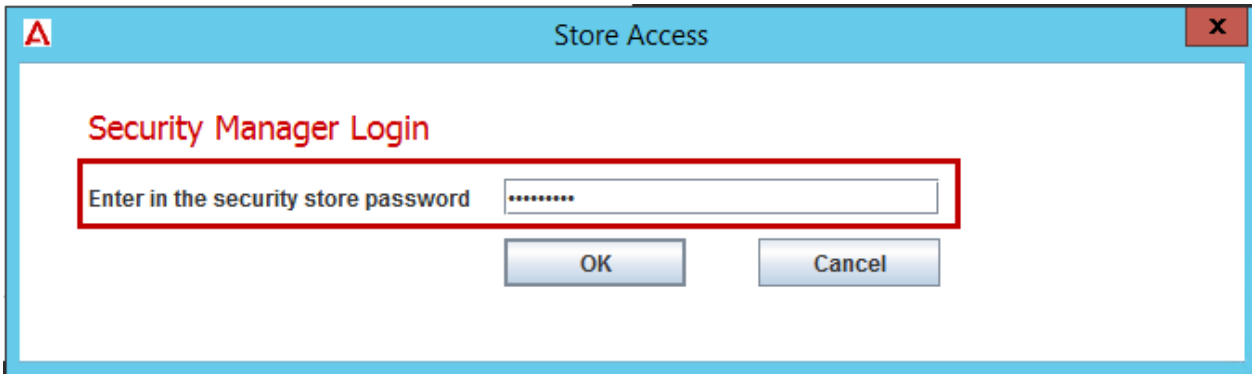



*Figure 132 Security Manager Login*

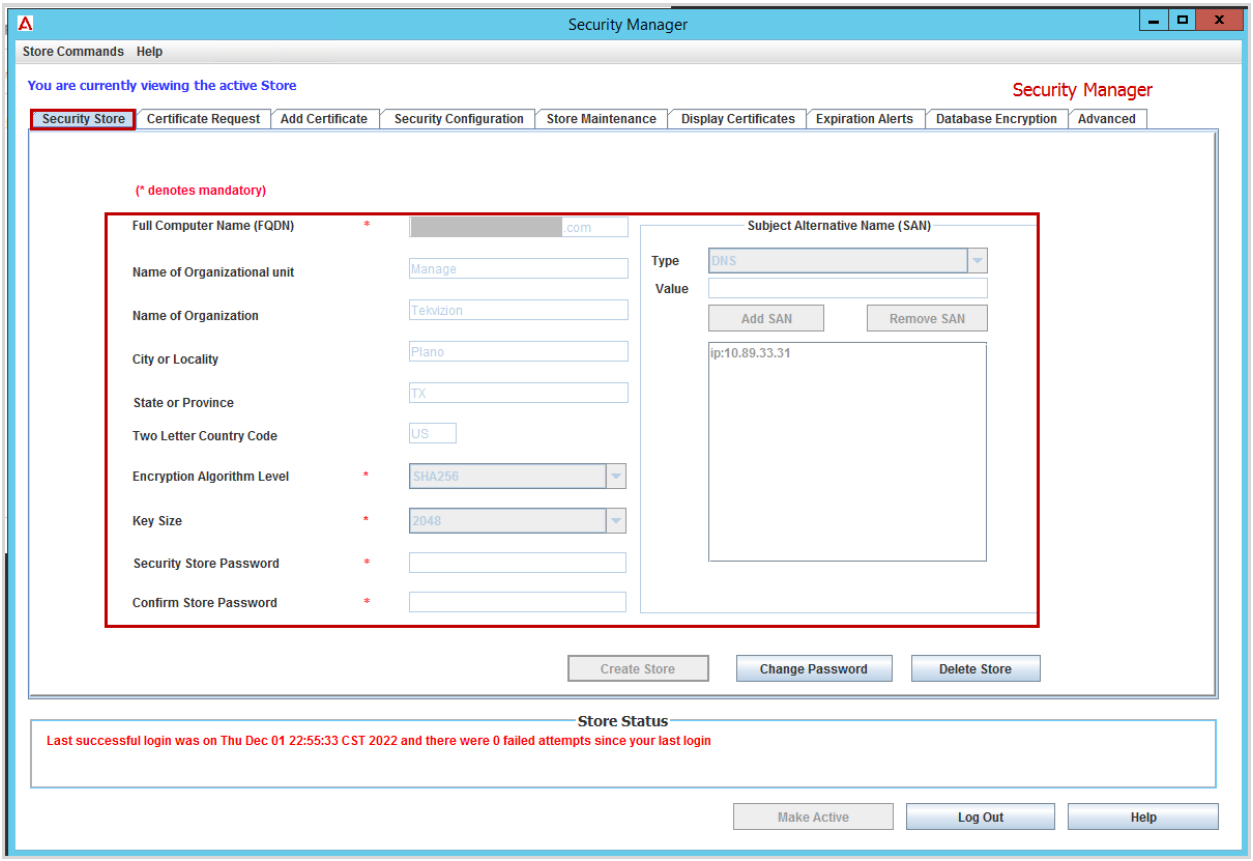



*Figure 133 Security Manager – Security Store*

- Navigate to **Certificate Request** and copy the contents of CSR text
- Get the CSR signed form Certificate Authority

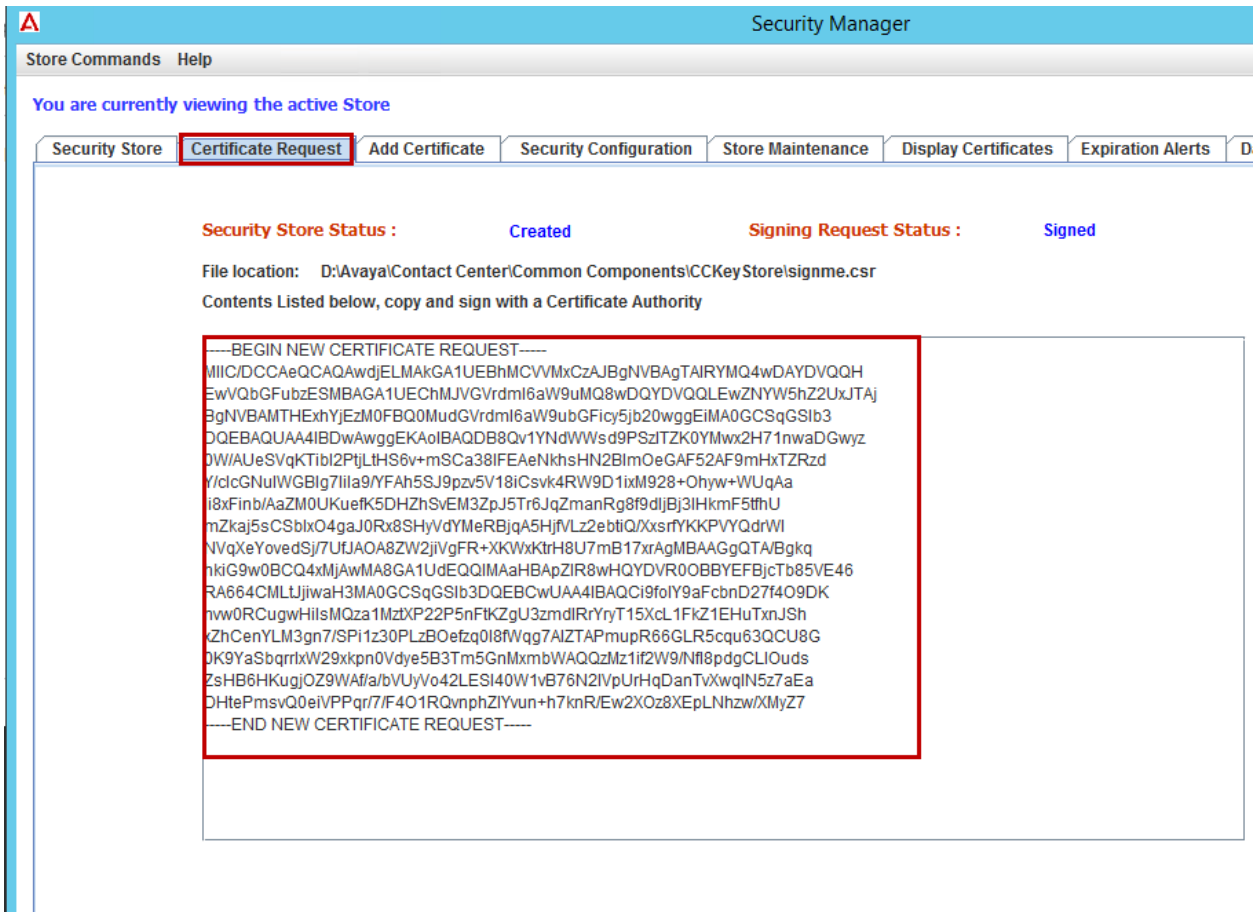



*Figure 134 Security Manager – Certificate Request*

- Navigate to **Add Certificate**
- Under **Add Certificates Automatically (Auto detects Identity, Root, PKCs#12 and p7b certificates),**
    - *Select Folder*: Click **Browse**
    - Navigate to the folder where Root CA and Server certificates are stored
    - *Certificates*: **AvayaCCsignedwithClientandserverAuth.cer, RootCA.p7b** (Root and Server Certificates)
- Click **Add all Certificates**

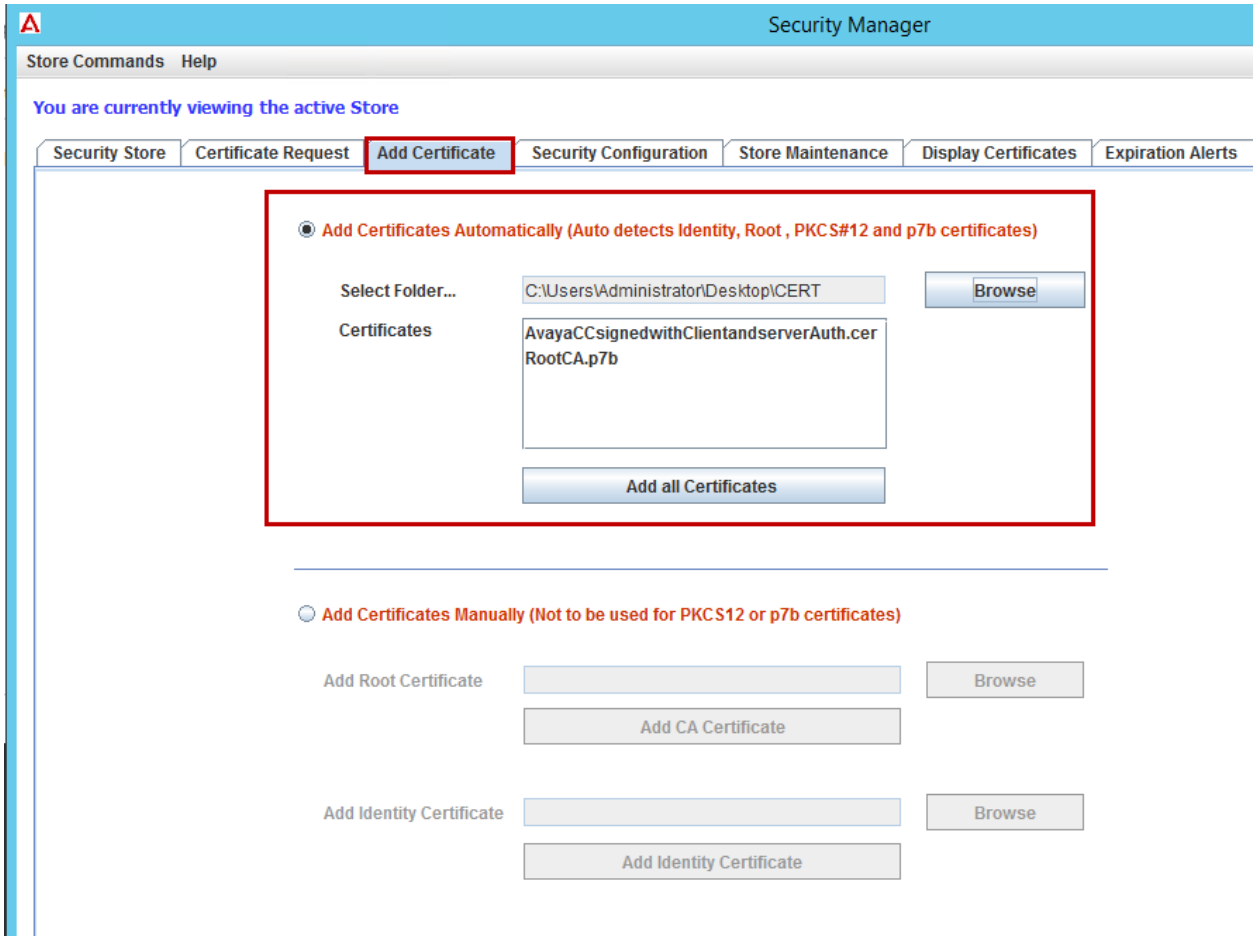



*Figure 135 Security Manager – Add Certificate*

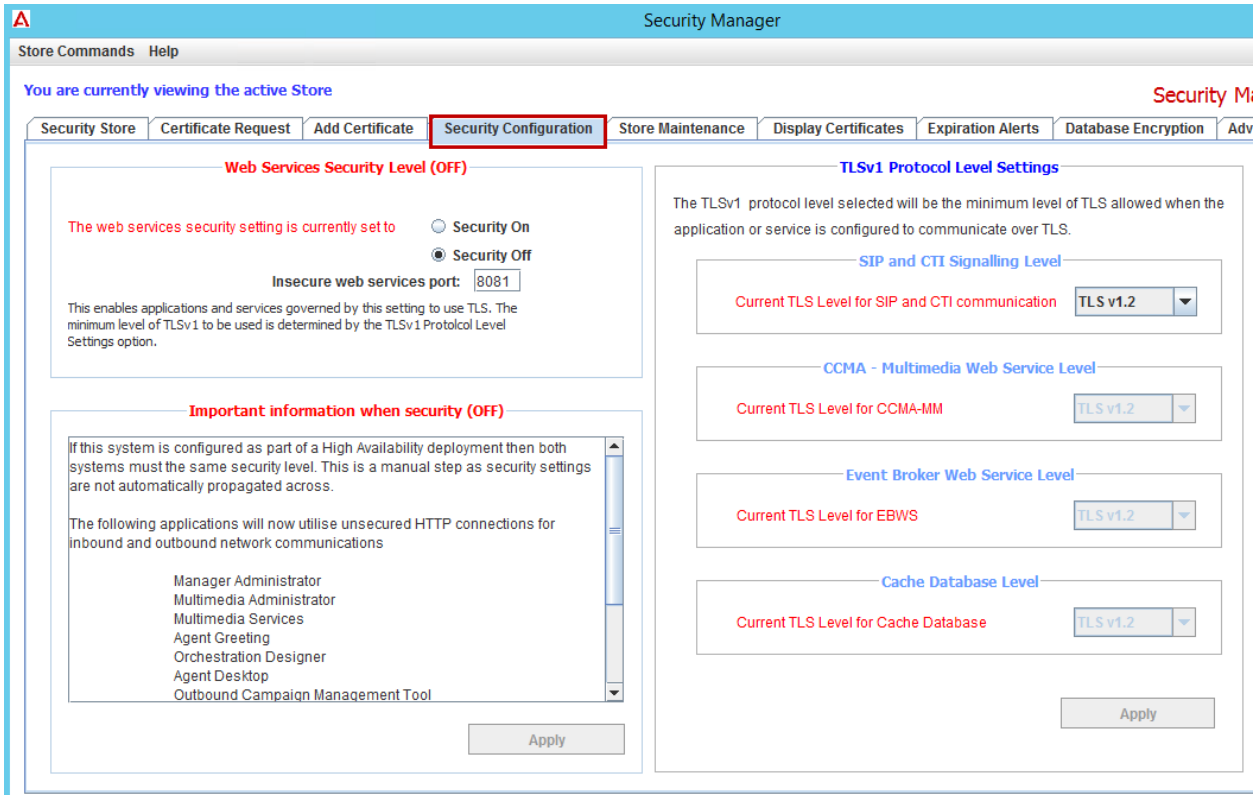



*Figure 136 Security Manager – Security Configuration*

- Navigate to **Store Maintenance**
- Under **Identity and Root Certificates that reside in the store**, the root and server Certificates that are added through **Add Certificate** tab are displayed

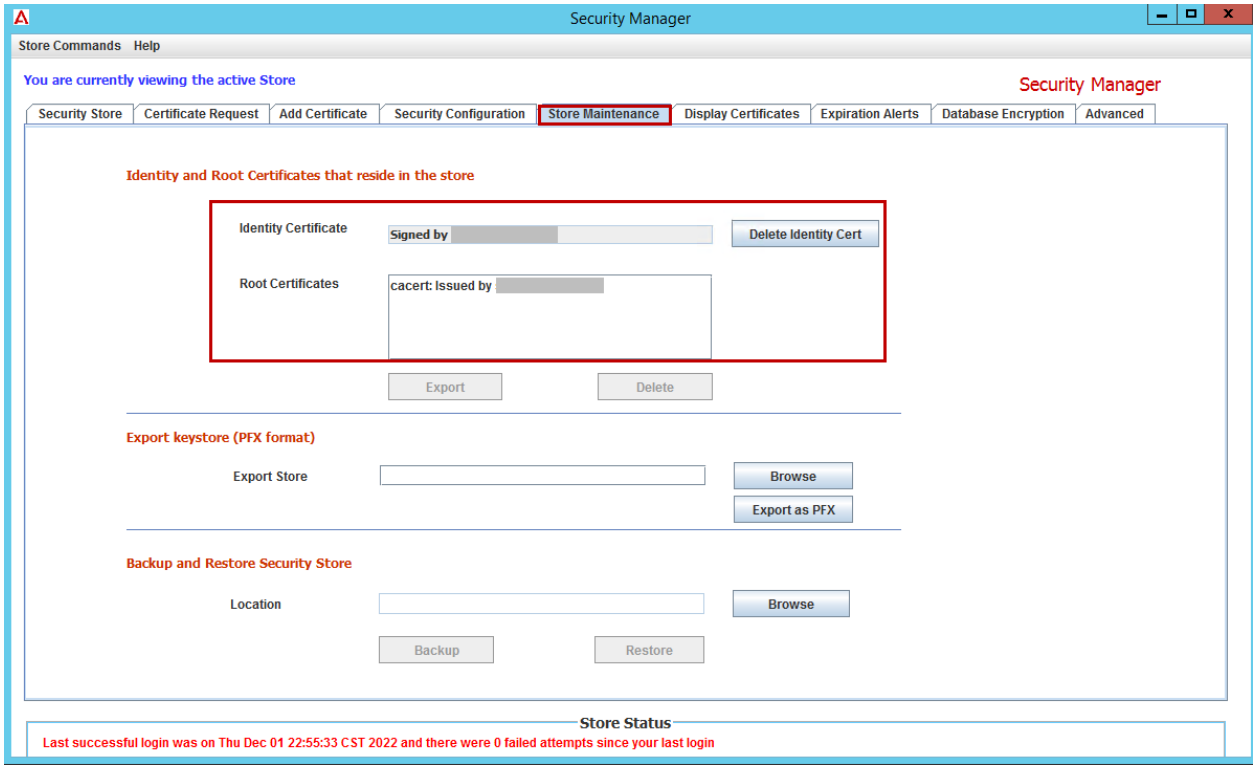



*Figure 137 Security Manager – Store Maintenance*

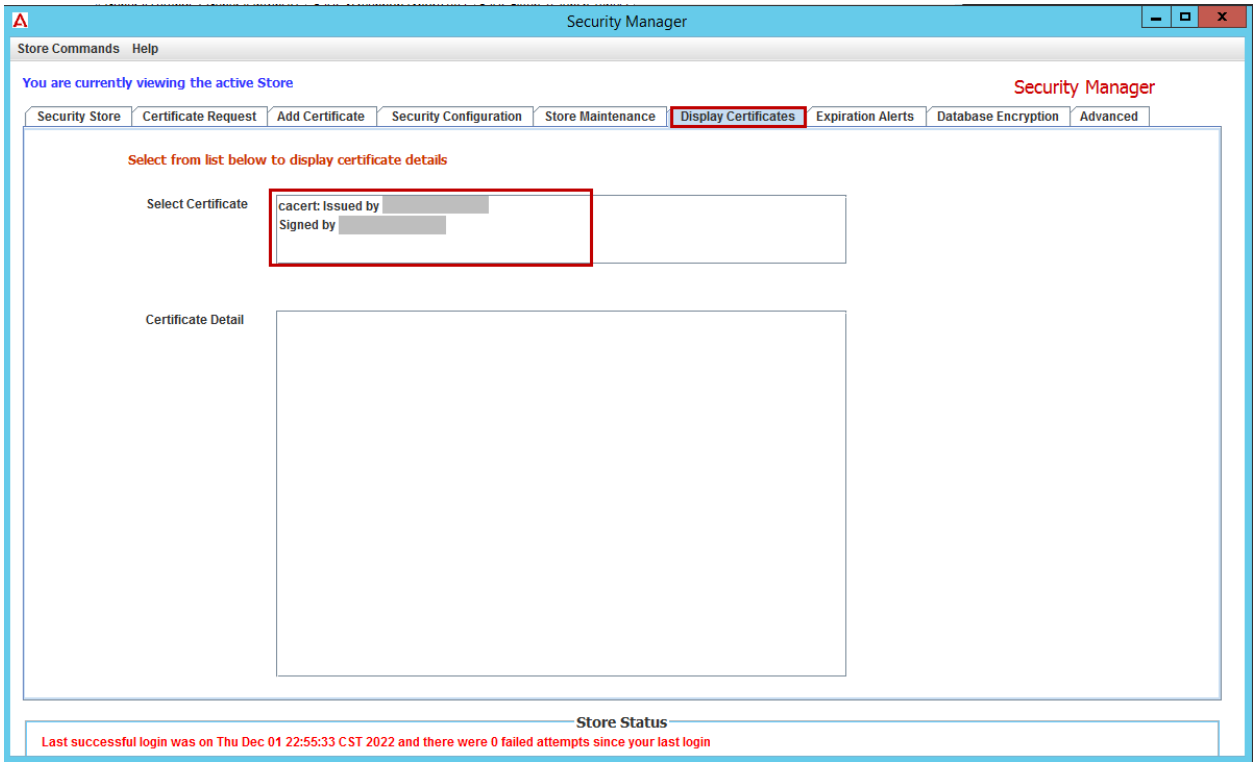



*Figure 138 Security Manager – Display Certificates*

## 1.5 Avaya Aura Media Server (AAMS) Configuration

- Access Avaya Aura MS Web login screen via **https://<Avaya Aura Media Server IP>**
- Use admin as **User ID** and associated **Password**
- Click **Sign In**

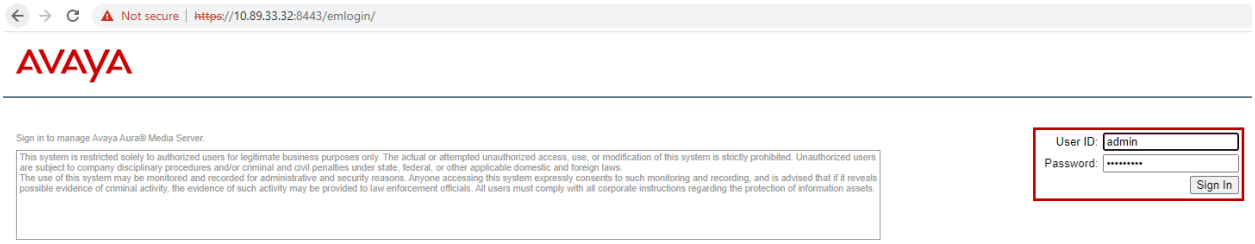



*Figure 139 Avaya Aura Media Server Login*

### 1.5.1 Nodes and Routes

- Navigate to **System Configuration > Signaling Protocols > SIP > Nodes and Routes**
- Under **Trusted Nodes**,
  - Click **Add**
- Under **Add SIP Trusted Node**,
  - *Host or Server Address*: **10.89.33.31** (Avaya Aura Contact Center Server IP)

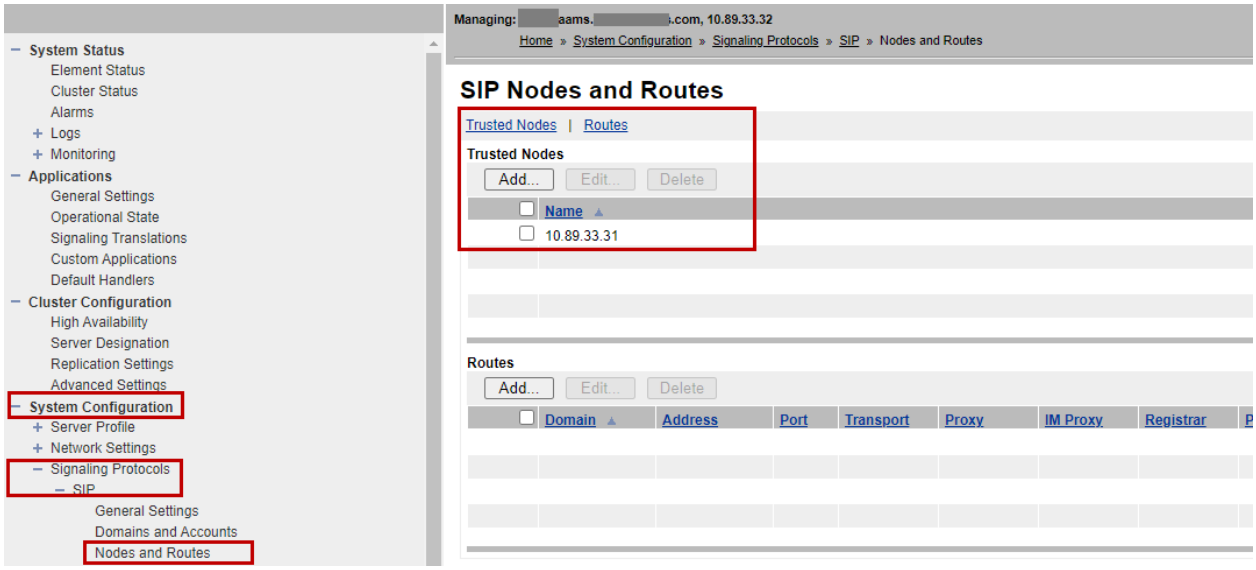



*Figure 140 Avaya Aura Media Server – Add Trusted Node*

## 1.6 Avaya Aura Contact Center Configuration

### 1.6.1 Avaya Aura Contact Center Login

- Login to Windows machine where Avaya Aura Contact Center is installed

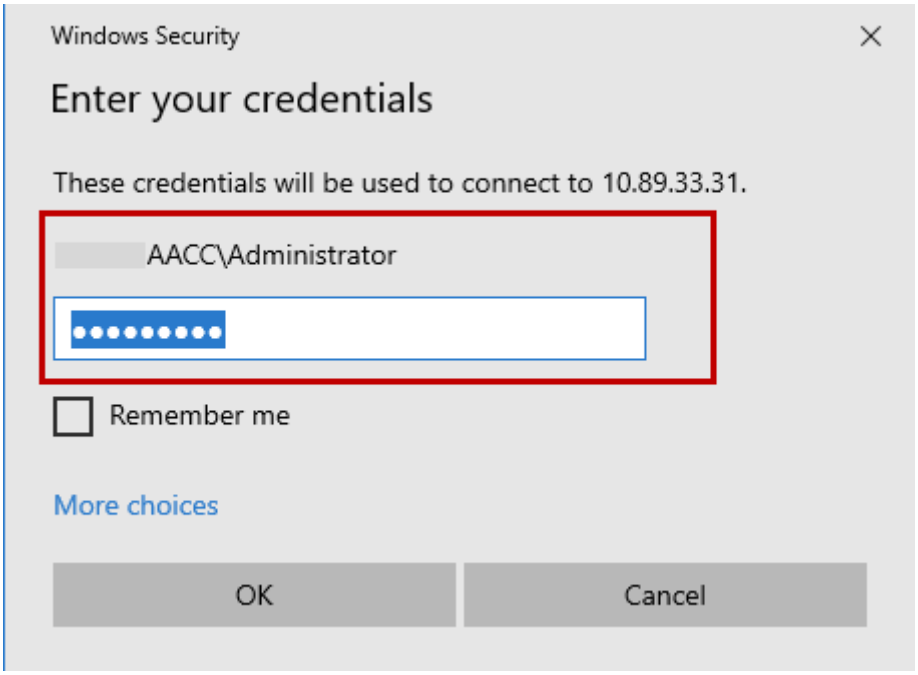



*Figure 141 Avaya Aura CC Login*

### 1.6.2 System Control and Monitor Utility

- Navigate to Windows search and type System Control and Monitor Utility
- Click **System Control and Monitor Utility**
- Ensure appropriate Contact Center Services are started in System Control and Monitor Utility

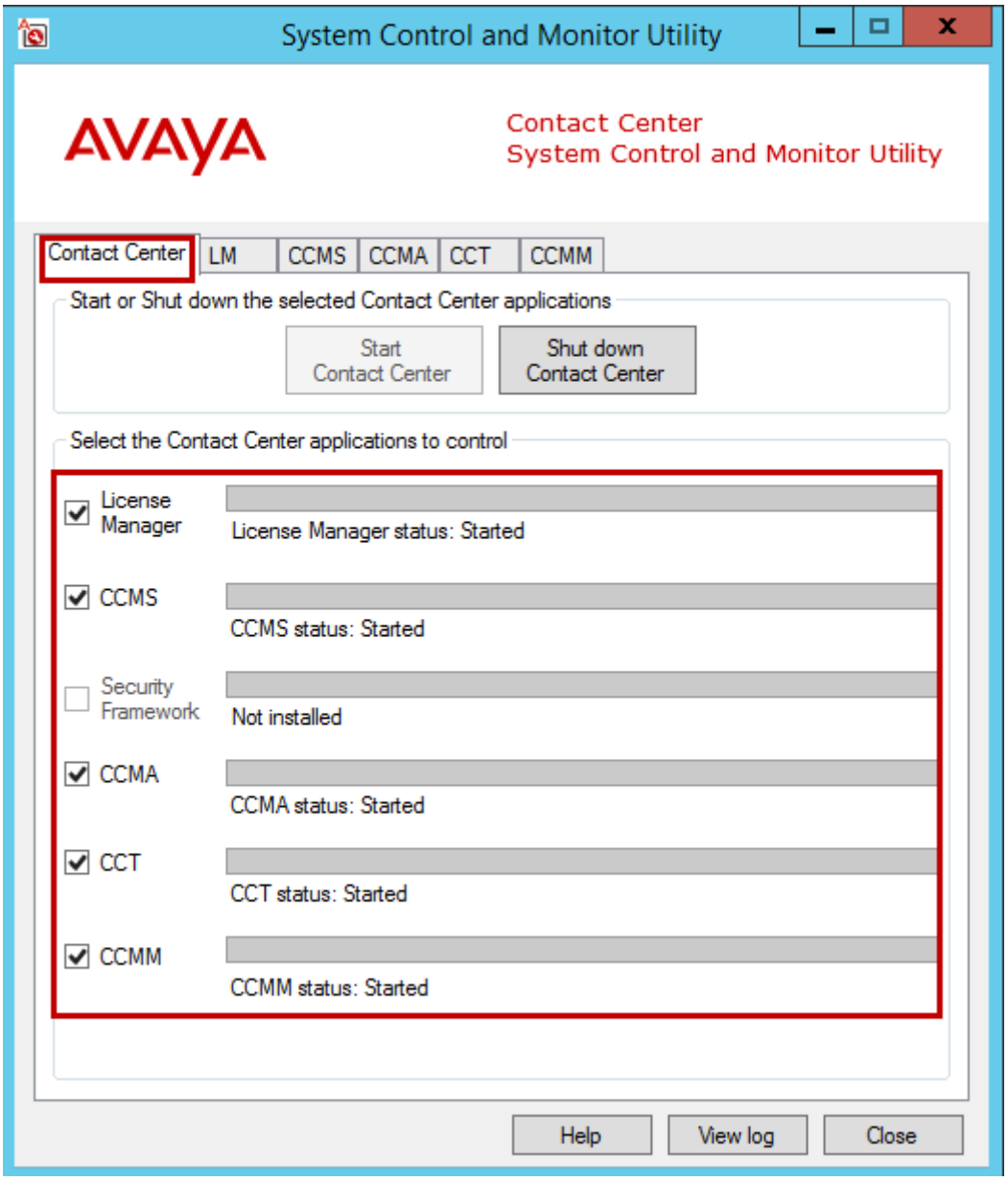



*Figure 142 System Control and Monitor Utility*

### 1.6.3 Server Configuration

- Navigate to Windows search and type Server Configuration
- Click **Server Configuration**
- Navigate to **Main Menu > Local Settings**
- Below are the default settings under **Local Settings**

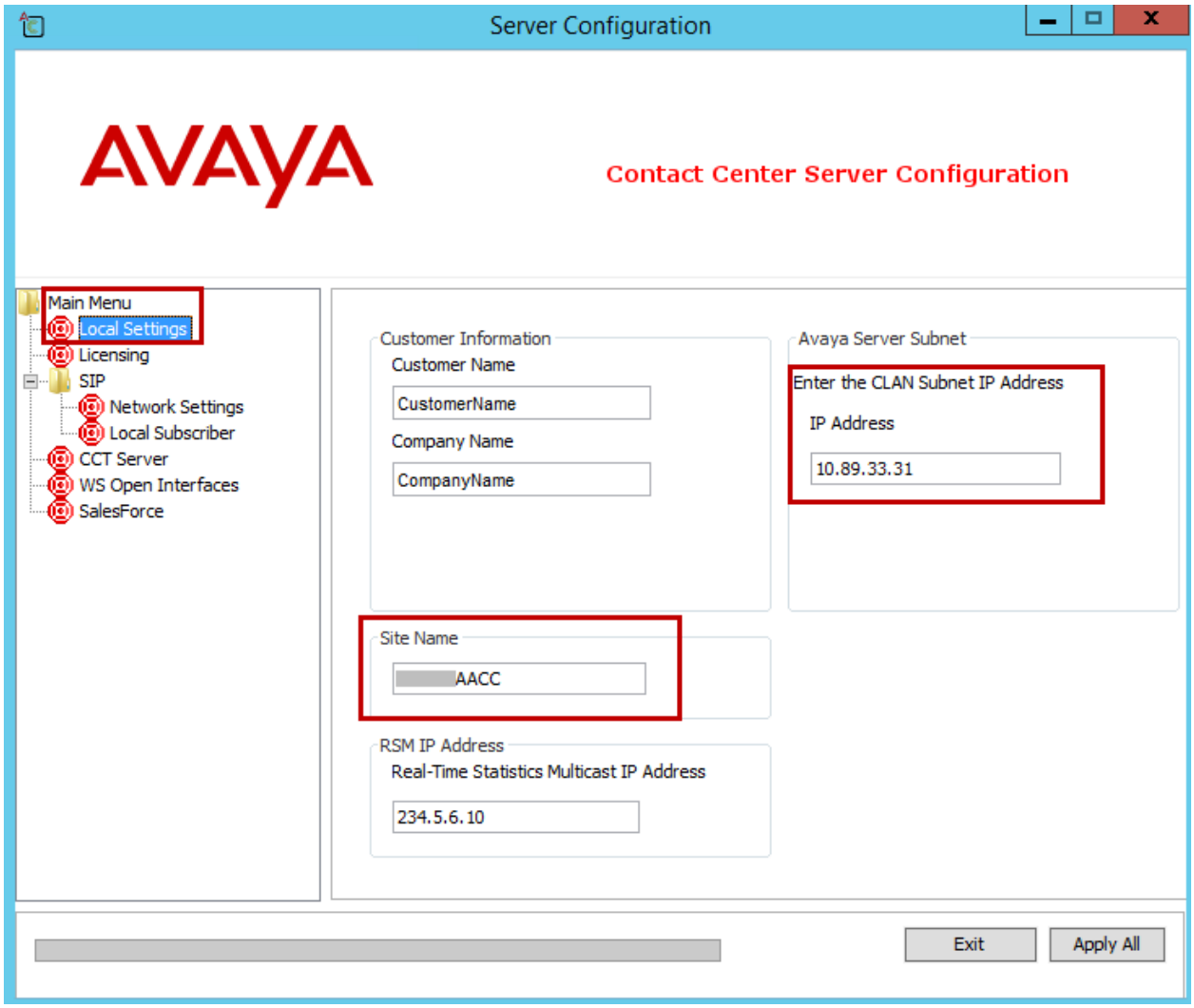



*Figure 143 Server Configuration – Local Settings*

- Navigate to **Main Menu > Licensing**
- Below are the default settings under **Licensing**

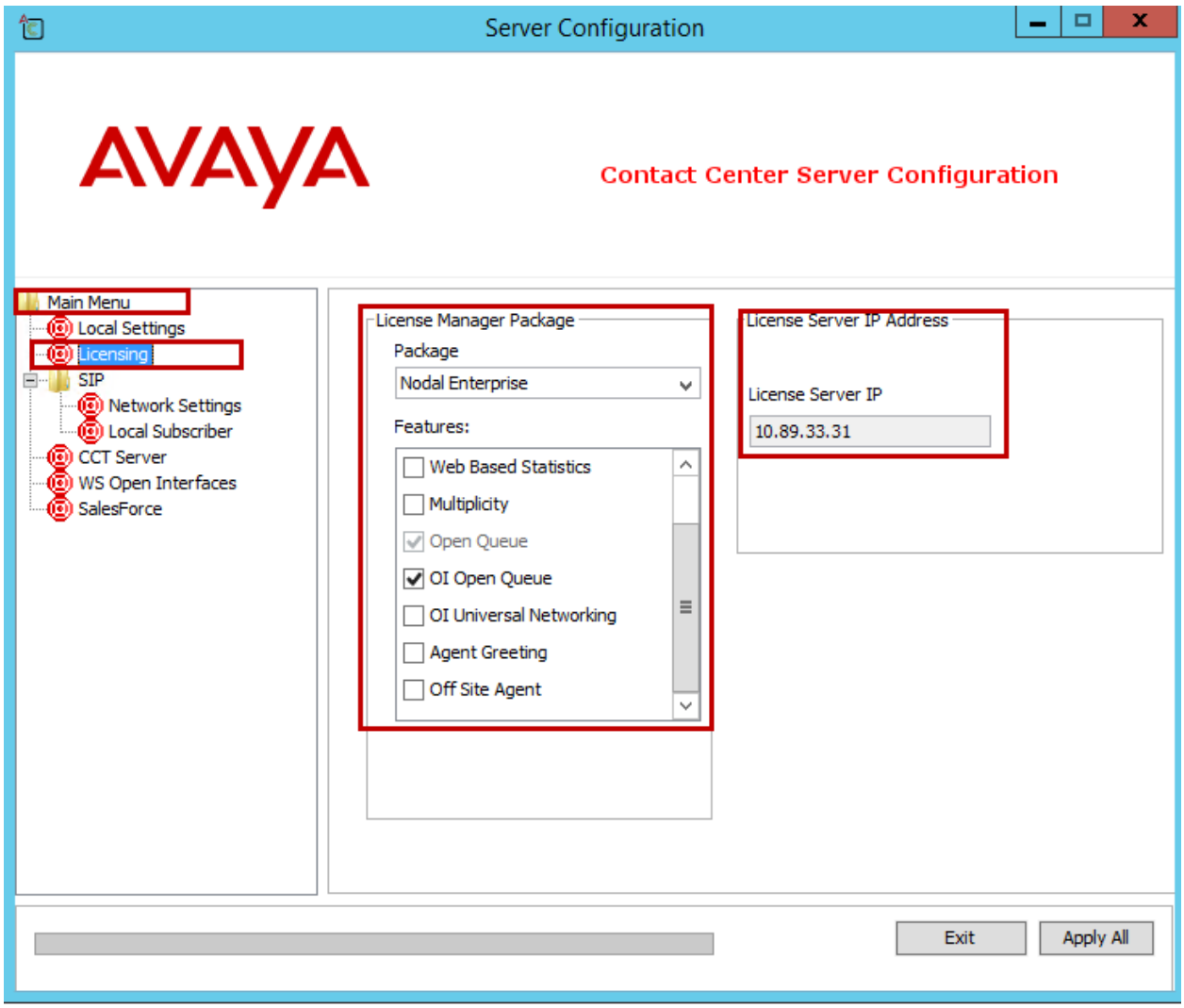



*Figure 144 Server Configuration – Licensing*

- Navigate to **Main Menu > SIP > Network Settings**
- Under **SIP Network Settings**,
    - *Voice Proxy Server*: Enter the IP Address of **Avaya Aura SM**
    - *Port*: **5062**
    - *Transport*: **TCP**
    - *CTI Proxy Server*: Enter the IP address of **Avaya AES server**
    - *Port*: **4723**
    - *Transport*: **TLS**

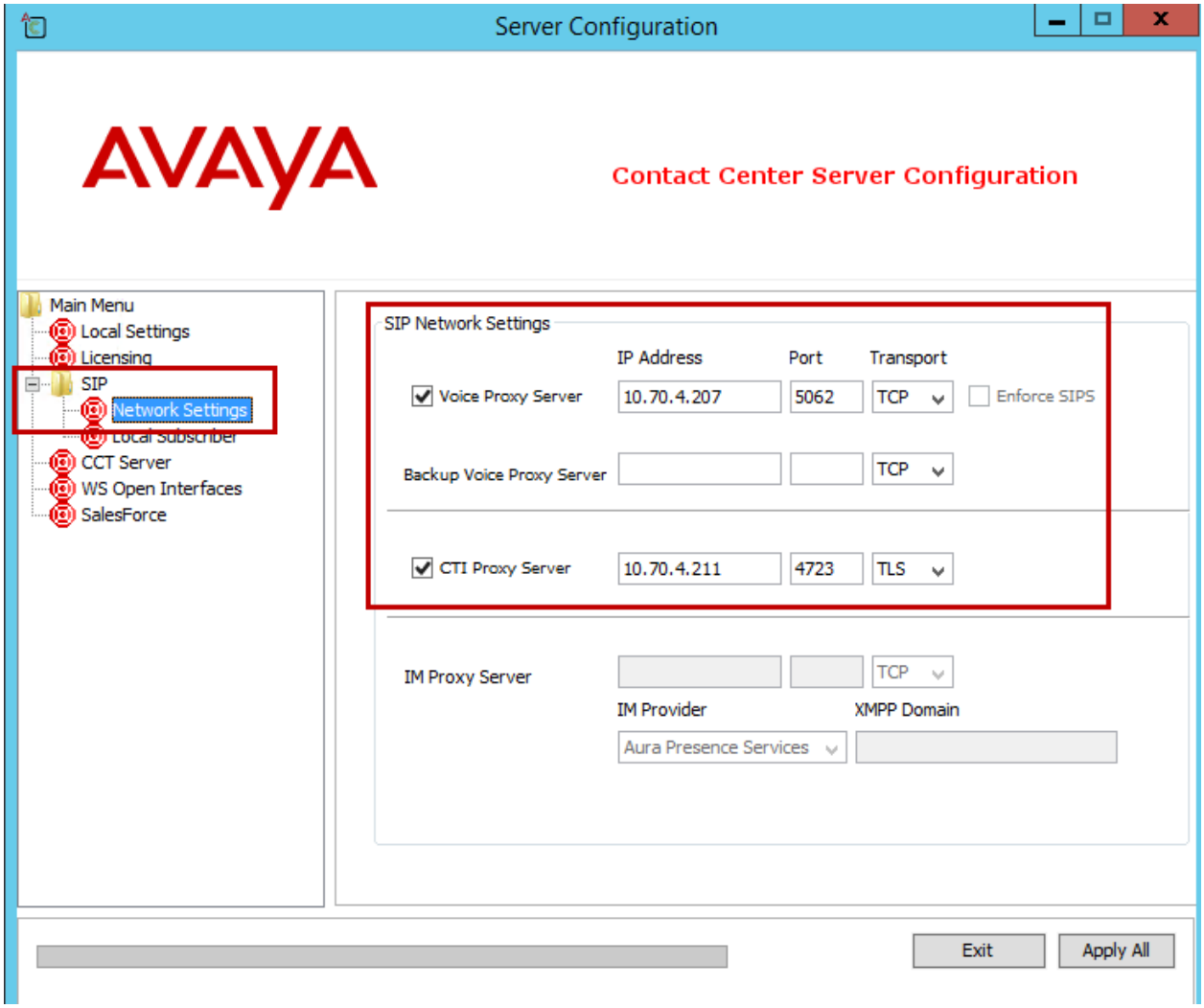



*Figure 145 Server Configuration – SIP Network Settings*

- Navigate to **Main Menu > SIP > Local Subscriber**
- Below are the default settings under **Local Subscriber**

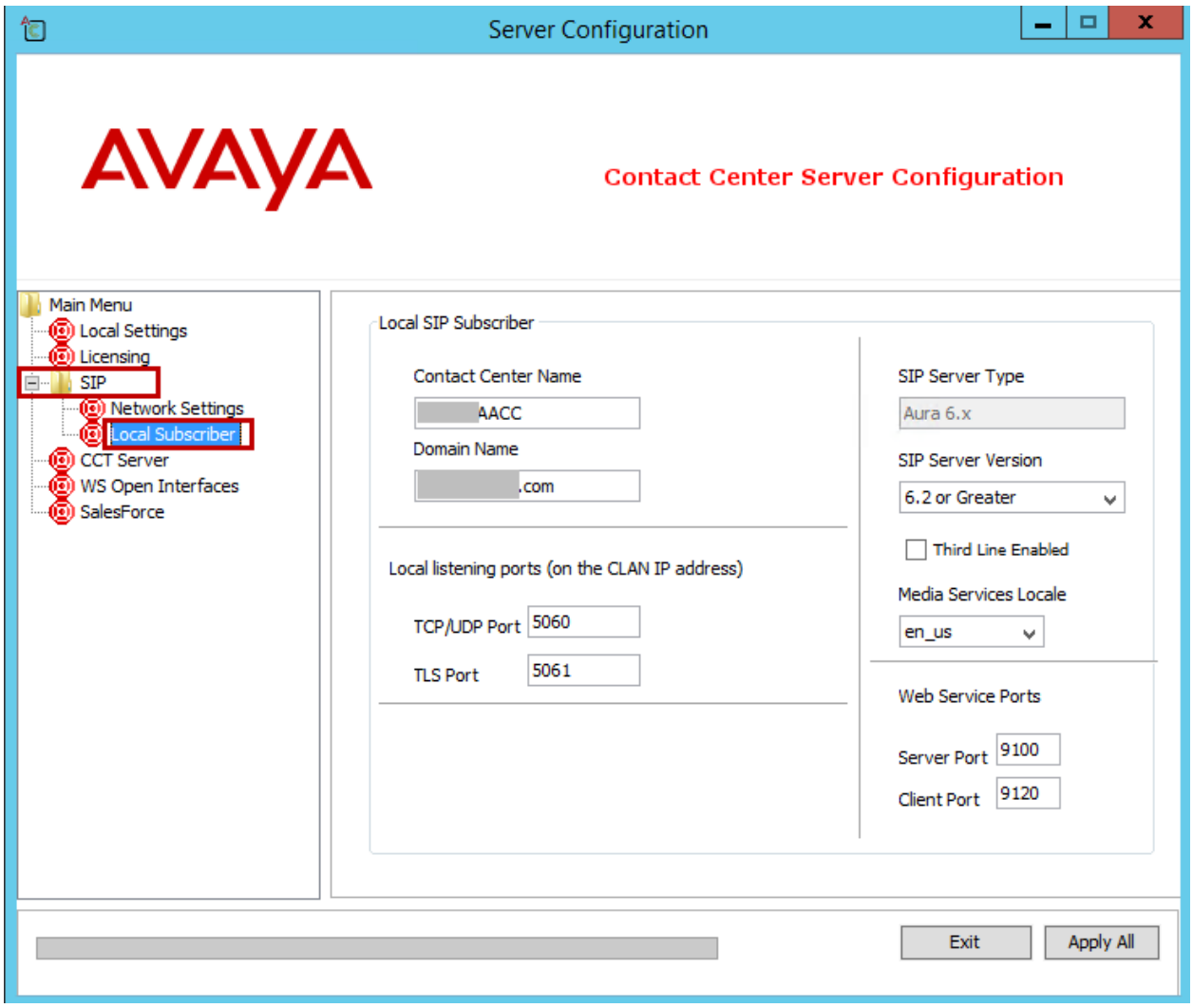



*Figure 146 Server Configuration – SIP Local Subscriber*

- Navigate to **Main Menu > CCT Server**
- Below are the default settings under **CCT Server**

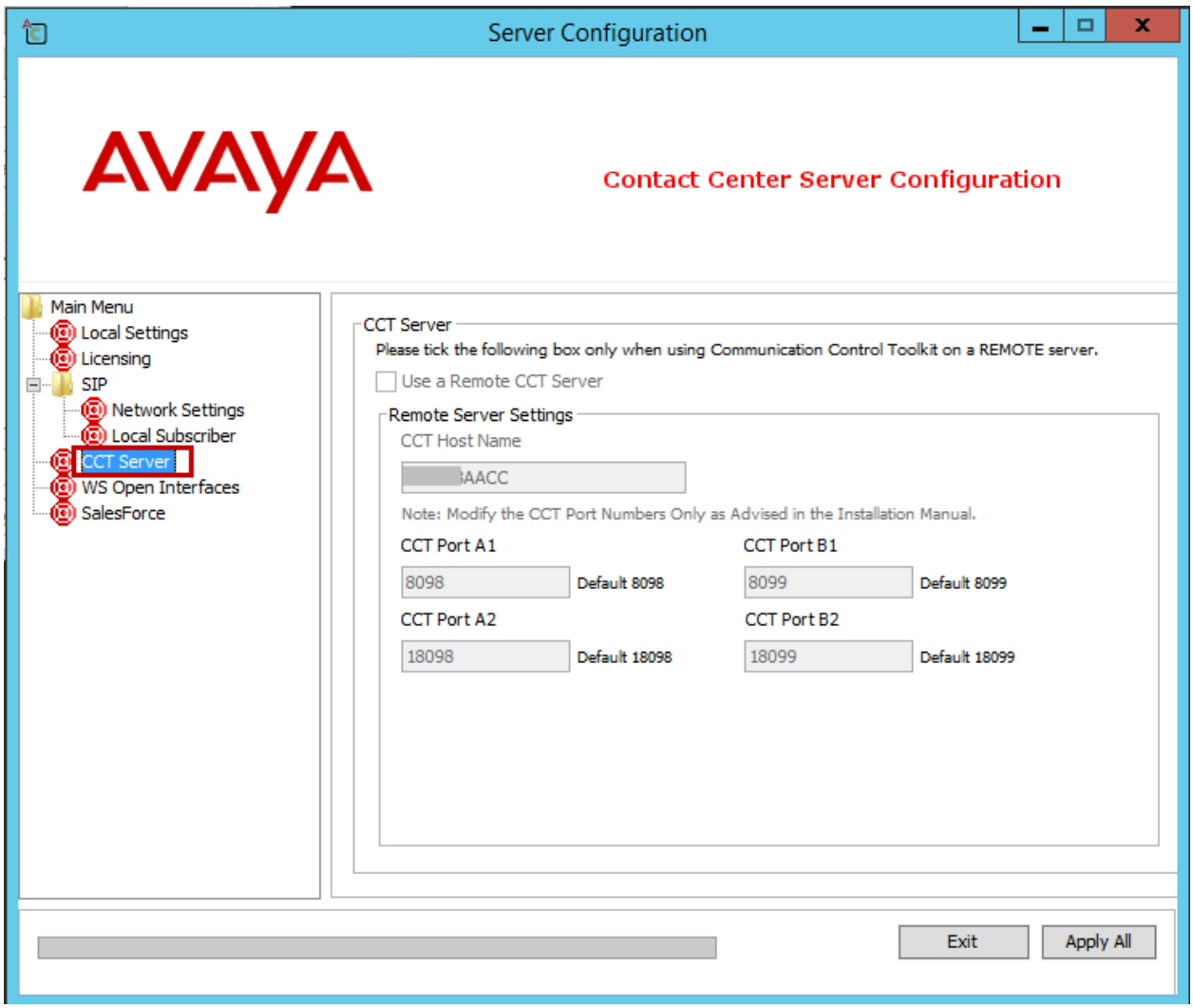



*Figure 147 Server Configuration – CCT Server*

### 1.6.4 SGM Management Client

- Navigate to Windows search and type SGM Management Client
- Ensure **Voice Outbound Proxy, CTI Proxy and Media Server(s)** state displays Connected

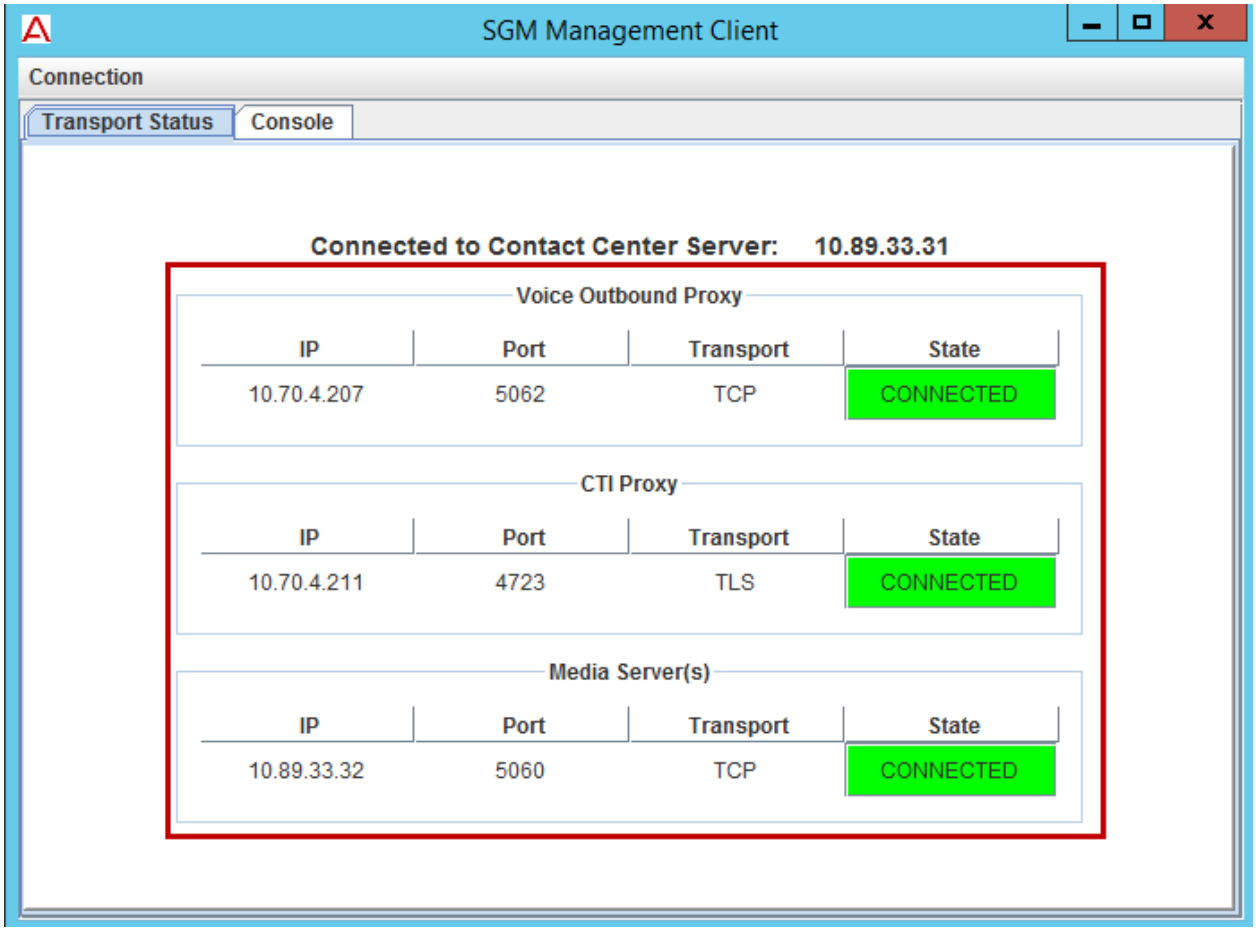



*Figure 148 SGM Management Client*

### 1.6.5 Avaya Aura Contact Center (CC) Manager Configuration

- Access Avaya Aura CC Web login screen via **http://<Avaya Aura CC IP Address>**
- Use webadmin as **User ID** and associated **Password**
- Click **Login**

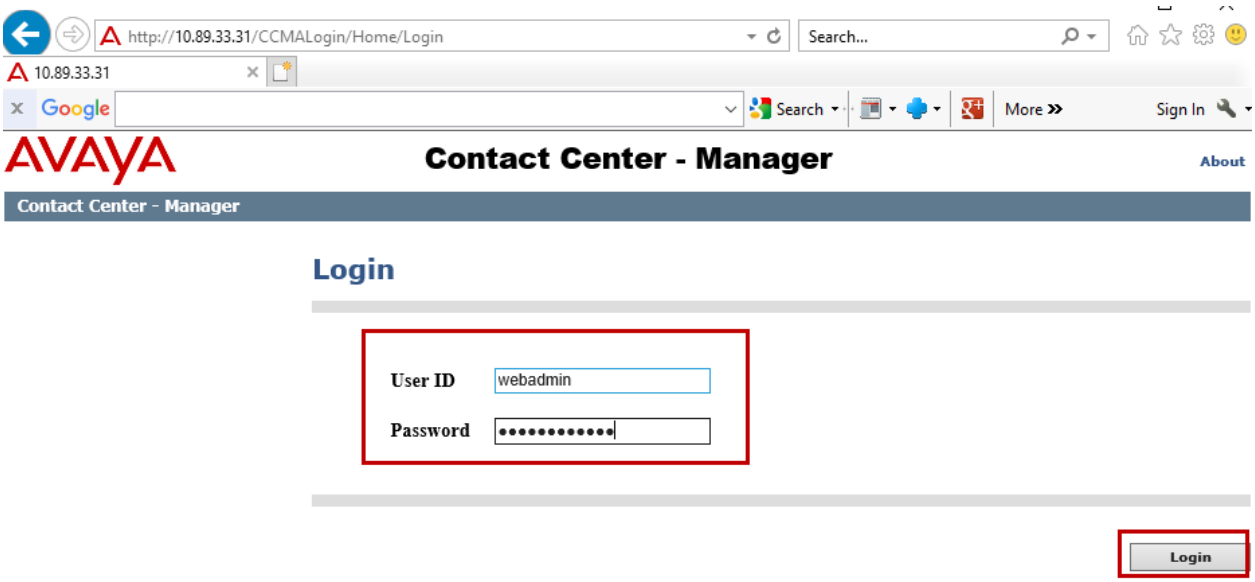



*Figure 149 Avaya Aura CC Manager Login*

### 1.6.6 CDNs (Route Point) Configuration

- Navigate to **Launchpad > Configuration**

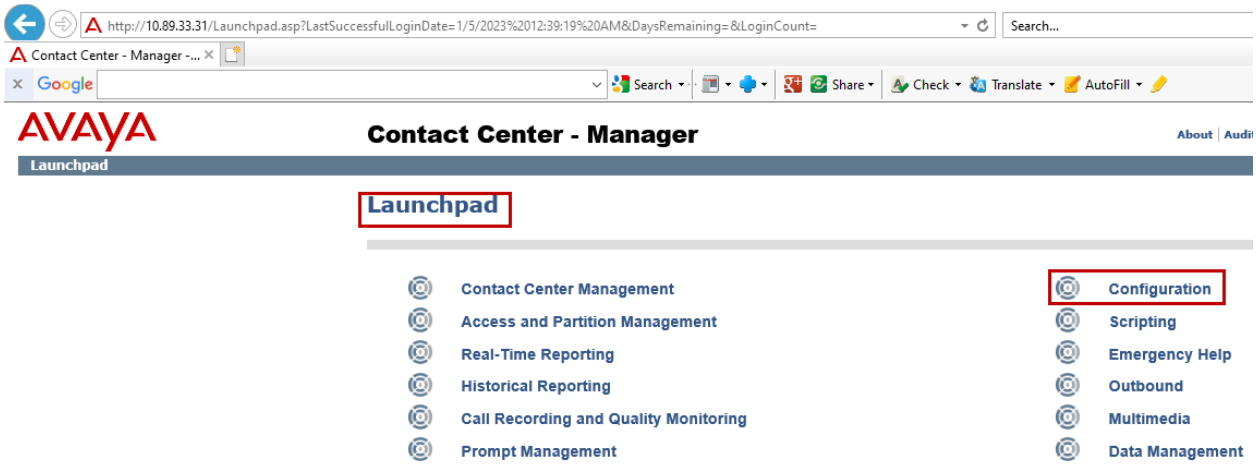



*Figure 150 Launchpad*

- Navigate to **AACC > CDNs (Route Points)**
- *Name*: **7500**
- *Number*: **7500**
- *URI*: **sip:7500@<Avaya Aura CC domain name>.com**
- *Call Type*: **Local**
- *Acquired*: **Enabled**

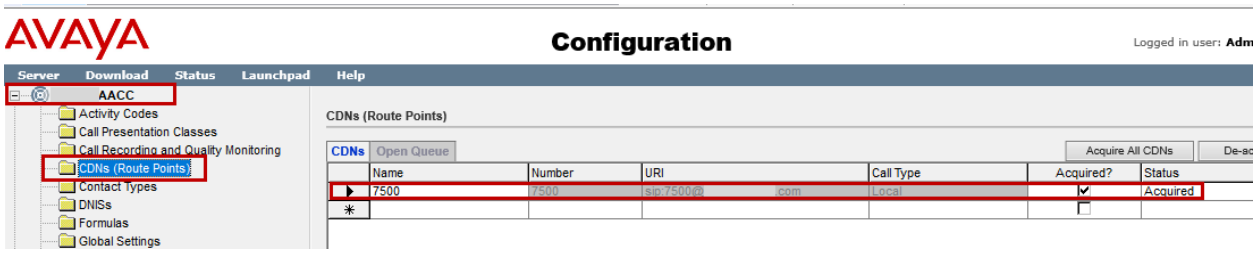



*Figure 151 CDNs (Route Point) configuration*

### 1.6.7 Media Server

- Navigate to **AACC > Media Servers**
- *Name*: **Enter the host name of Avaya Aura Media Server (e.g. AAMS)**
- *IP Address*: **10.89.33.32 (IP address of Avaya Aura Media Server)**
- *Transport*: **TCP**
- *Master Content Store*: **Enabled**

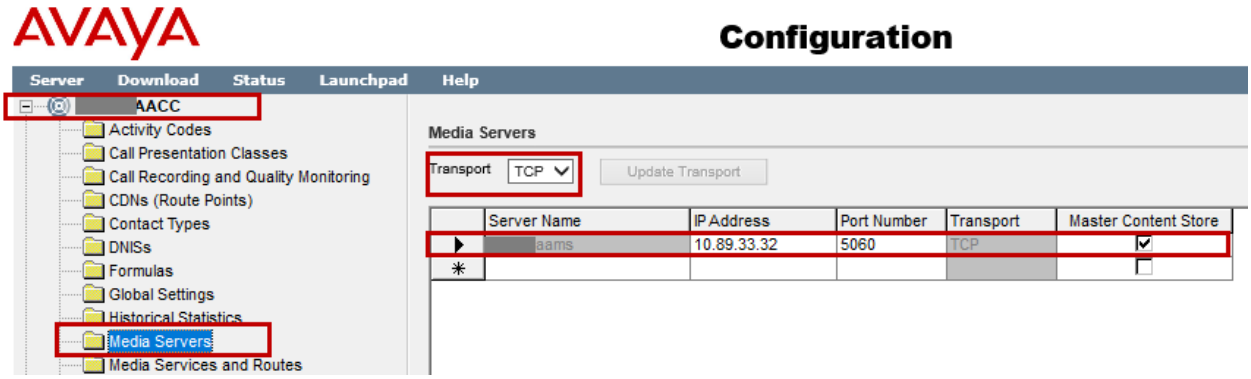



*Figure 152 Add Media Server*

### 1.6.8 Media Server Configuration

- Navigate to **AACC > Media Services and Routes**
- Under **Media Services and Routes**,
  - *Service Name*: Ensure for each of the Service Names (e.g. **AAC_APP_ID, CONF, MDIALOG**) with Treatment address (e.g. **sip_sip-conf@<domain name>.com**), the media server e.g. **AAMS** is associated
  - *Routes (Target Media Servers)*: Select a row from the service name (e.g. **ACC_APP_ID**) and move the media server e.g. **AAMS** from **Available** to **Selected** column
- Repeat the same step for Service Names **CONF** and **MDIALOG**
- Click **Submit**

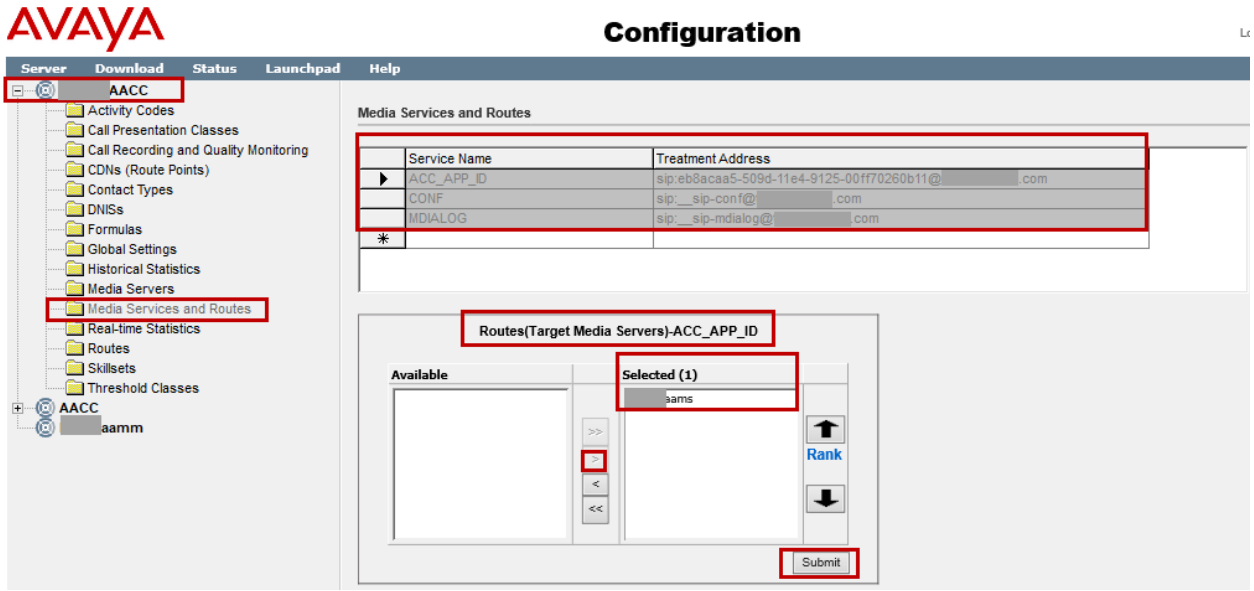



*Figure 153 Media Server Configuration*

### 1.6.9 CCT Server

- Navigate to **Launchpad > Configuration**

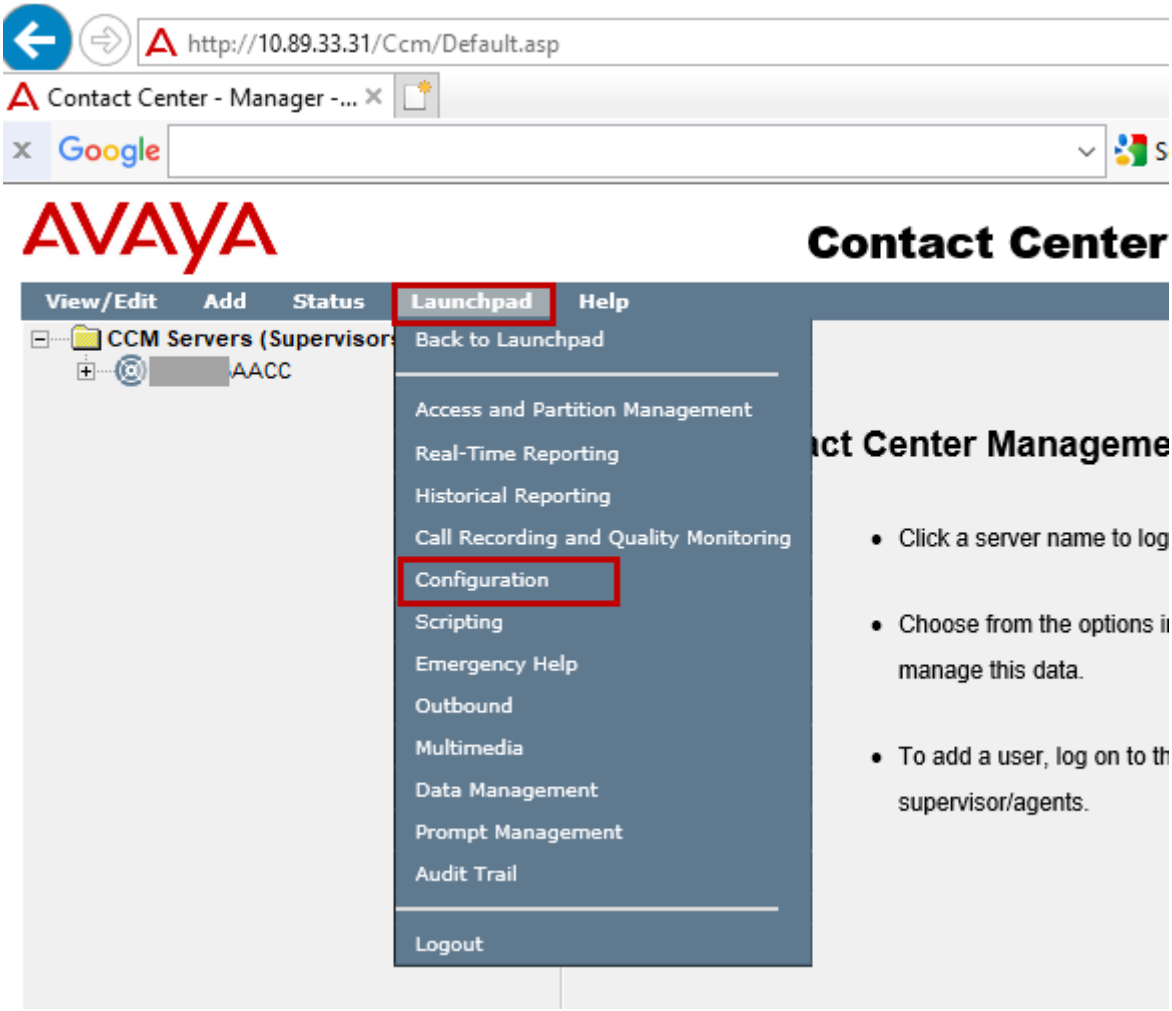



*Figure 154 Add CCT Server*

- Navigate to **Server > Add Server**
- Under **Server Properties**,
    - *Type*: **CCT**
    - *Server Name*: **AACC** (Host name of the Avaya Aura Contact Center Server)
    - *IP Address*: **10.89.33.31** (Avaya Aura Contact Center Server IP populates automatically)
    - *Display name*: **AACC**
    - *Port Number*: **8081**
    - *Associated CCMS Servers*: **AACC** (Select the Avaya Aura Contact Center server radio button)
- Click **Submit**

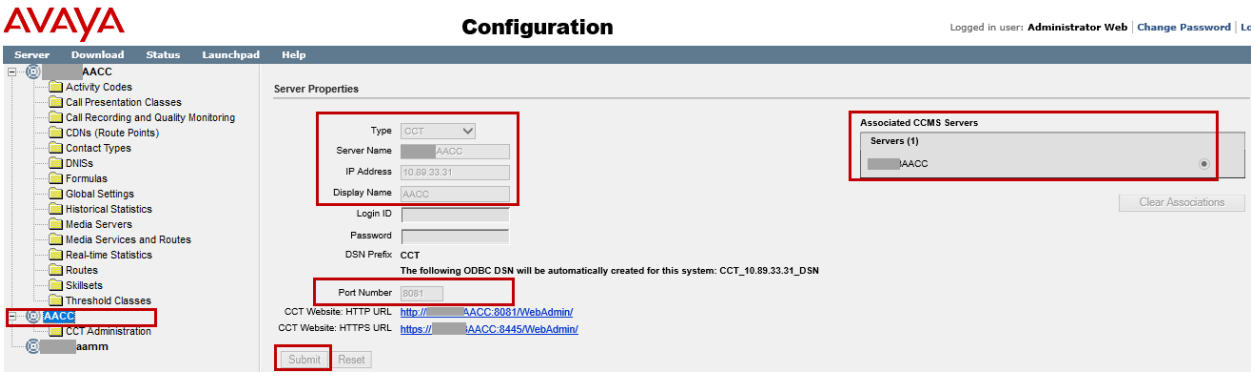



*Figure 155 Add CCT Server Continuation*

### 1.6.10 Contact Center Agent

- Navigate to **Launchpad > Contact Center Management > CCM Servers (Supervisors) > AACC > Supervisor Default > Right Click > Add Agent**
- Under **User Details**,
  - *User Type*: **Agent**
  - *First Name*: **seetha**
  - *Last Name*: **agent1**
  - *Login ID*: **3001**
  - *Voice URI:* **sip:3001@<Avaya Aura CC domain name>.com**
- Under **Account Type**,
  - *Create CCT Agent*: **Enabled**

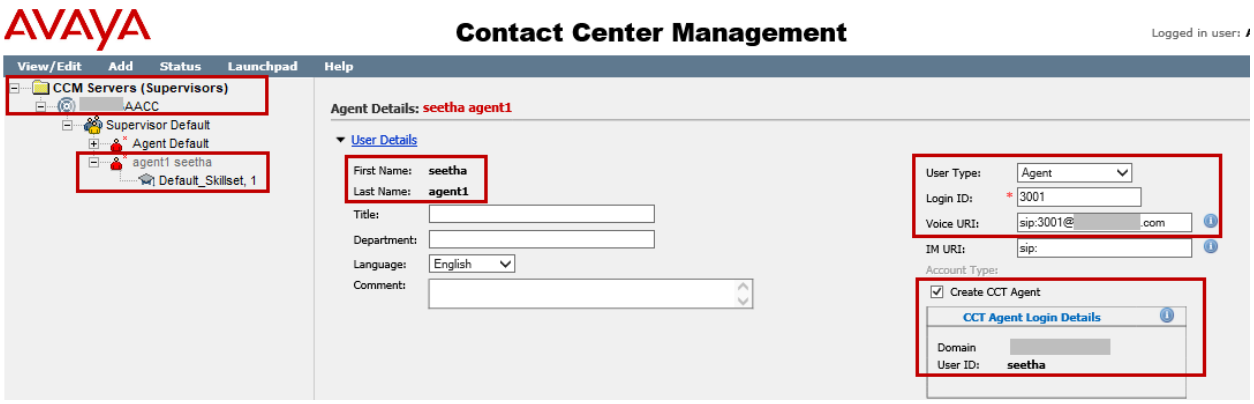



*Figure 156 Contact Center Agent*

- Under **Associate User Account**
  - *Search domain users*: **Enabled**
  - Under **Domain Details**,
  - *Server Name or IP*: **10.85.0.12**
- Under **Domain Account**
  - *(Domain\User ID):* **<Avaya Aura CC domain name>\seetha**
  - *Password*: **Enter the associated user password**
- Under **Search all accounts where**
  - Select **First Name**
  - *Starts with*: **seetha**
  - Click **Search**
- Select the row listed with User Name as **seetha**

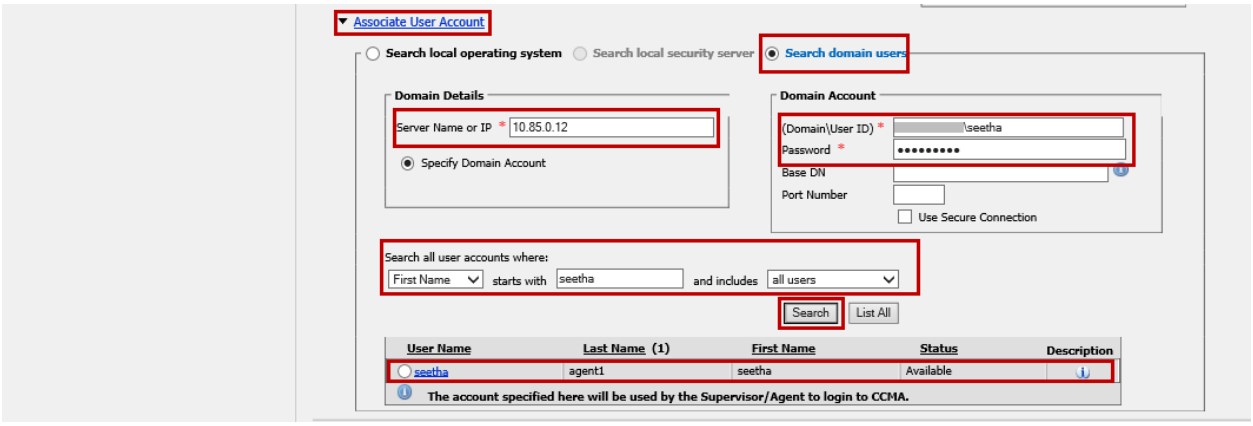



*Figure 157 Contact Center Agent Continuation*

- Under **Contact Types**,
    - *Voice*: **Enabled**

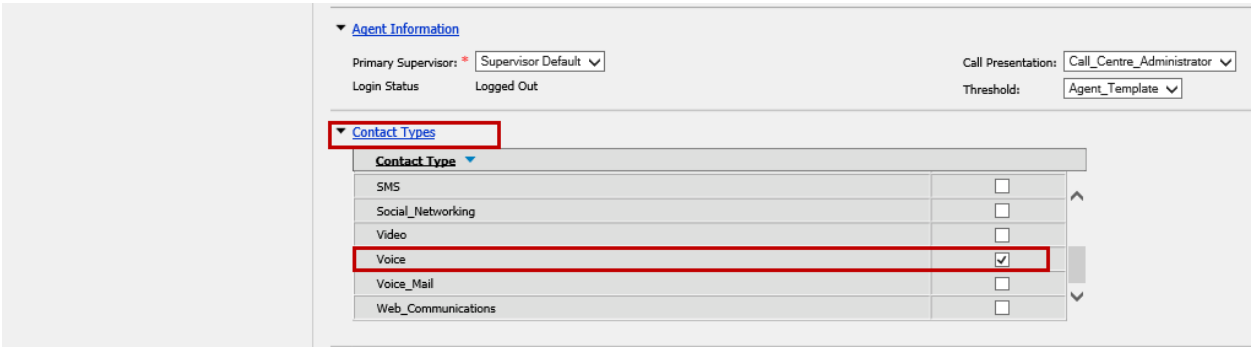



*Figure 158 Contact Center Agent Continuation*

- Under **Skillsets**,
    - *Skillset Name:* **Default_Skillset**
- Click **Submit**

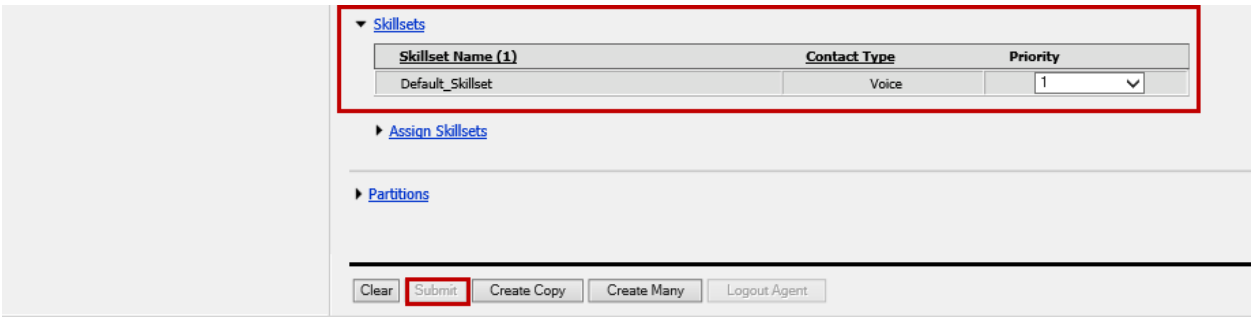



*Figure 159 Contact Center Agent Continuation*

### 1.6.11 TFE REST Configurator

TFE REST Configurator is used to define the Twilio API URL with authorization credentials. The endpoint ID **1** is used by in the Application script (Section 6.7.12 - **CallTwilio_API**) to invoke the Twilio API URL

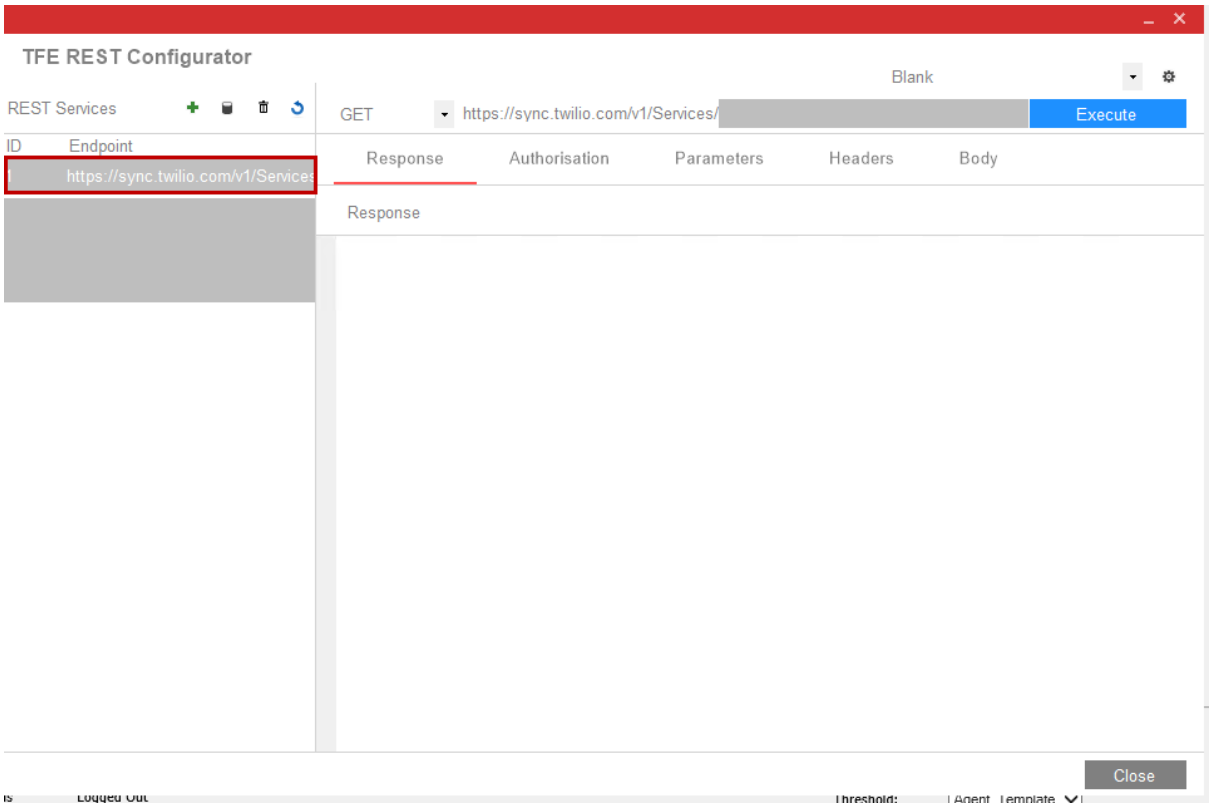



*Figure 160 TFE REST Configurator*

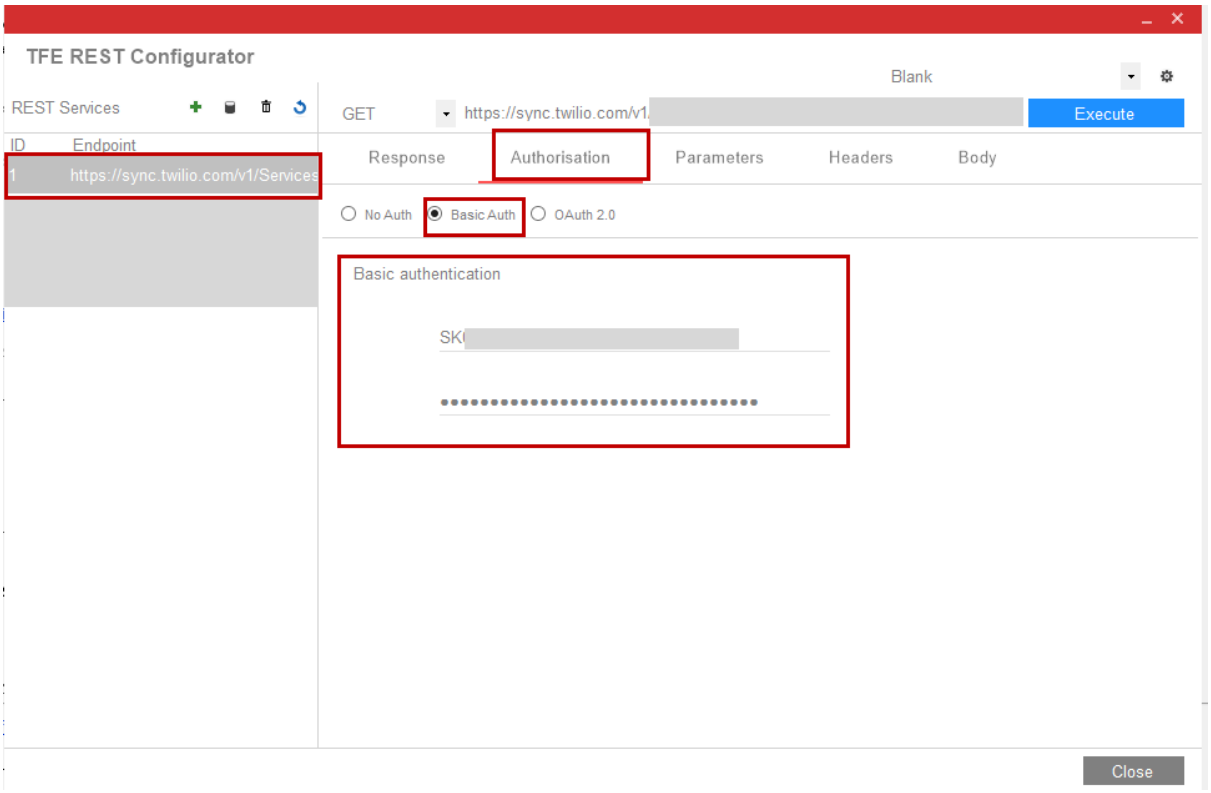



*Figure 161 TFE REST Configurator Continuation*

### 1.6.12 Avaya Aura Orchestration Designer Application Script

Applications contains instructions that determine the sequence of steps that a contact follows after the contact arrives at Avaya Aura Contact Center

- Navigate to **Launchpad > Scripting > Orchestration Designer > Launch Orchestration Designer**
- **Orchestration Designer** tool is launched

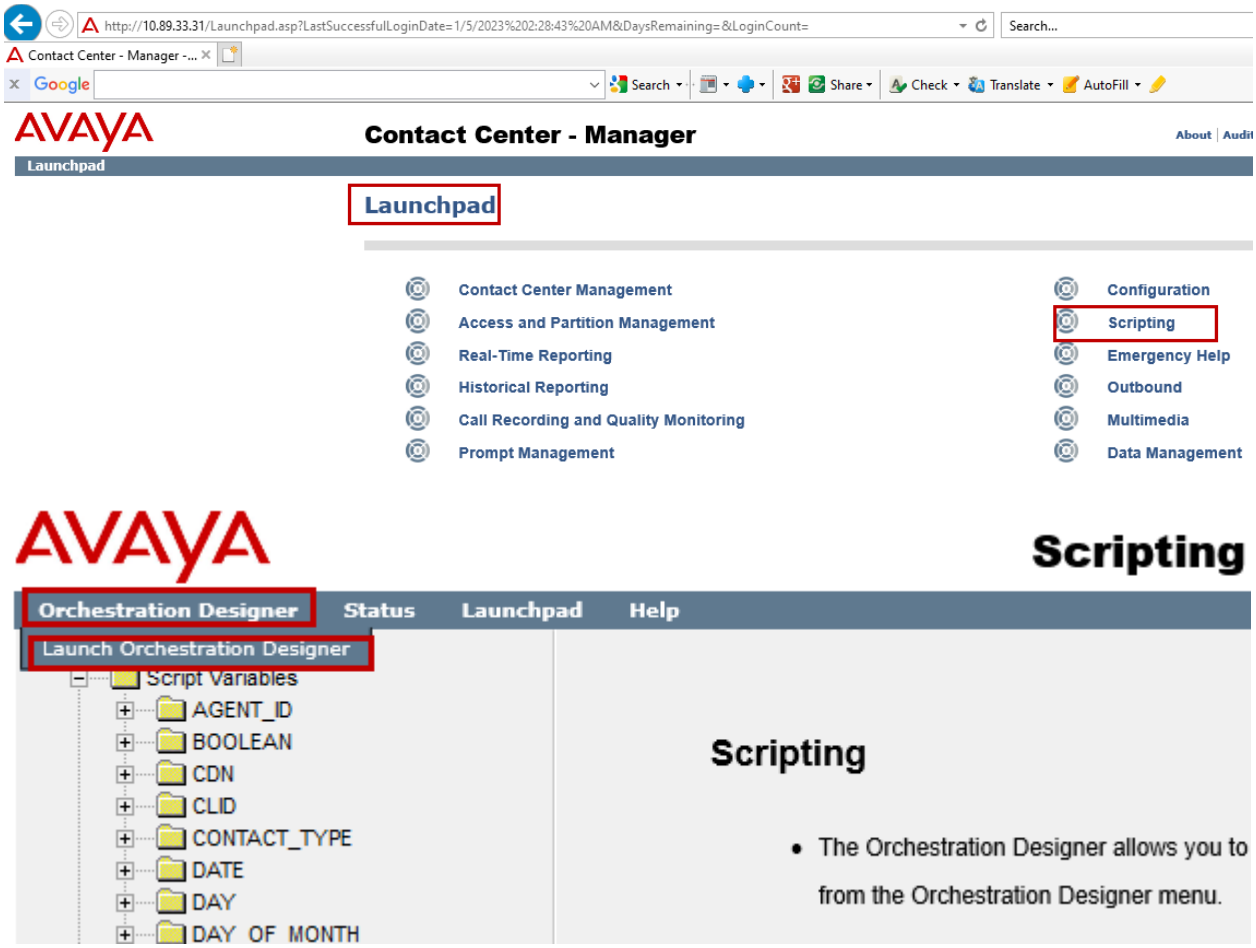



*Figure 162 Launch Orchestration Designer*

- In the Orchestration Designer, navigate to **Contact Center**
- Click **Connect to CCMA**
- *CCMA*: **10.89.33.31 (IP address of Avaya Aura CC)**
- *Port*:**80**
- *User ID*: **webadmin**
- *Password*: Type the appropriate **password**
- Click **OK**

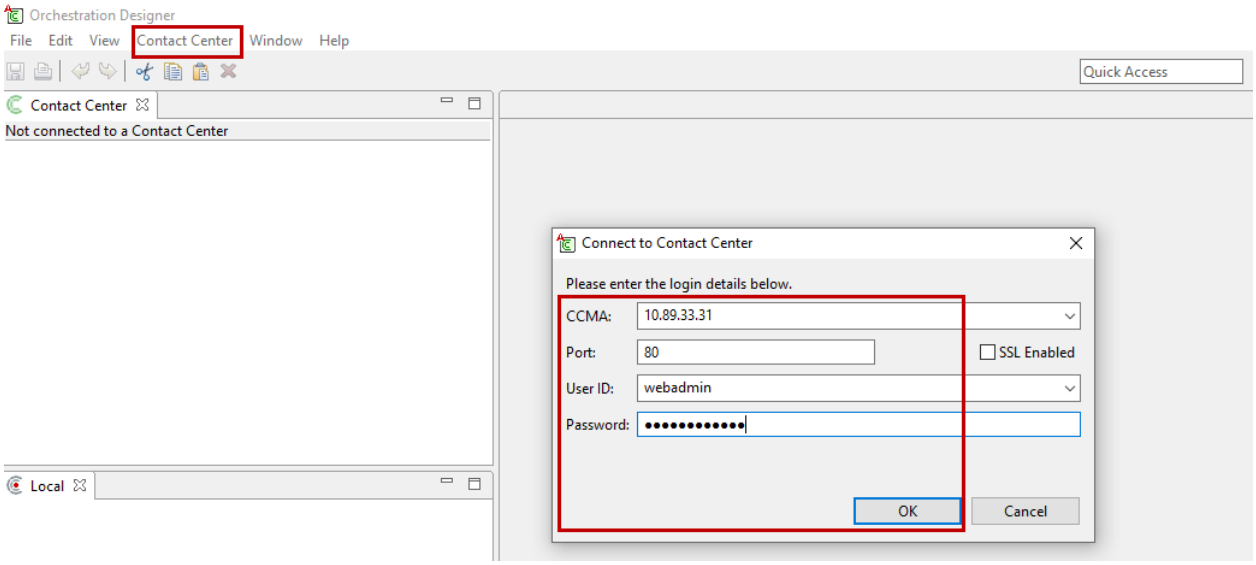



*Figure 163 Orchestration Designer Login*

- Navigate to **10.89.33.31 > AACC > Applications [Full Control] > Right Click > New > Application**

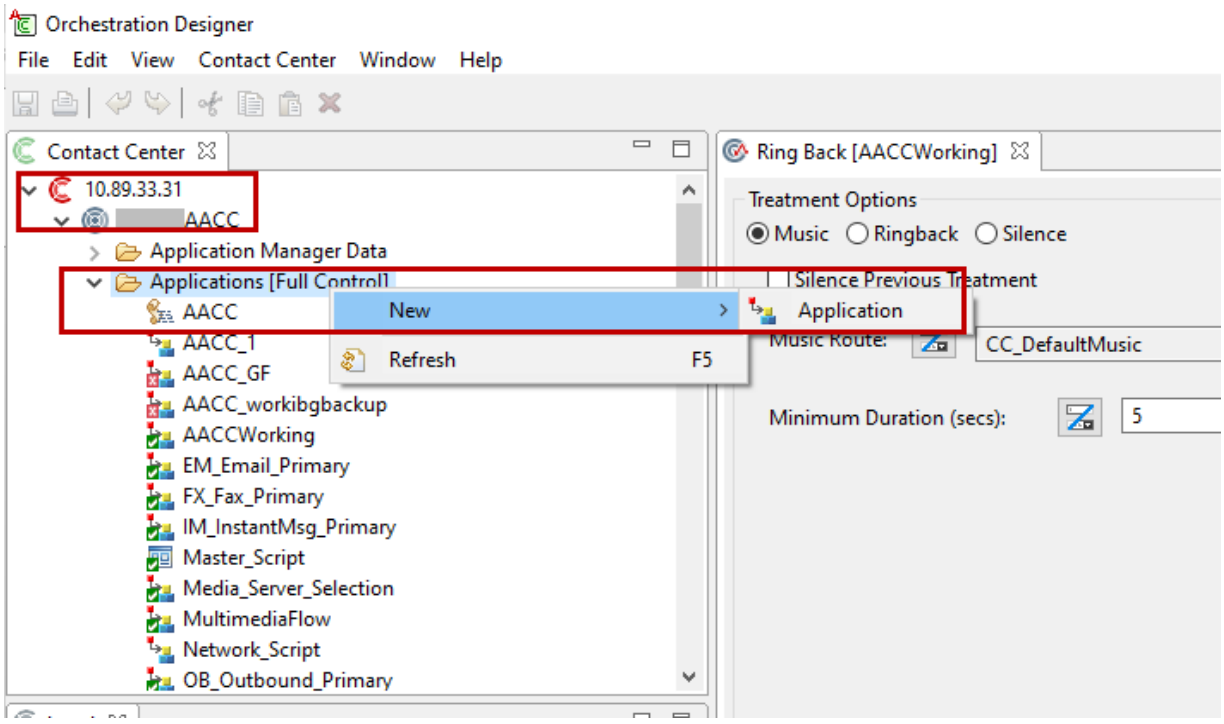



*Figure 164 Orchestration Designer – Create New Application*

In the **New Application** window,

- *Application Name*: **AACCWorking**
- *Application Type*: **Graphical Flow**
- *Application Template*: **New_Flow**
- Click **Finish**

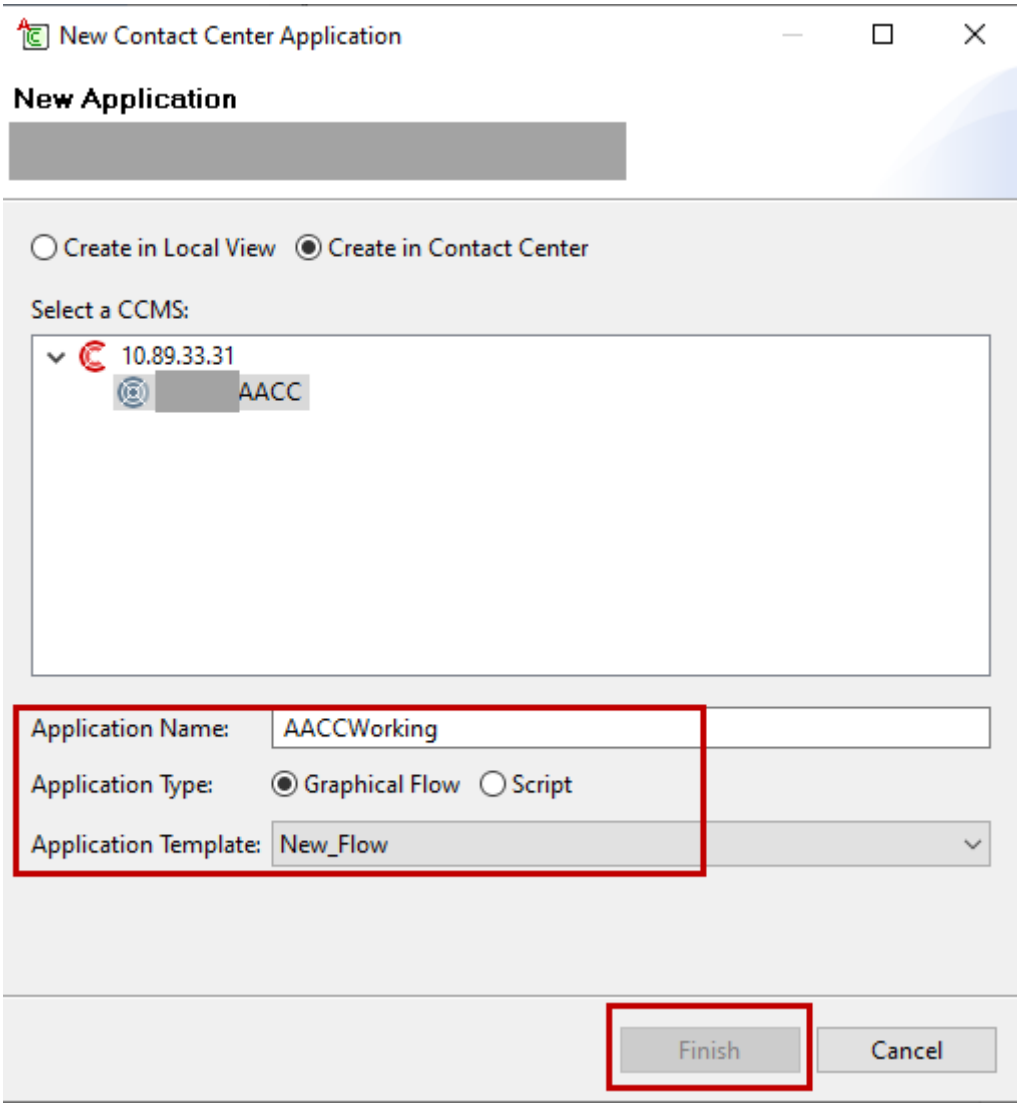



*Figure 165 Orchestration Designer – Create New Application Continuation*

The Graphical flow **AACCWorking** is designed to send the IVR digit entered by the PSTN user to be displayed in the Avaya Aura Agent Desktop

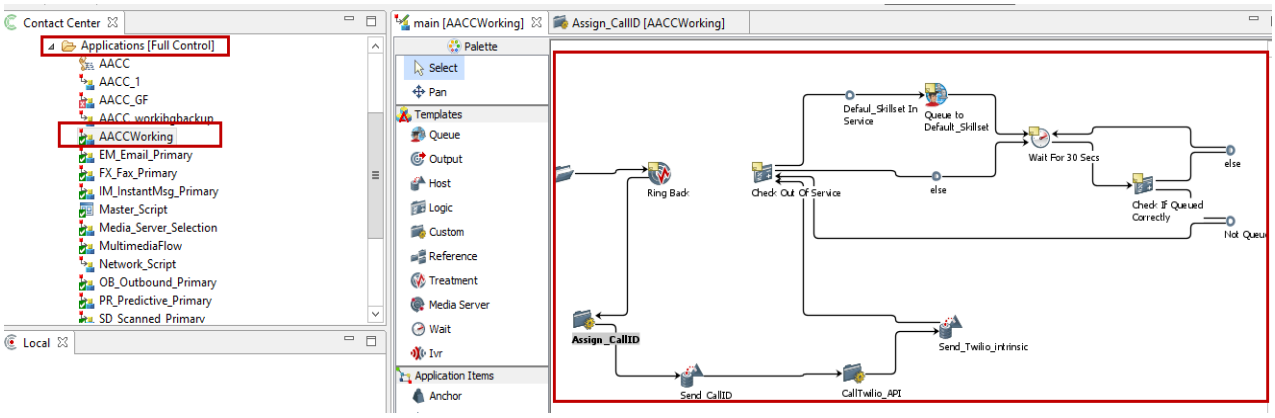



*Figure 166 Orchestration Designer – Graphical Flow*

**Detailed description of AACCWorking Graphical flow blocks**

**Ringback –** Plays music when the PSTN caller waits in the queue

- Double click on the **Ring Back** icon in the Graphical Flow (Figure 124)
- Under **Treatment Options**,
  - *Music*: **Enabled**
  - *Music Route*: Click **Browse** and choose **CC_DefaultMusic**
  - *Minimum Duration (secs):* **5**
- Click **Save** (Navigate to **File > Save**)

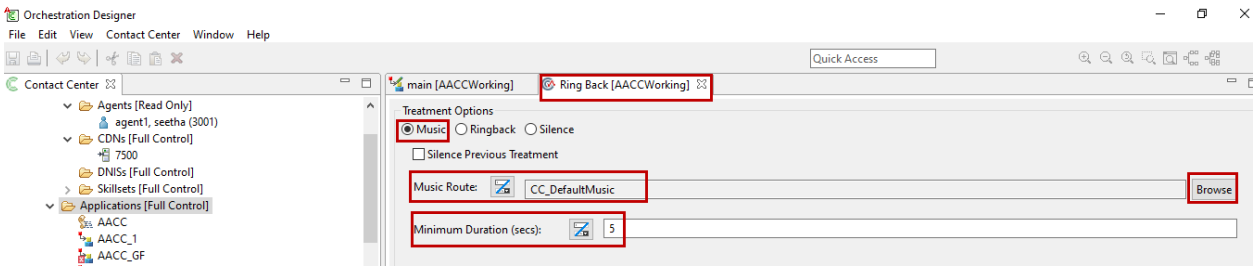



*Figure 167 Orchestration Designer – Graphical Flow Continuation*

**Assign_CallID –** Assigns the Twilio Call ID to a variable **mysearchURL_cv**. Refer Figure 130 and 131 to create a variable **mySearchURL_cv**

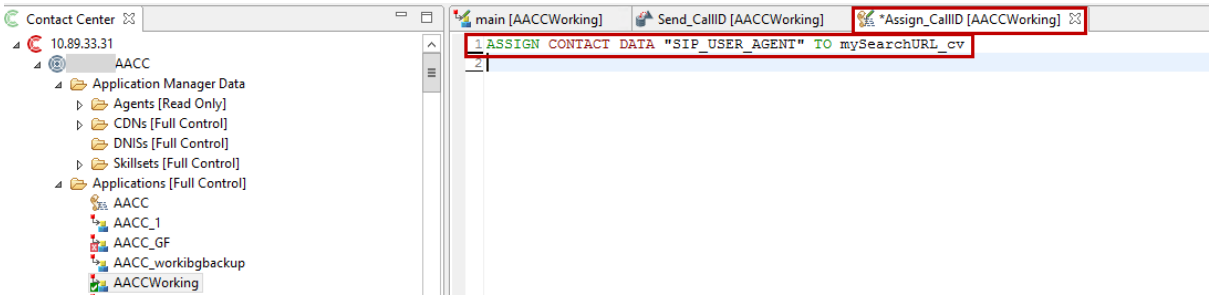



*Figure 168 Orchestration Designer – Graphical Flow Continuation*

**Send_CallID –** Sends the Twilio Call ID to the next block **CallTwilio_API**

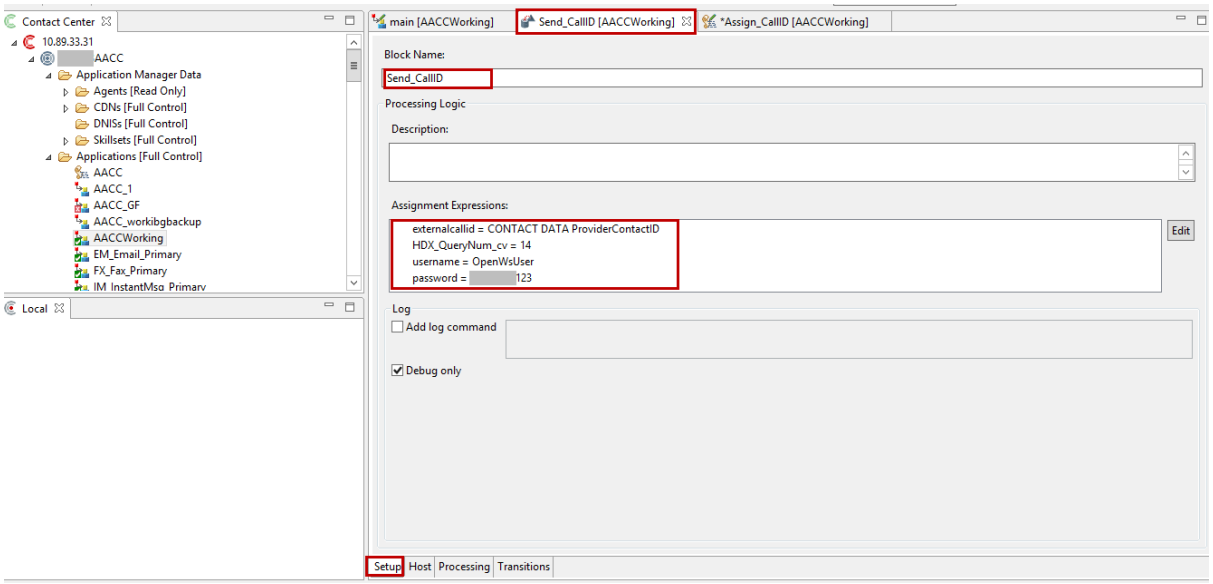



*Figure 169 Orchestration Designer – Graphical Flow Continuation*

Request Parameter **intrinsickey_cv** variable is assigned the value **Twilio_CallID**. Refer Figure 130 and 131 to create a variable **intrinsickey_cv**

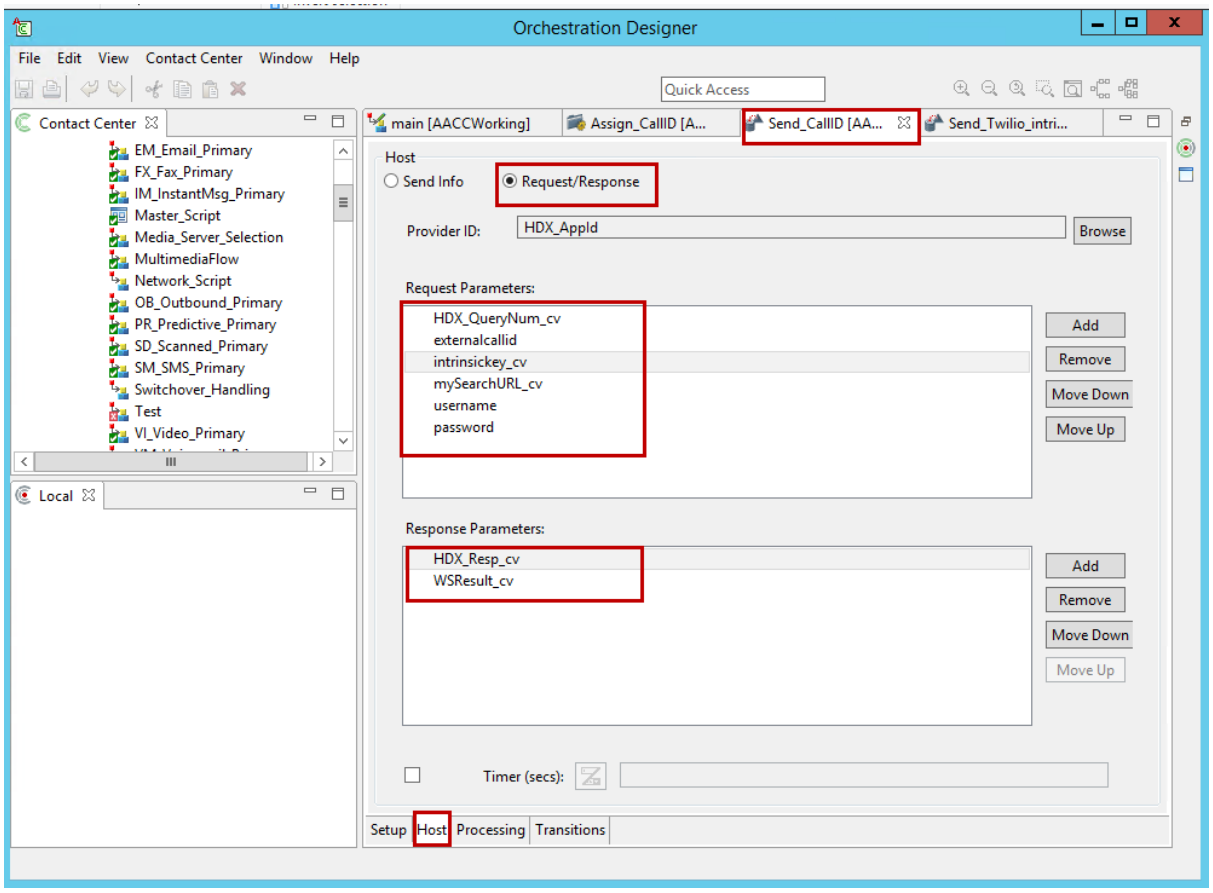



*Figure 170 Orchestration Designer – Graphical Flow Continuation*

**CallTwilio_API –** Query the Twilio API URL and sends the output to the next block

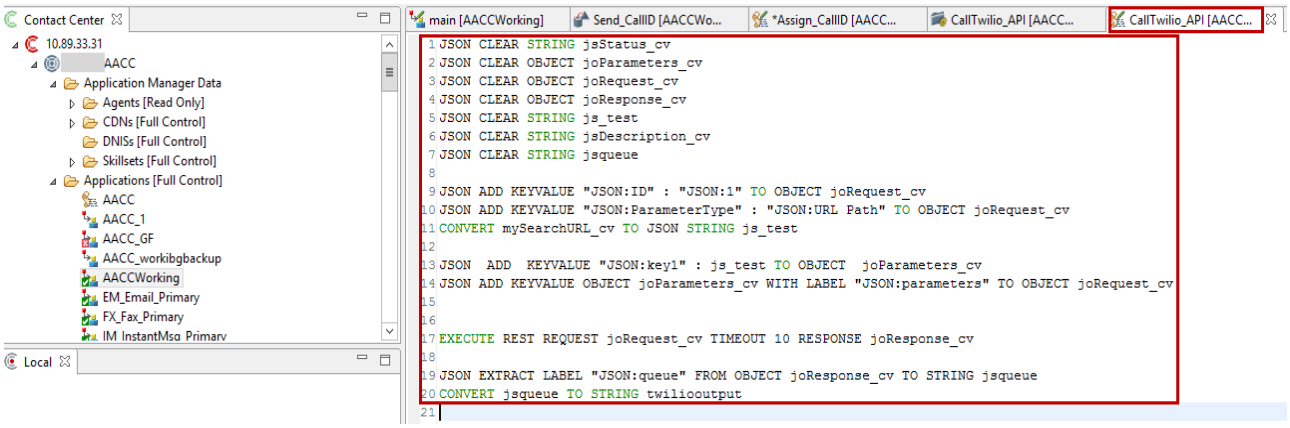



*Figure 171 Orchestration Designer – Graphical Flow Continuation*

**Send_Twilio_intrinsic –** Sends the response from Twilio API URL to Avaya Agent desktop

Navigate to **10.89.33.31 > Application Variables > STRING**

Create a global string variable named **intrinsicKey_gv** and assign a value IVR_Digit_entered. This is the string which is displayed in the Avaya agent desktop with the corresponding digit entered by the user.

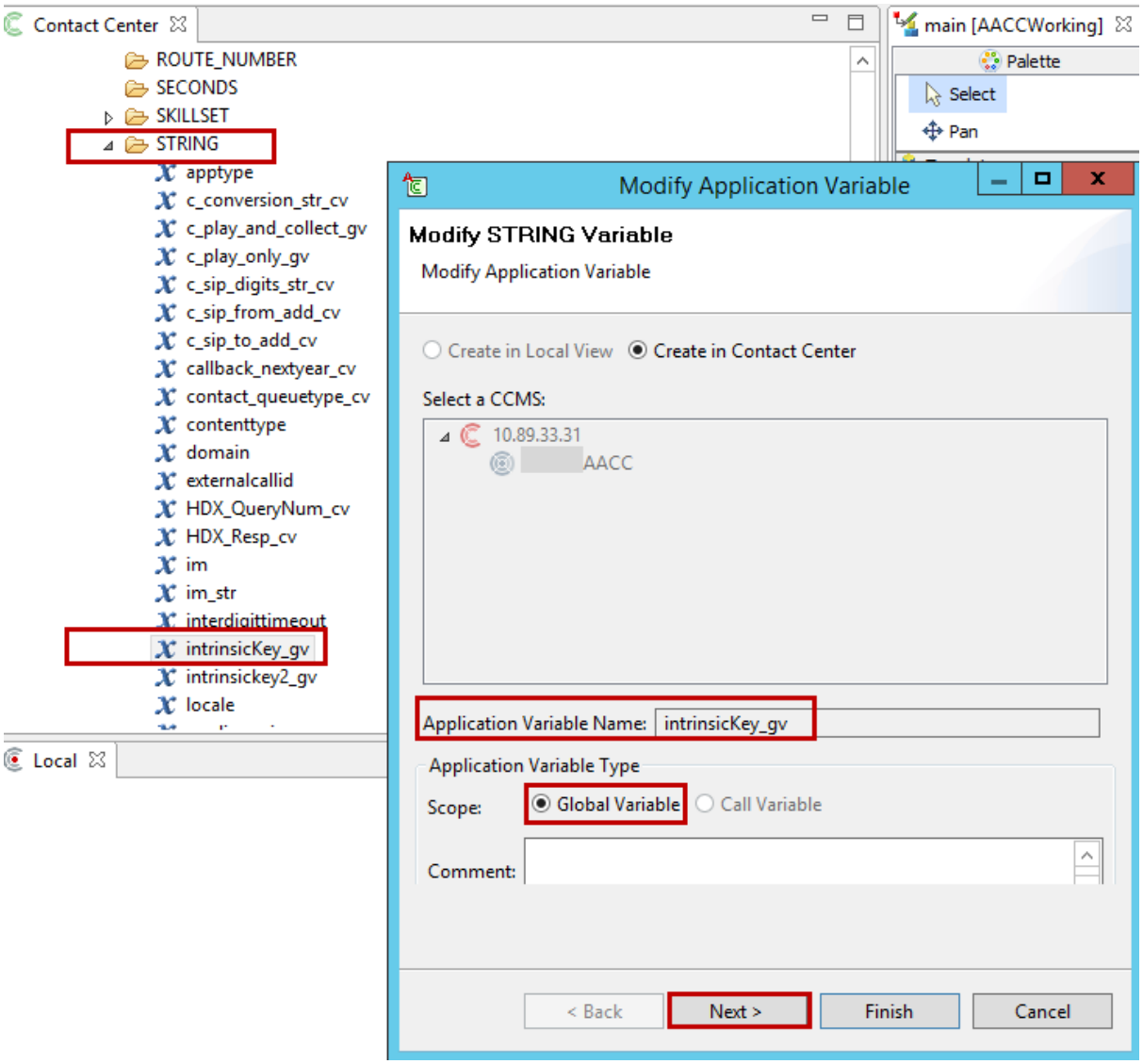



*Figure 172 Orchestration Designer – Graphical Flow Continuation*

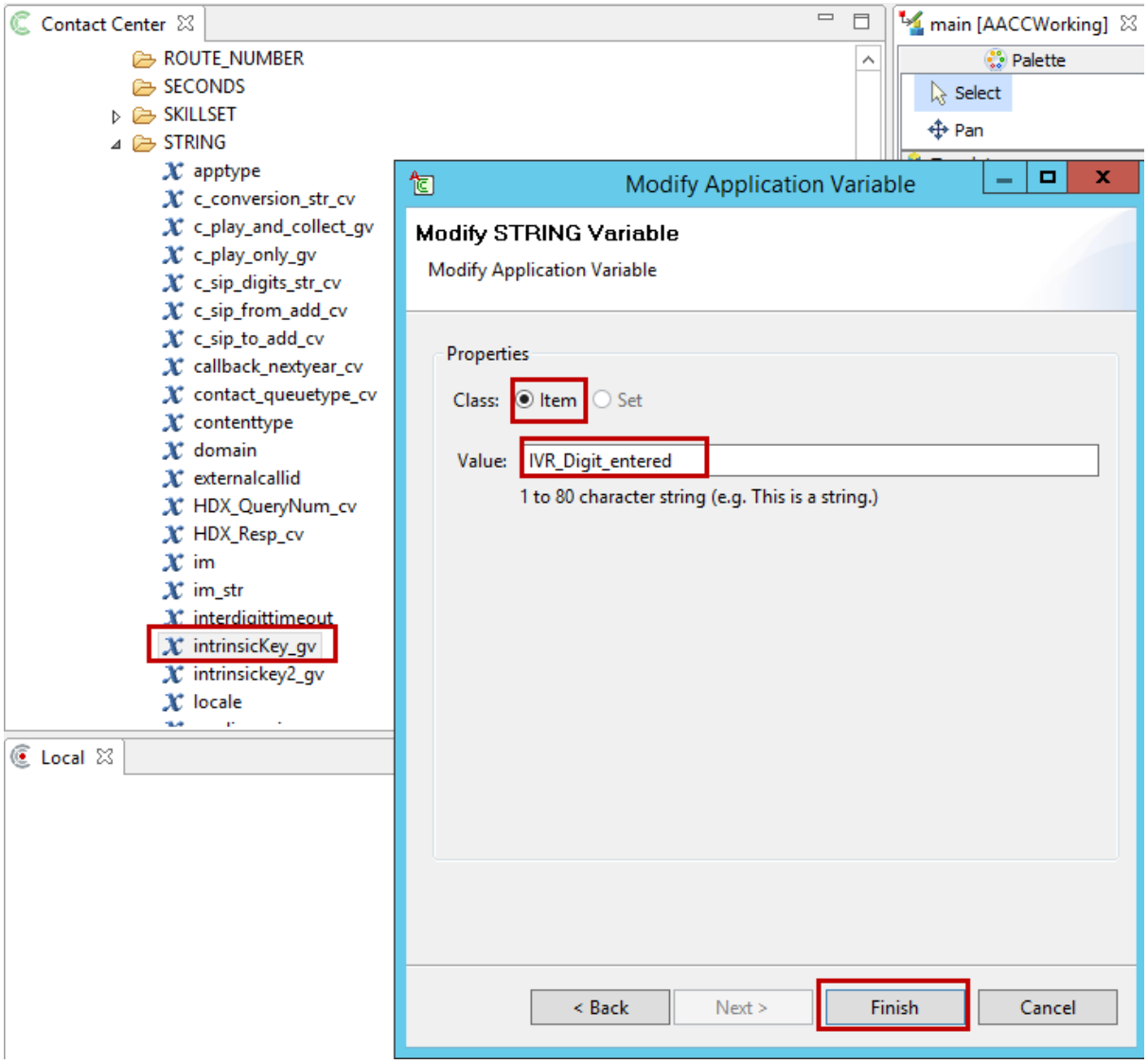



*Figure 173 Orchestration Designer – Graphical Flow Continuation*

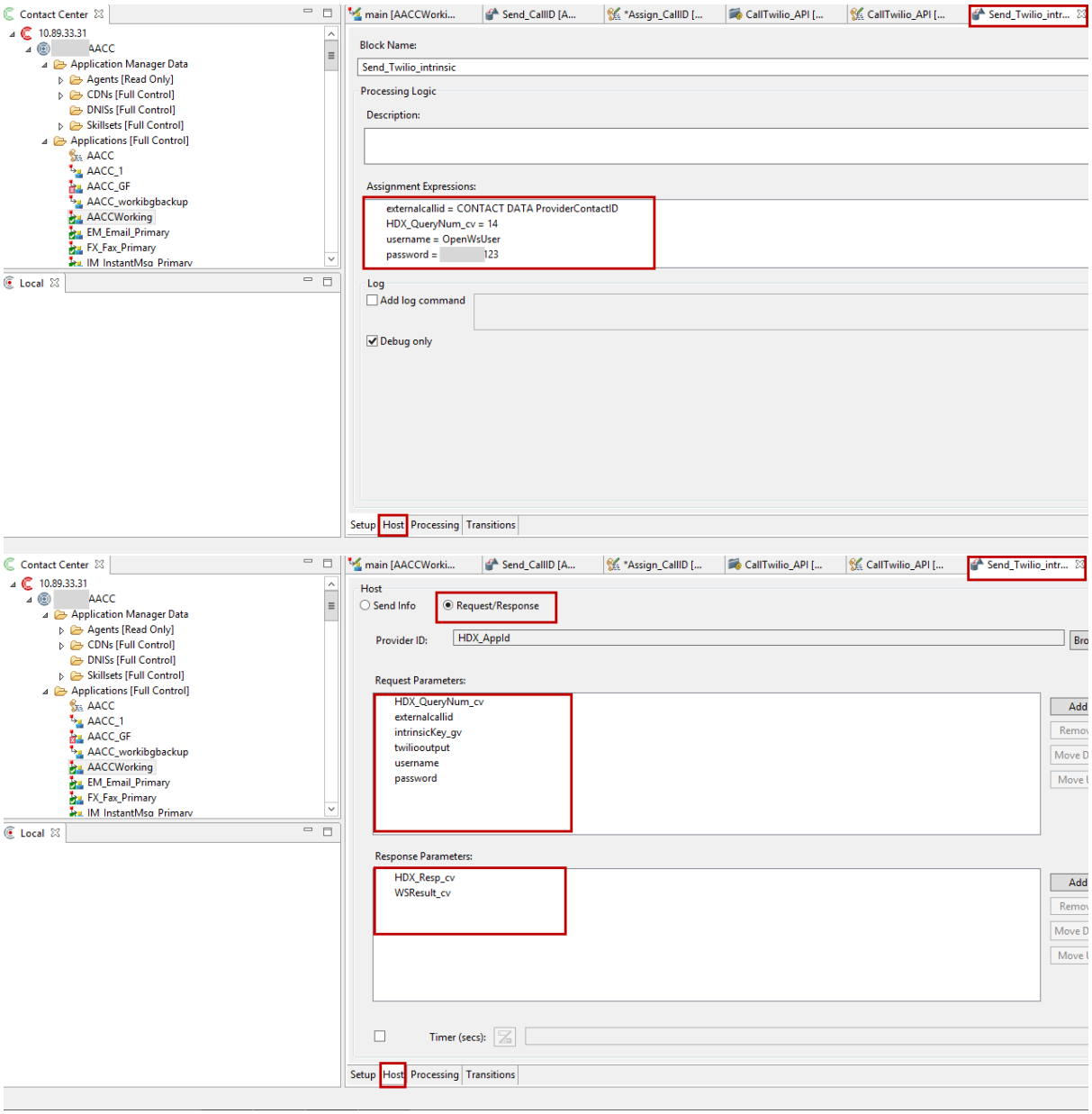



*Figure 174 Orchestration Designer – Graphical Flow Continuation*

- Right click on the application **AACCWorking**
- Click **Activate** to activate the application

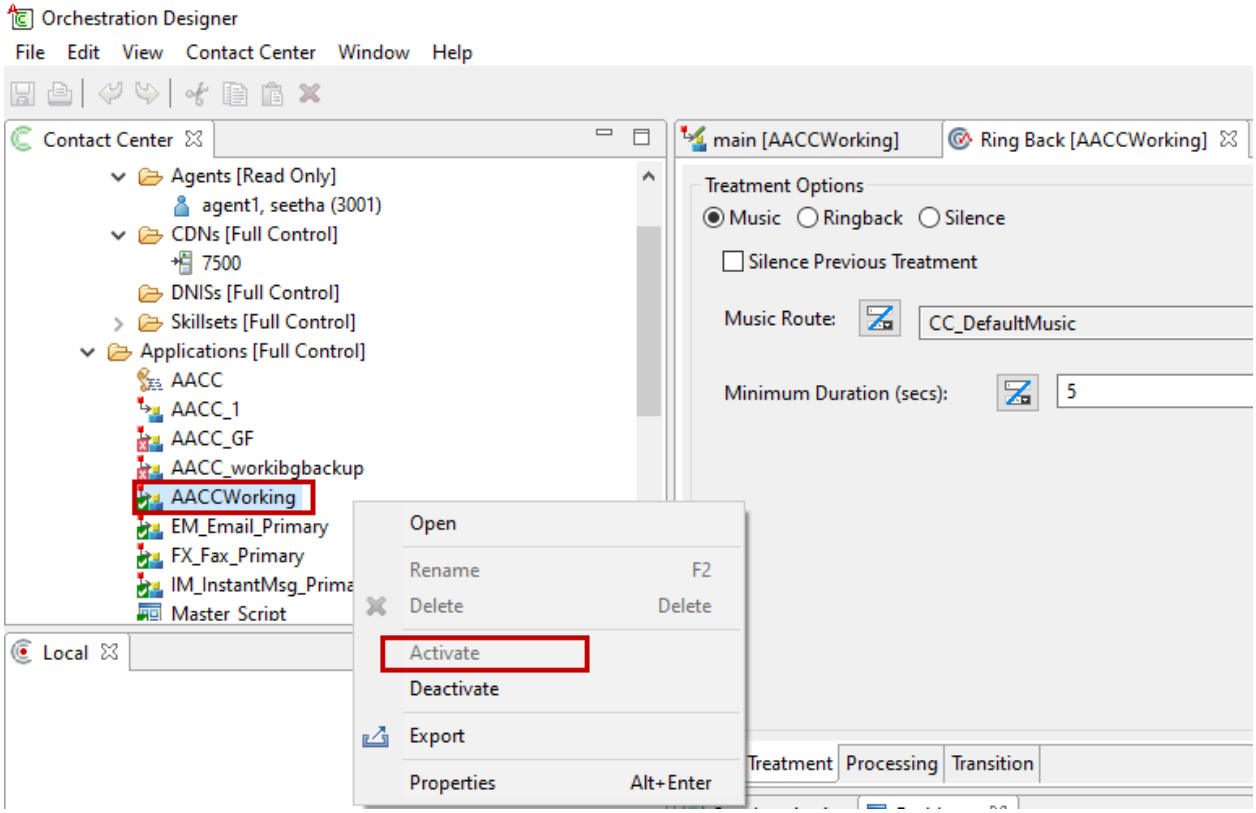



*Figure 175 Orchestration Designer – Activate Application*

- Navigate to **Application > Master_Script** to map the application **AACCWorking** with the CDN 7500
- Under **Configured Routes**,
    - Navigate to **Application Manager Data > CDNs > 7500**
    - Click **Add/Edit**
    - Select the application **AACCWorking** from **Application Chooser > Valid Applications**
    - Click **OK**

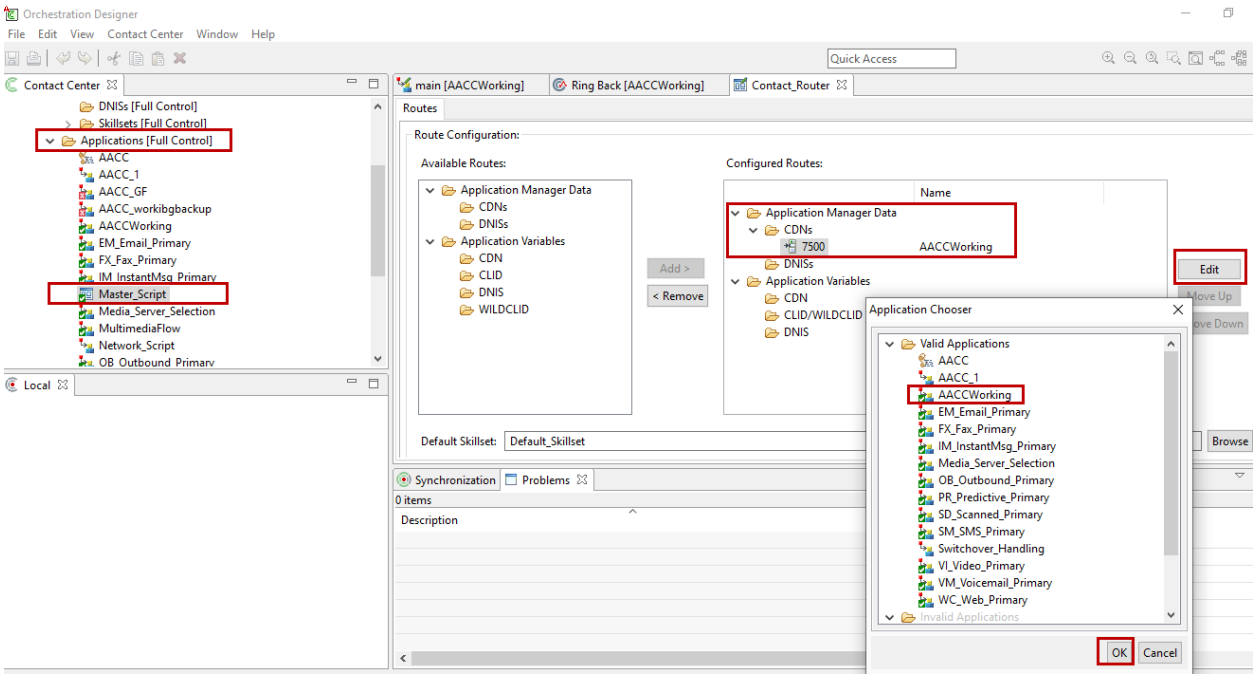



*Figure 176 Orchestration Designer – Master Script mapping with Application*

### 1.6.13 CCMM (Contact Center Multimedia) Administration

- Navigate to **Launch Pad > Multimedia** to configure the Avaya Aura Agent Desktop Settings


*Figure 177 CCMM*

- Click **Launch Multimedia Client**
- Click **Install prerequisite software** for the necessary pre-requisite software to be installed for CCMM Administration tool installation

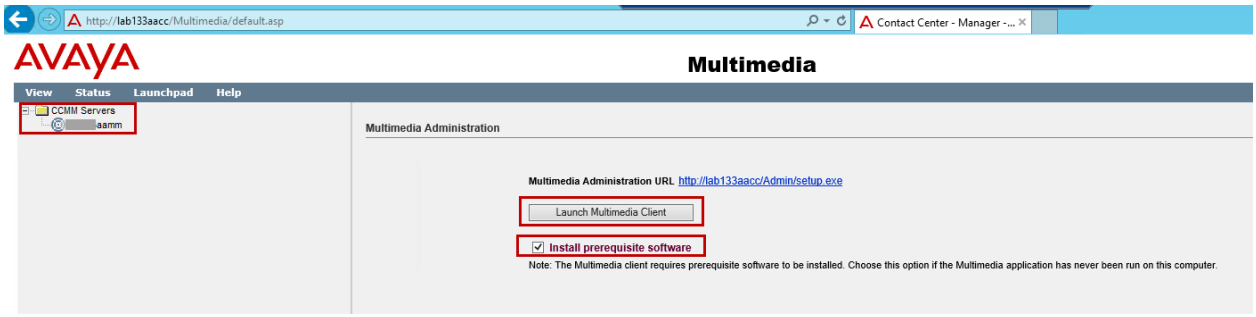



*Figure 178 CCMM Continuation*

- Navigate to **Agent Desktop Configuration > Basic Screenpops > General Settings**
- Below are the default settings under **General Settings**

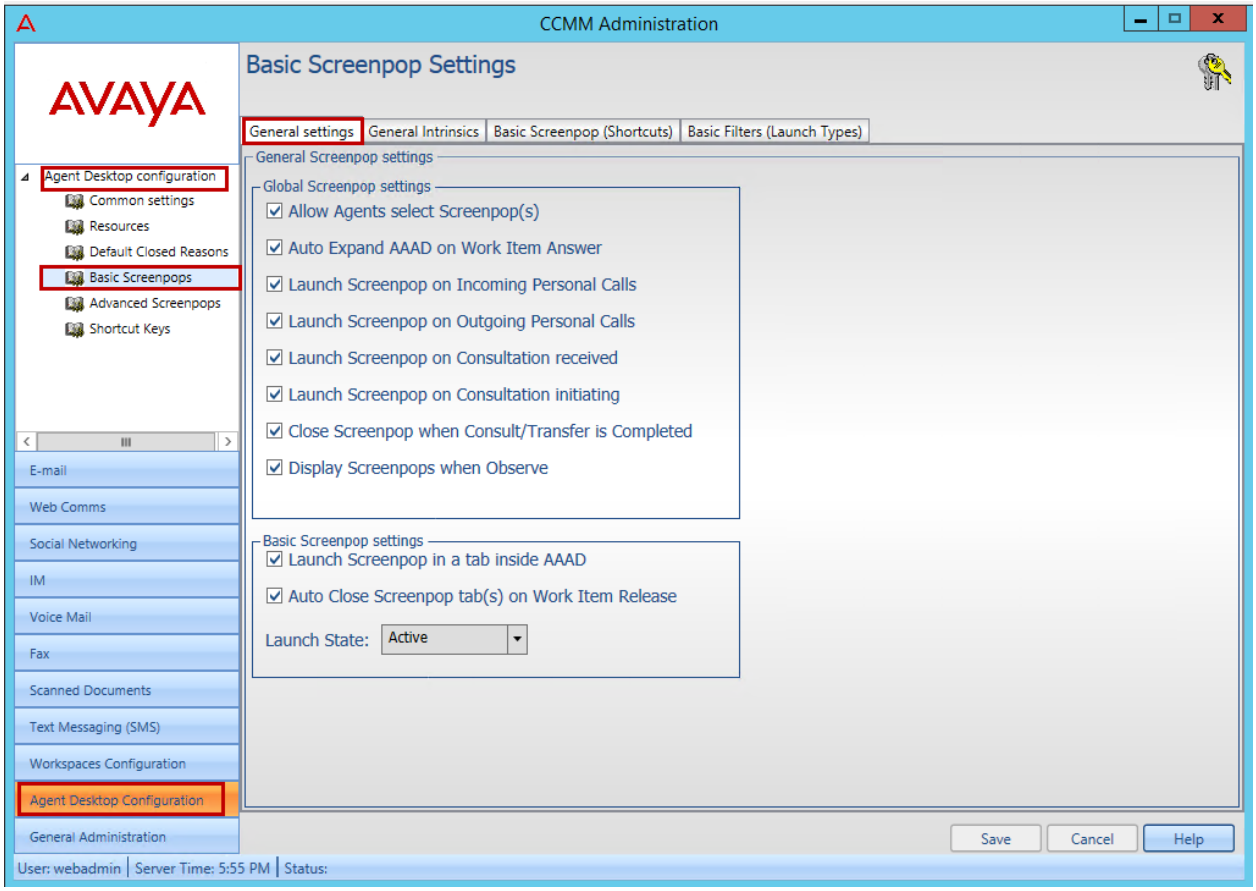



*Figure 179 CCMM Continuation*

- Navigate to **Agent Desktop Configuration > Basic Screenpops > General Intrinsics**
- Click **Add**
- *Name*: **IVR_Digit_entered**
- *Friendly Name*: **IVR_Digit_entered**
- *Display*: **Enabled**

Note:

- Avaya Aura Agent Desktop (AAAD) General Intrinsic settings maps the Screenpop Intrinsics **IVR_Digit_entered** with the IVR_Digit_entered intrinsic received from the Avaya Aura Orchestration Designer Application Script named **AACCWorking** and displays the digit entered by the PSTN user in the Avaya Aura Agent desktop

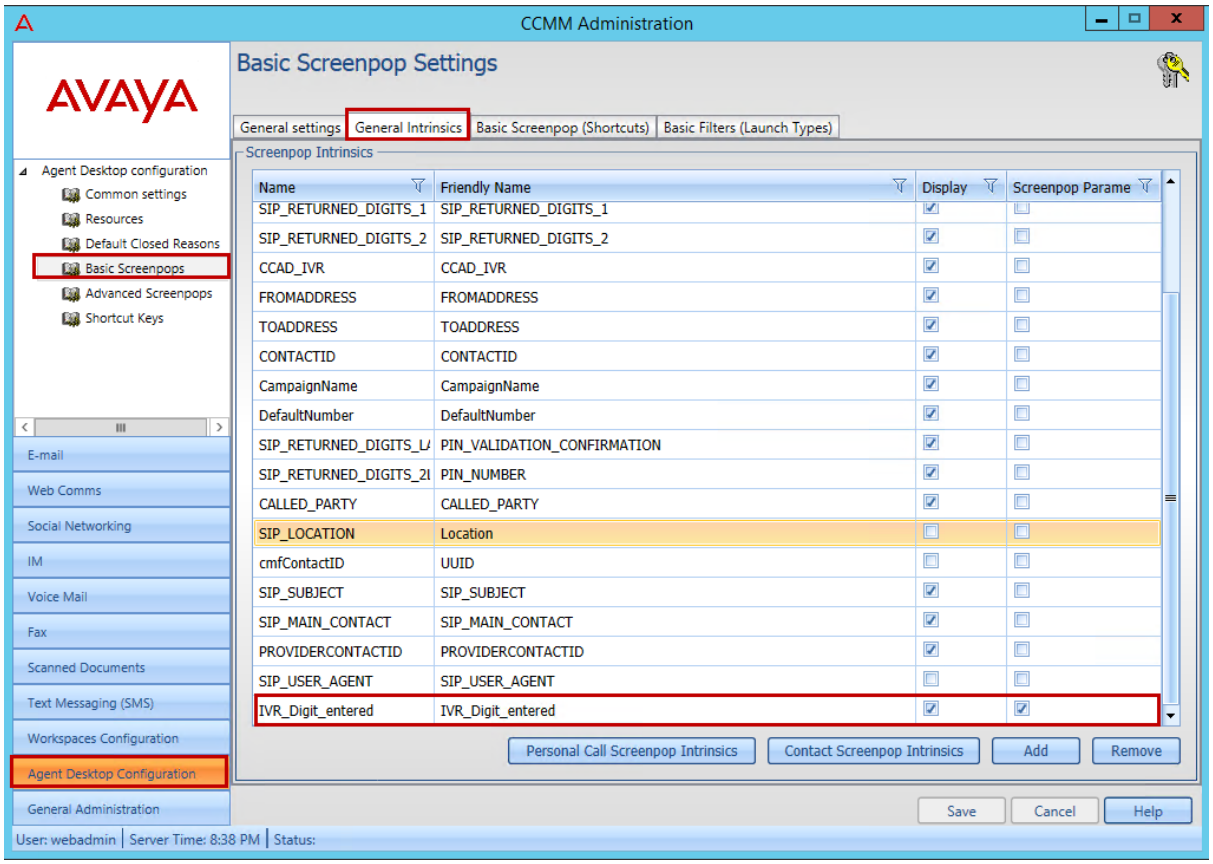



*Figure 180 CCMM Continuation*

## 1.6.14 Avaya Aura Agent Desktop

When the Avaya Aura Agent desktop client answers the call, it displays the IVR digit entered by the user as a SIP intrinsic named **IVR_Digit_entered** as shown below

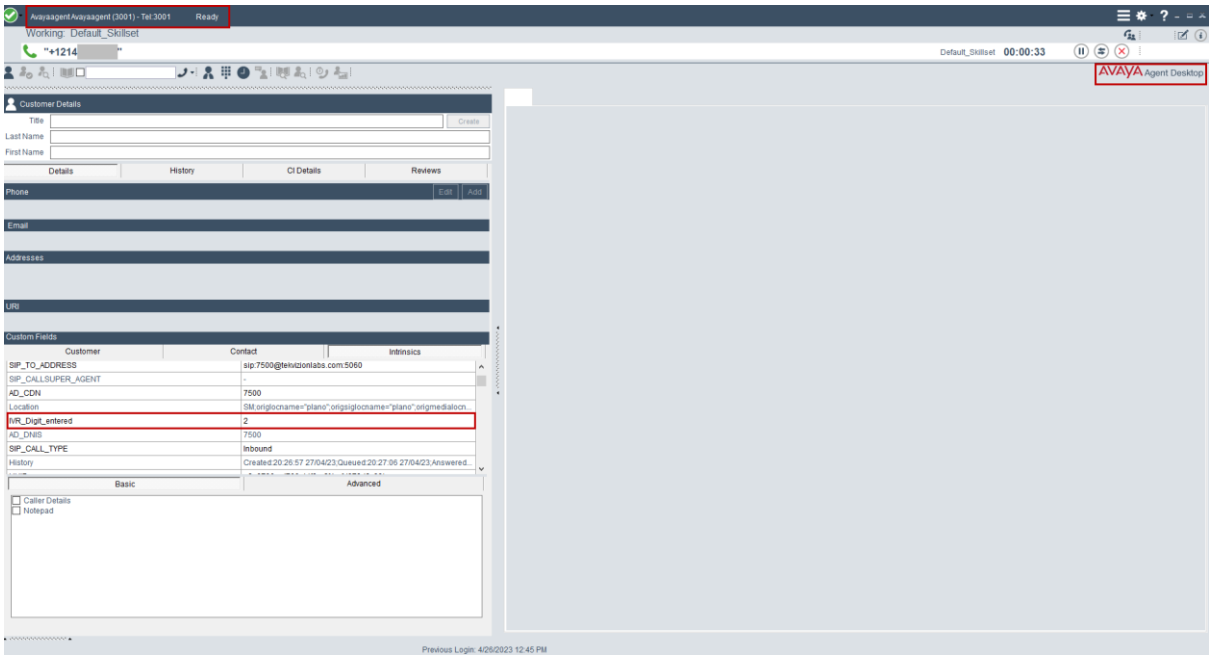



*Figure 181 Avaya Aura Agent Desktop*

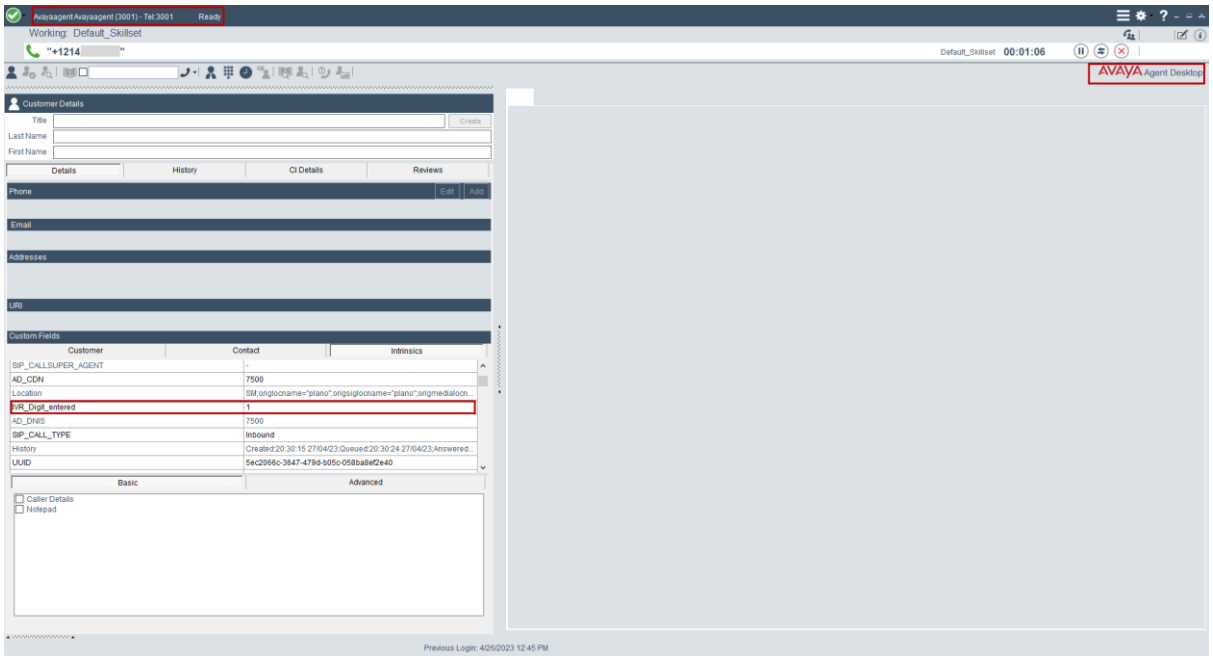



*Figure 182 Avaya Aura Agent Desktop Continuation*