LAB-2310

Your practical guide to troubleshooting for Webex Contact Center

Anuj Bhatia, D.K. Singh

Table of contents

1. Lab

1.1 Overview

1.1.1 Learning Objectives

This lab session is an intermediate-level session for engineers and administrators interested in the voice channel and its related troubleshooting techniques in a Webex Contact Center Solution. The session is divided into three exercises that cover following topics:

- **Configure and manage inbound call flows**: Participants will learn to set up inbound call flows and call distribution strategies.
- **Debug and troubleshoot common issues**: The lab will focus on using tools like flow debuggers to identify and fix configuration errors, whether in inbound flows or outbound dialing setups.
- **Handle call transfers and data preservation**: Participants will gain experience in ensuring a seamless call transfer process that maintains crucial data (flow variables) for agents.
- **Analyze agent performance**: The exercises will allow participants to use call reports to understand and analyze agent behavior, helping them identify the root cause of specific call events like rejections or "Re-Route on No Answer" (RONA) events.
- **Set up outdial capabilities**: The lab will cover the process of configuring a system to enable agents to make outbound calls.

Essentially, the lab is designed to give participants a holistic view of the call center solution, from initial setup and configuration to troubleshooting complex issues and analyzing performance related to the voice channel.

1.1.2 Disclaimer

Although the lab design and configuration examples could be used as a reference, for design related questions please contact your representative at Cisco, or a Cisco partner.

1.1.3 Lab Access

This lab leverages Webex Contact Center cloud environment.

1.2 Lab 1 - Configure and Debug Tenant Call Flows

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

`Control Hub URL`	https://admin.webex.com
`Username`	labuser**ID**@wx1.wbx.ai *(where **ID** is your assigned pod number; this ID will be provided by your proctor)*
`Password`	webexONE1!

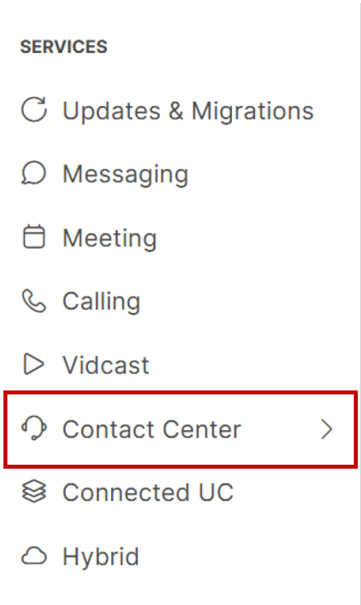
1.2.1 Objective

The objective of this self-paced lab is to provide participants with hands-on experience in configuring and debugging tenant call flows, as well as analyzing agent performance through call reports. This excercise has been divided into three sections

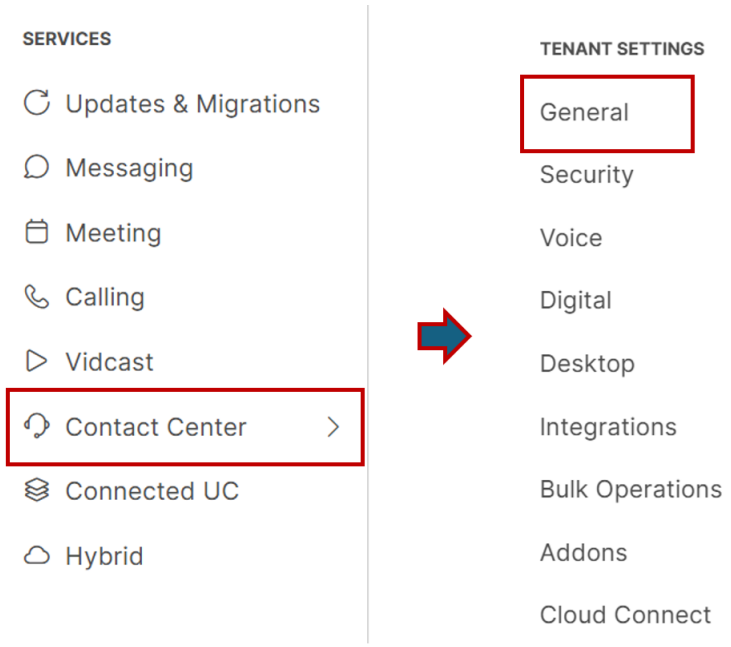
- **Section 1**: Discover Tenant Call Flow Configuration: This section aims to familiarize participants with the existing call flow configurations within a tenant environment.
- **Section 2**: Configure and Debug an Inbound Flow: In this part, participants will set up an inbound flow with a whisper announcement and a "Team" and "Longest Available" distribution strategy. They will use the flow debugger to identify and resolve common misconfigurations encountered during the setup process.
- **Section 3**: Analyze Agent Performance with Call Reports: After creating the inbound flow and presenting calls to a test agent, participants will intentionally trigger specific events. They will reject one call and allow a second to result in a Re-Route on No Answer (RONA) event. This will enable them to explore and analyze call reports to understand agent behavior and pinpoint the root cause of the rejected and RONA calls.

1.2.2 Section 1 : Discover Tenant Call Flow Configuration

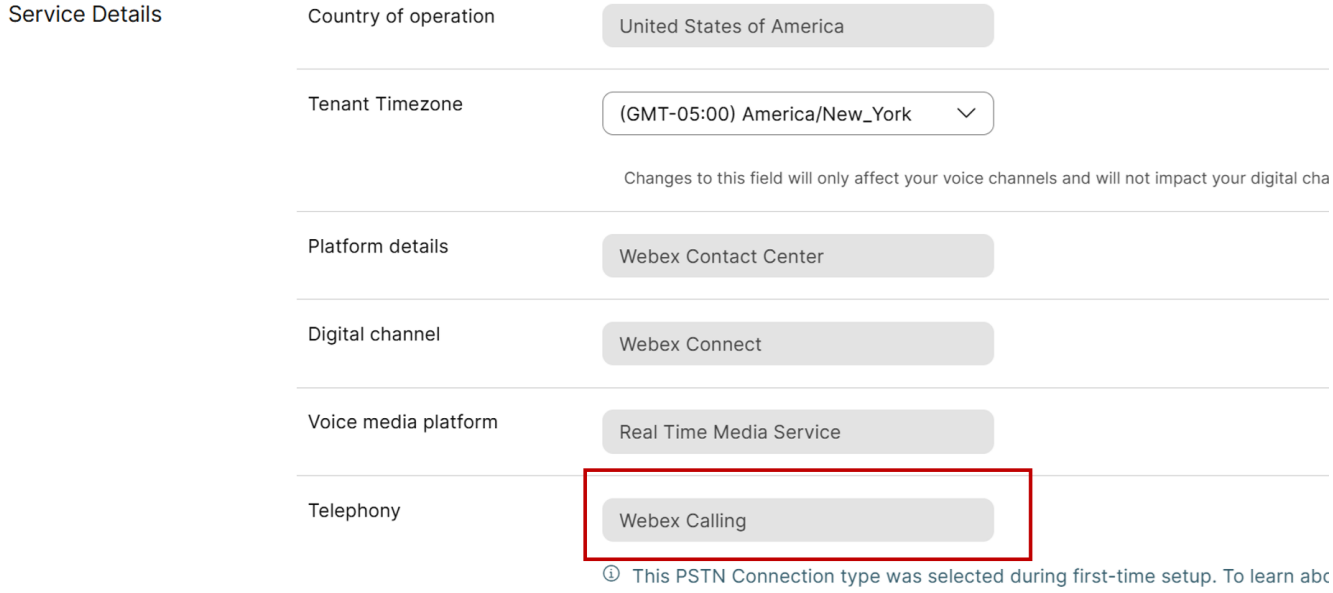
- Open your web browser and navigate to https://admin.webex.com
- Log in using the provided credentials.
- In Control Hub, locate Services and click on Contact Center.



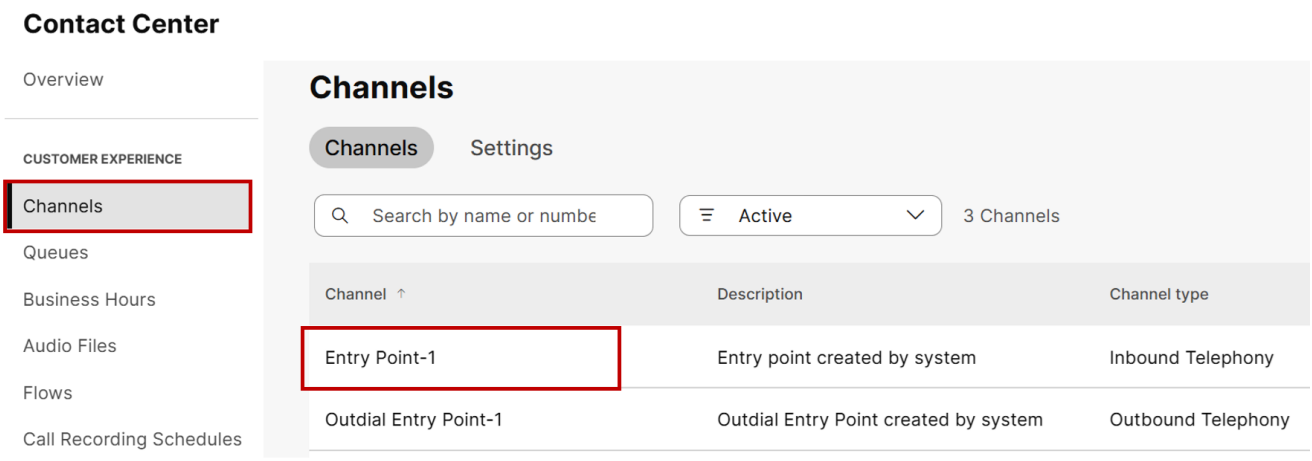
- In the Contact Center navigation pane on the left side, scroll down to the Tenant Settings section and click on General.



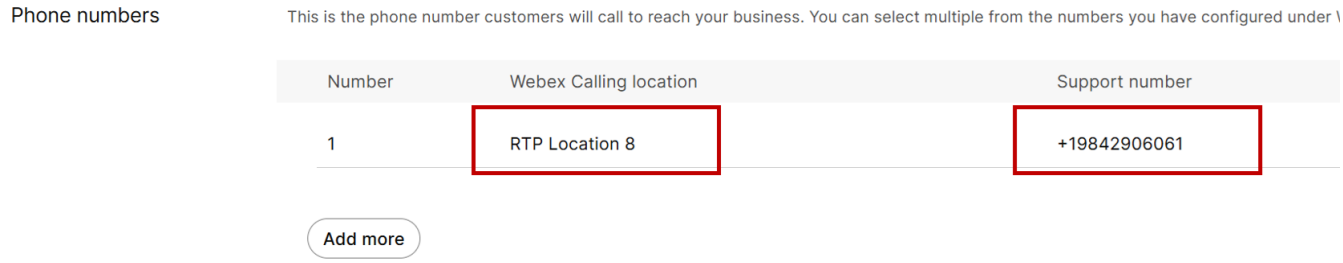
- Under the General settings, locate the Telephony type. Confirm that it states: **Webex Calling**.



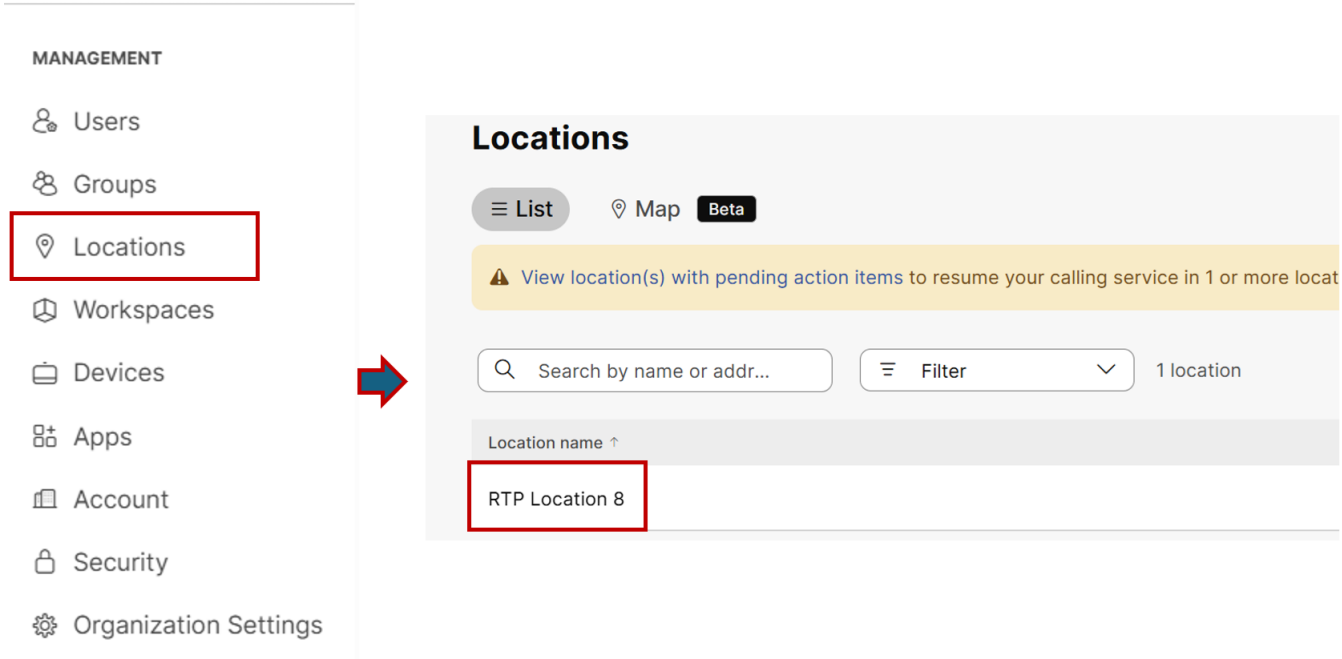
- In the Contact Center navigation pane, under the Customer Experience section, select Channels and search for the entry point named: **Entry Point-1**.



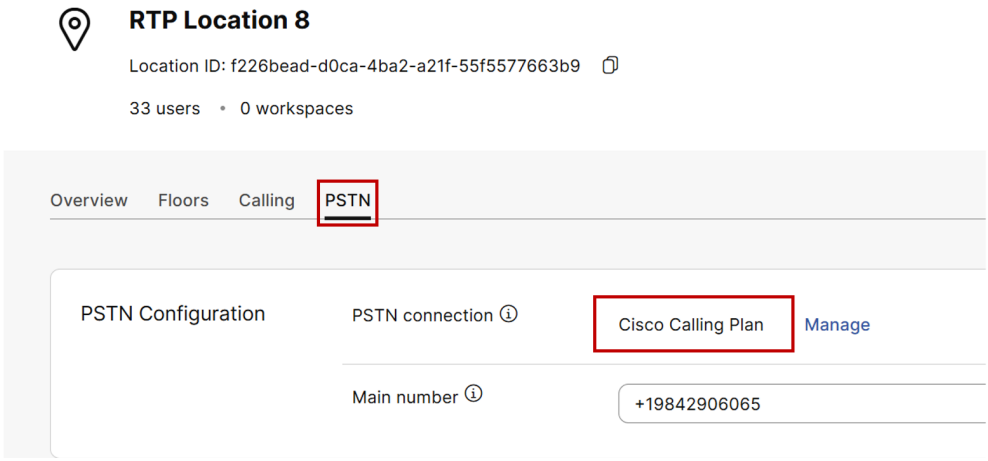
- Within the details for **Entry Point-1**, locate the Phone Number section and note the Webex Calling location and the Support Number.
- Webex Calling location: [**RTP Location 8**]
- Support Number: [+19842906065]



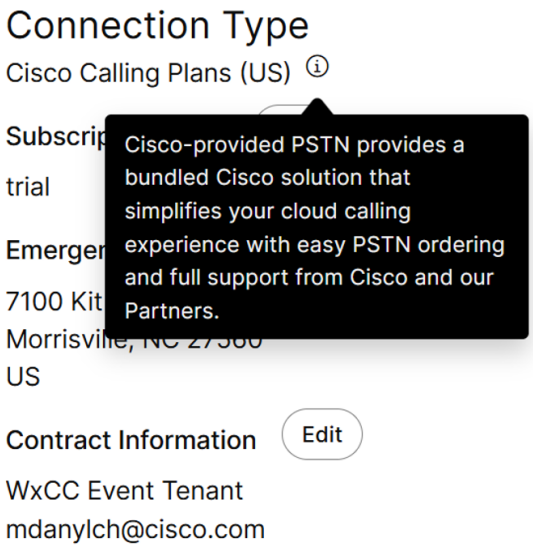
- Go back to the main menu of Control Hub.
- Under the Management section, select Locations.
- In the Locations search bar, enter the Webex Calling location name you discovered in the previous step.



- Select the discovered location.
- In the **PSTN tab**, under the **PSTN Configuration** section, the identified connection is **Cisco Calling Plan**.



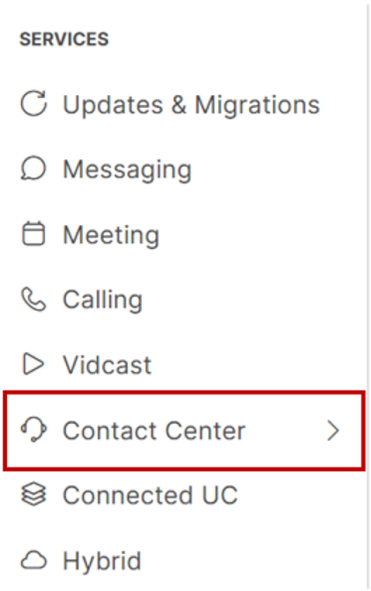
- Click on the **Manage** section and it provides all the details of the **connection type**



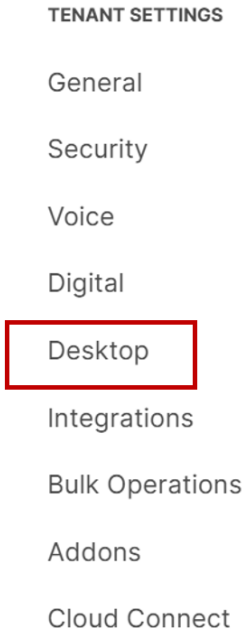
- Based on the observed calling connection (e.g., Cisco Calling Plan) and the information presented in the call flow slides, this configuration matches Flow Type 1.

1.2.3 Section 2 : Configure and Debug an Inbound Flow

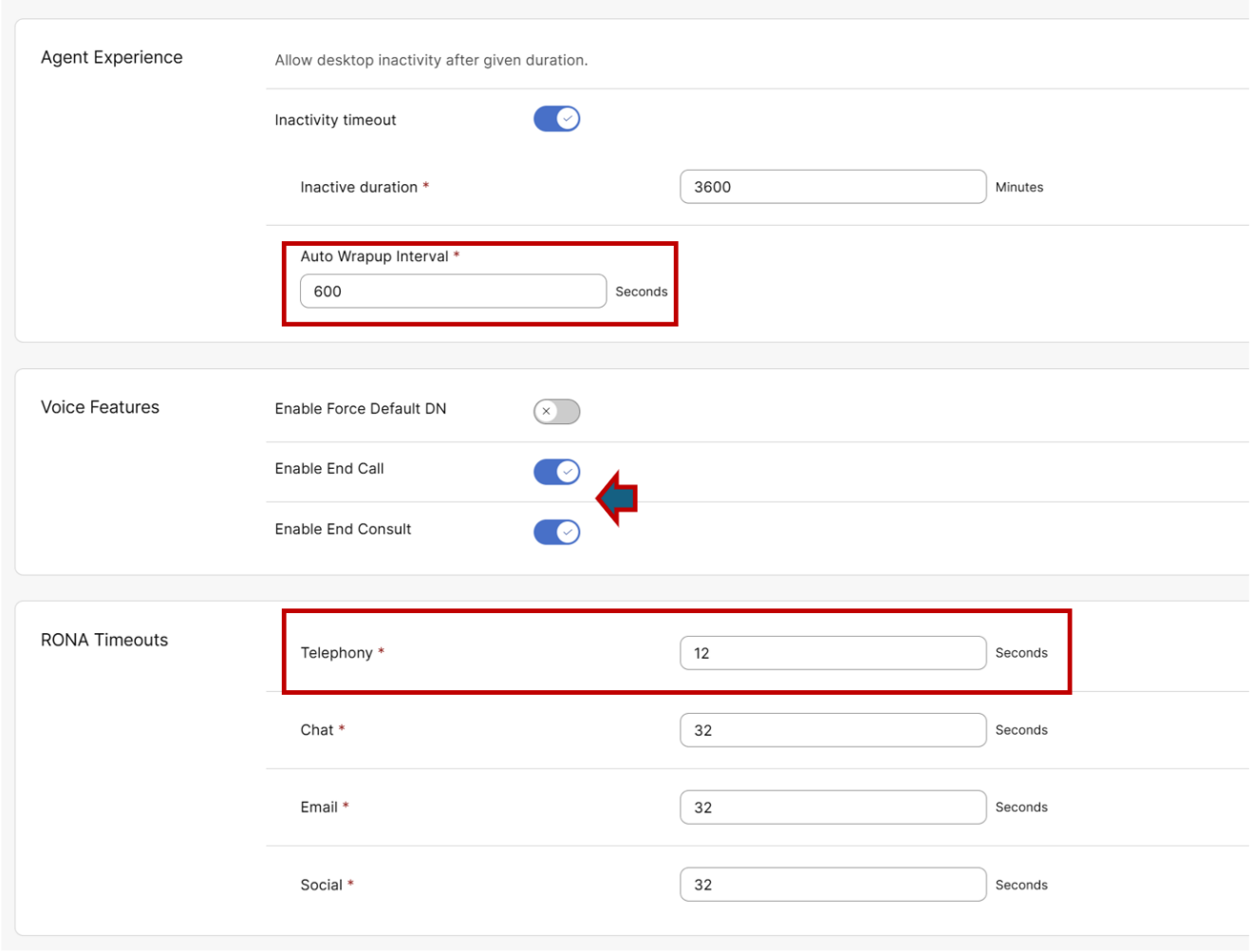
- The steps below covers the process for handling an incoming call, ensuring all necessary settings are in place for the agent to receive the call.
- First, to enable call handling options for the agent, navigate to "Contact Center" in the Control Hub.



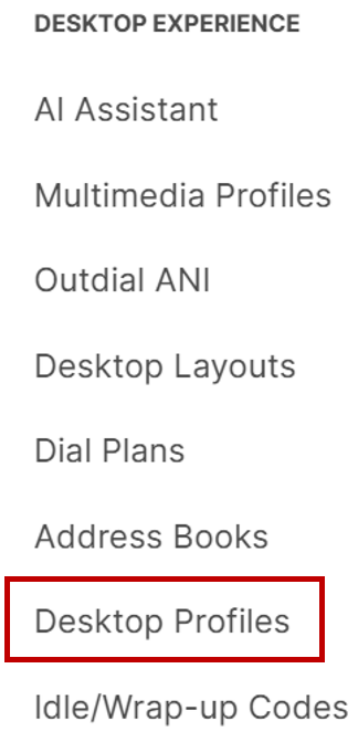
- Go to "Tenant Settings" and select "Desktop"



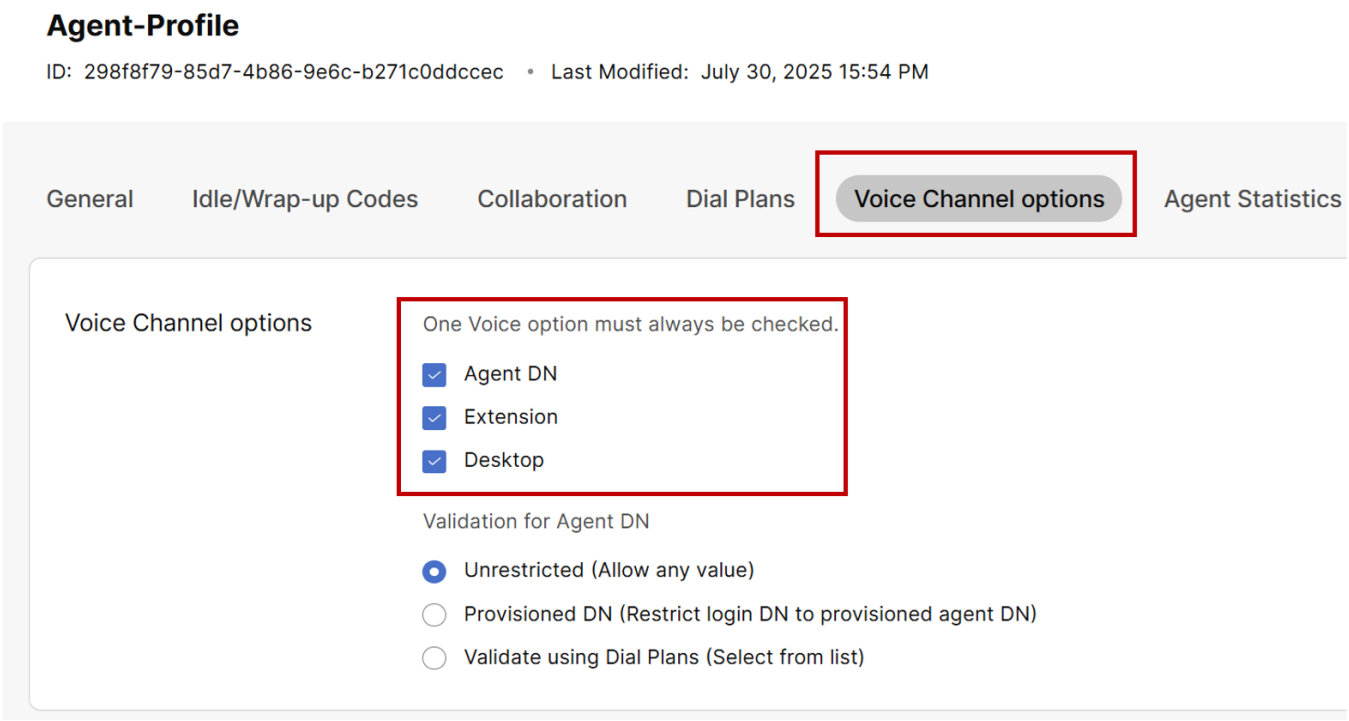
- Confirm that
- "**End Call**" and "**End Consult**"is enabled.
- "Auto Wrap-up interval" is set to **600 seconds**.
- "Telephony RONA timeout" is set to **12 seconds**.



• Navigate to "Desktop Experience" and select "Desktop Profiles."



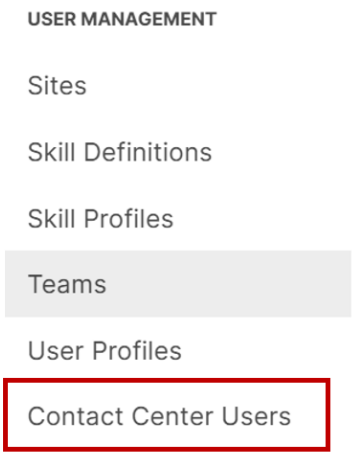
- Choose the **WebexOne_AgentProfile** and verify that the "Voice Channel" options are configured as shown in the following screenshot.



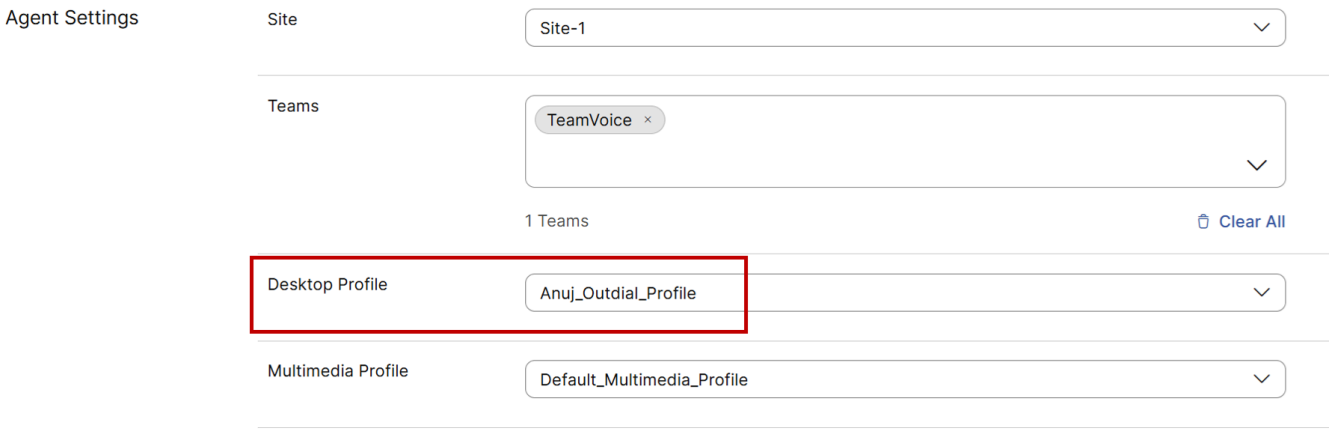
Note

Depending on the customer's business needs, the desired option can be selected. For example, if a customer wants all of their agents to use WebRTC and no other endpoints, only the 'Desktop' option can be checked. Agents will then only see the desktop option when logging into the desktop.

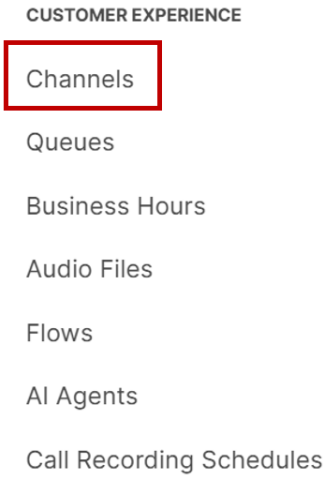
- To confirm that the agent is mapped to this profile, go to the "Contact Center users" section in "User management."



- Search for your agent **labuserID@wx1.wbx.ai** and, in their agent settings, ensure that the "desktop profile" is mapped to **WebexOne_AgentProfile**.

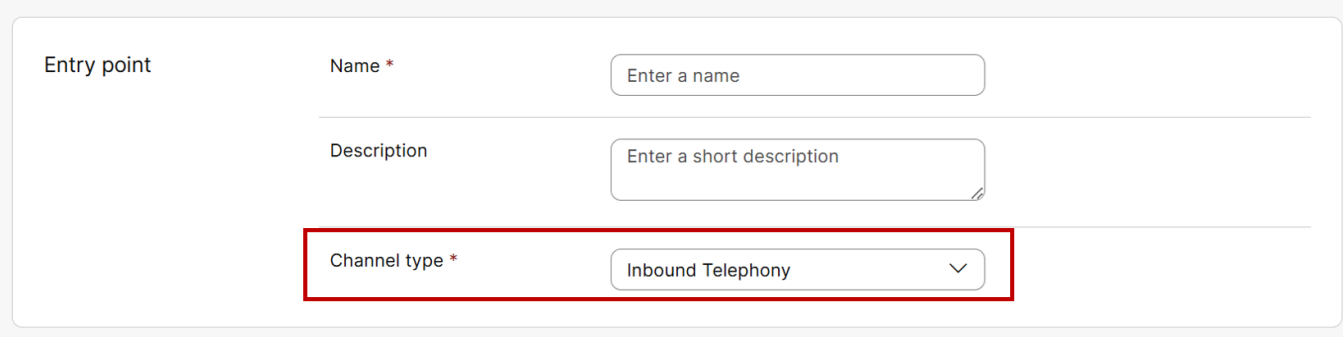


- Now, let's start building the entry point for the external call. To do this, navigate to "Customer Experience" and select "Channels."

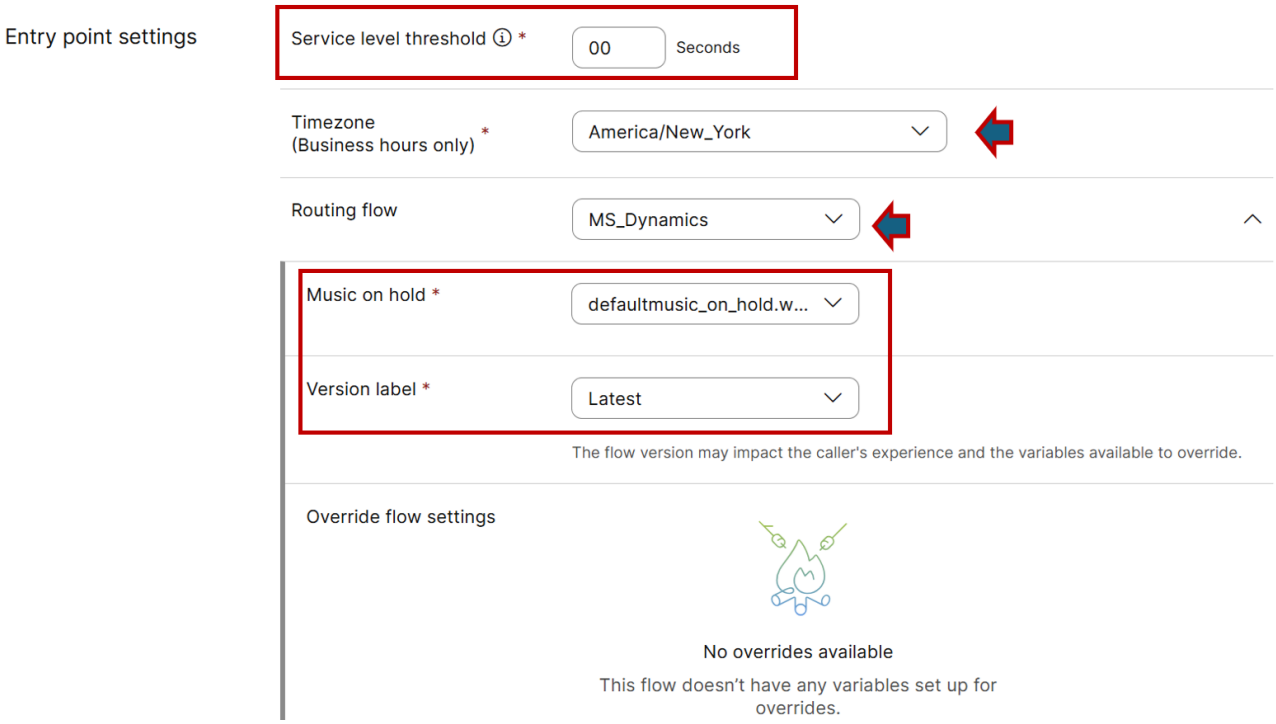


- Click "Create a channel."

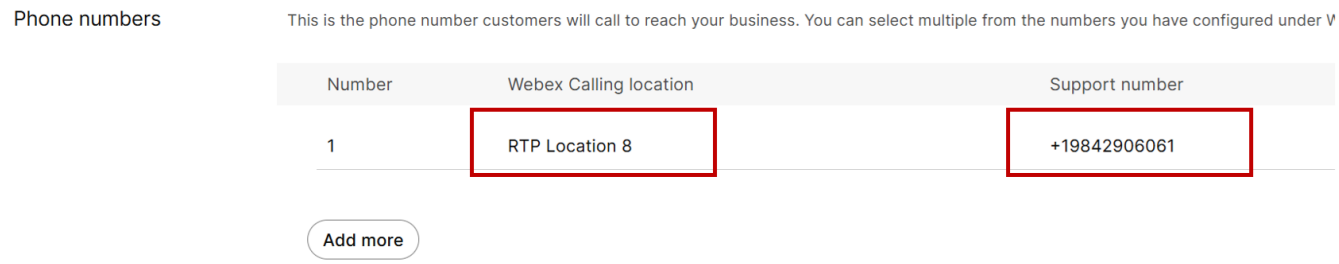
- In the "Channel Creation Wizard," provide the following details:
- Name: **Provide a descriptive name for your channel**
- Channel Type: **Inbound Telephony**



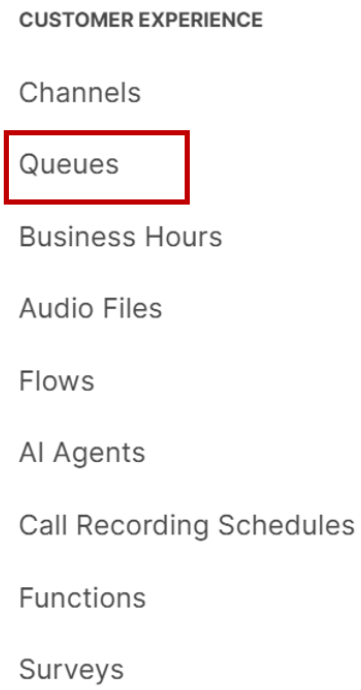
- Service Level Threshold: **30 seconds**
- Timezone: **America/New York**
- Routing flow: **Select Webexone_Flow_[num]** (Num is the assigned assignee's number)
- Music on hold: **defaultmusic_on_hold**
- Version label: **Latest**



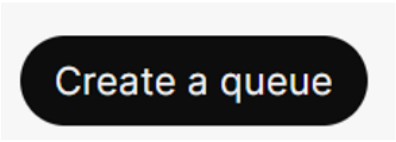
- In the "Phone numbers" section, select the following:
- Number Webex Calling location: **RTP Location 8**
- Support number: **Choose the available Number**
- PSTN Region: **Default**



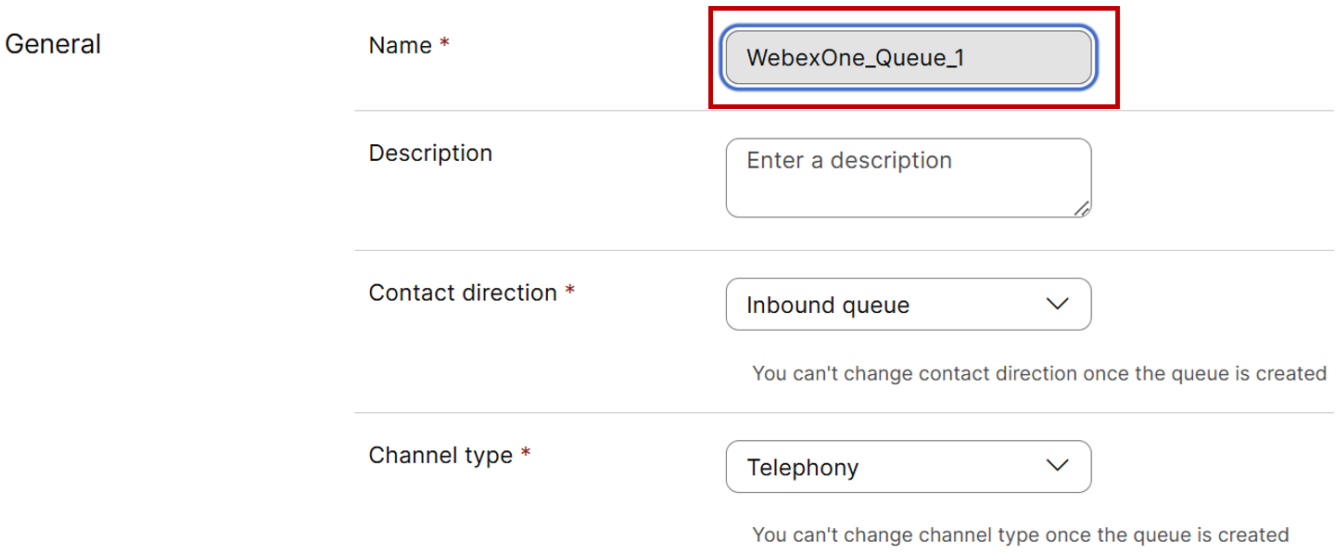
- After adding these settings, click "Create" to finalize the channel.
- The way an incoming call is presented to an agent in agent-based routing is via the entry point to the routing flow and the flow mapped to the queue which has the team the agent belongs to.
- With this logic, let's check the queue and make the necessary configuration.
- In the Contact Center navigation pane, under Customer Experience, select Queues.



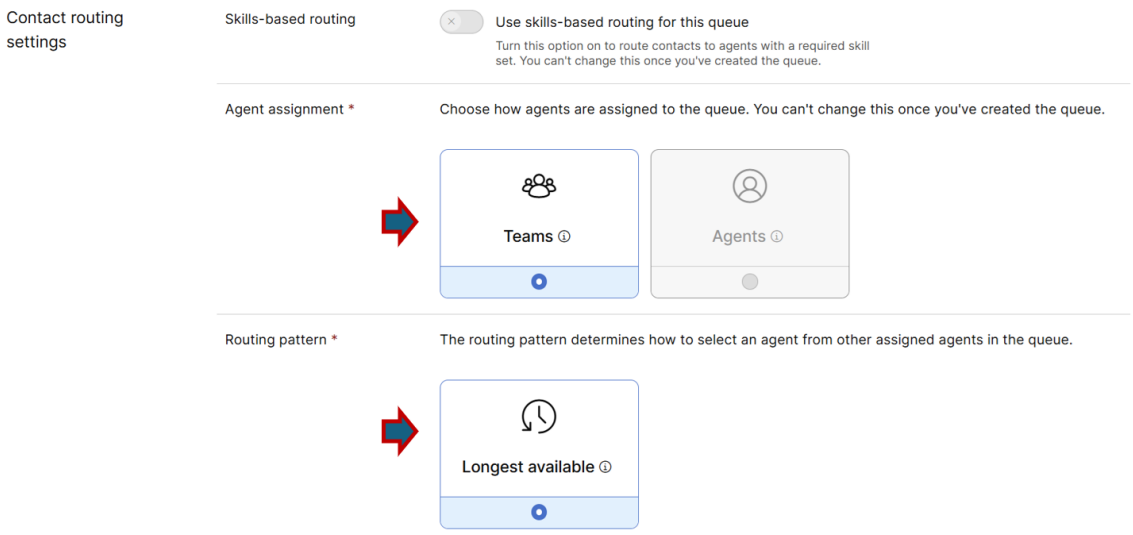
- Click the **Create a Queue** button



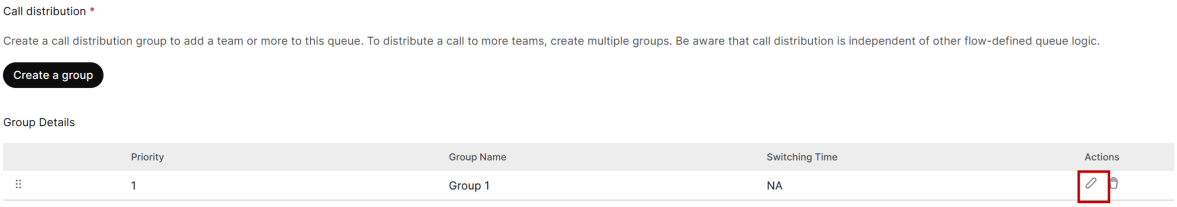
- Configure the new queue by providing the name **WebexOne_Queue_[num]** and ensuring these settings match:
- Contact Direction : Inbound Queue
- Channel Type : Telephony

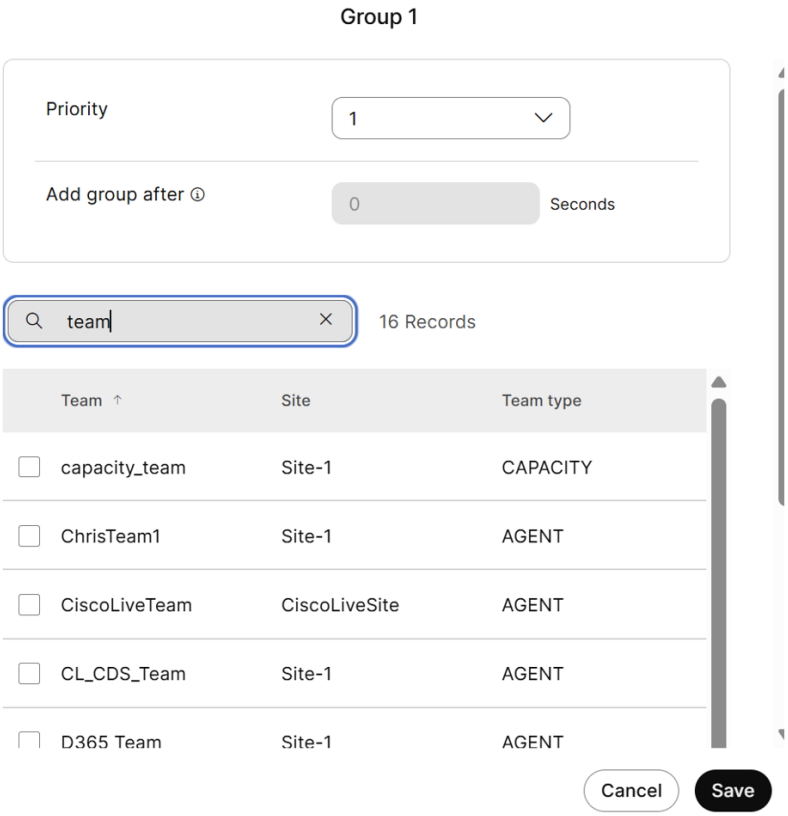


- In Contact Routing Settings, confirm these settings:
- Agent Assignment: Teams
- Routing Pattern: Longest available

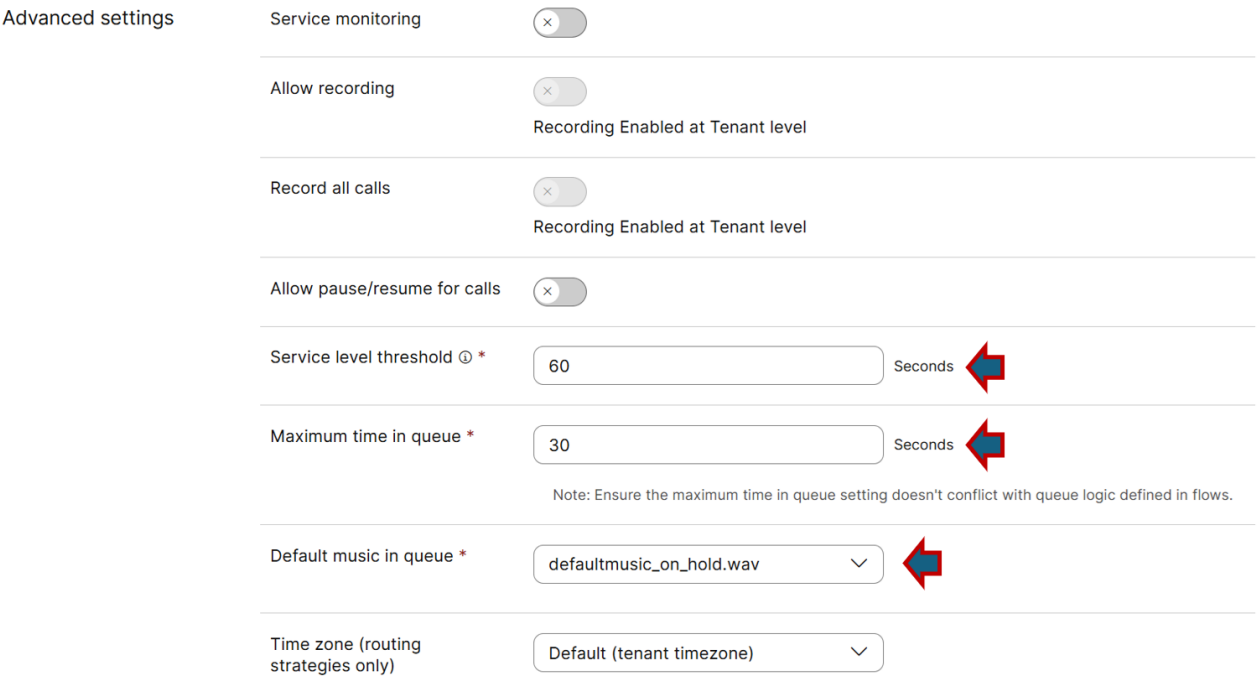


- For, Call Distribution: In group 1, click on the action and ensure the team **WebexOne_Team_[num]** is added.



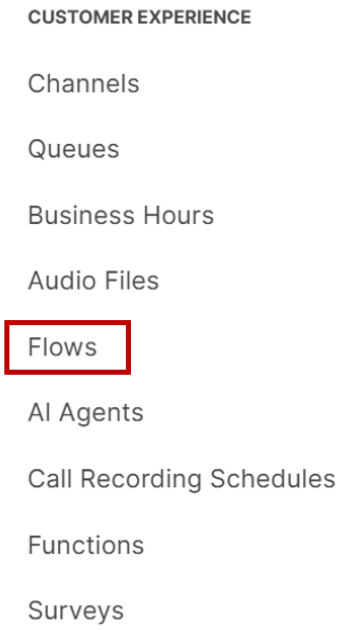


- In Advanced Settings, ensure that these values are accurate:
- Service Level Threshold: **30 seconds**
- Maximum Time in Queue: **30 seconds**
- Default Music in Queue: **defaultmusic_on_hold.wav**

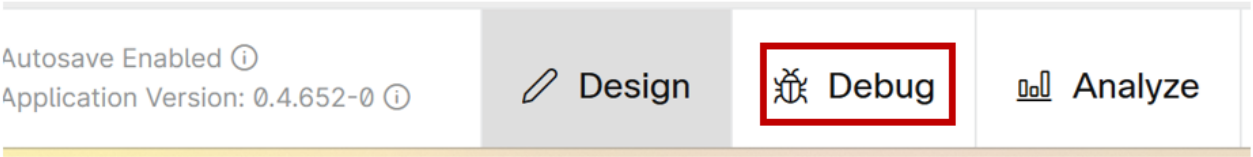


- Click Save to finalize the queue.

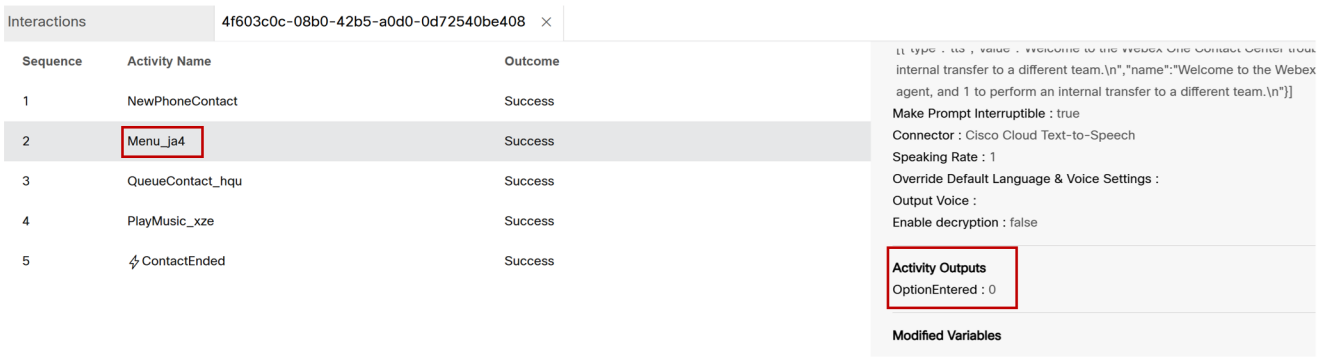
- It's time to now test the call. To do this, log in to the Agent Desktop using the provided credentials.
- URL: https://desktop.wxcc-us1.cisco.com/
- Username: **Contact the lab proctor if information is unavailable.**
- Password: **Contact the lab proctor if information is unavailable.**
- Please select **Desktop** as the telephony option, set the Team as **WebexOne_Team_[num]** , and log in.
- Ensure that the agent is in **Avaialable** Status on the desktop.
- Place a call from your cell phone to the Dialed Number assigned to your entry point and select option 0 to reach the logged-in agent.
- However, you will notice the call is never presented to the agent, and the caller hears music on hold.
- Let's troubleshoot to see why the call is not reaching the agent. The best way will be to look at the flow to see what is happening with the call.
- In the browser tab/window containing Contact Center navigation pane select "Flows" under customer experience section and search for the flow you have mapped to the entry point, **WebexOne_Flow_[num]**



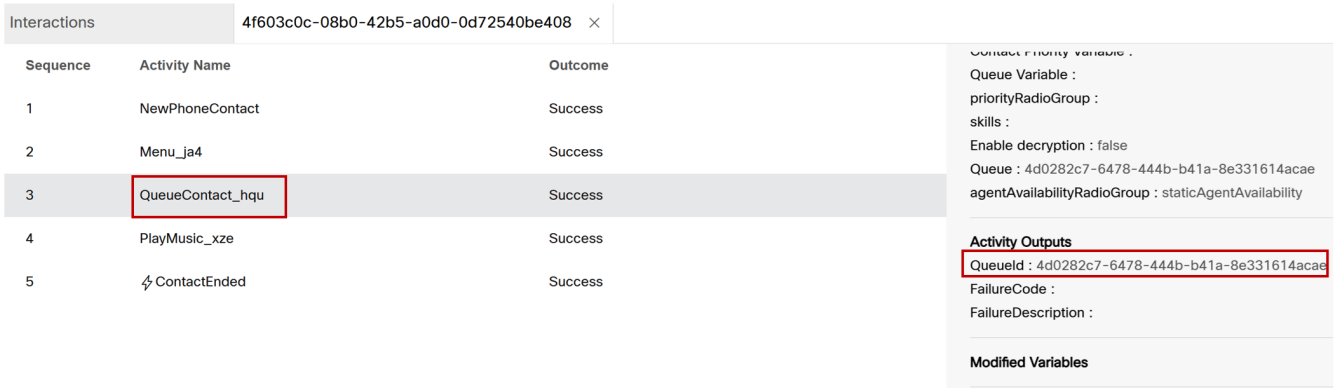
- Click on the "Debug" option and review the last call.



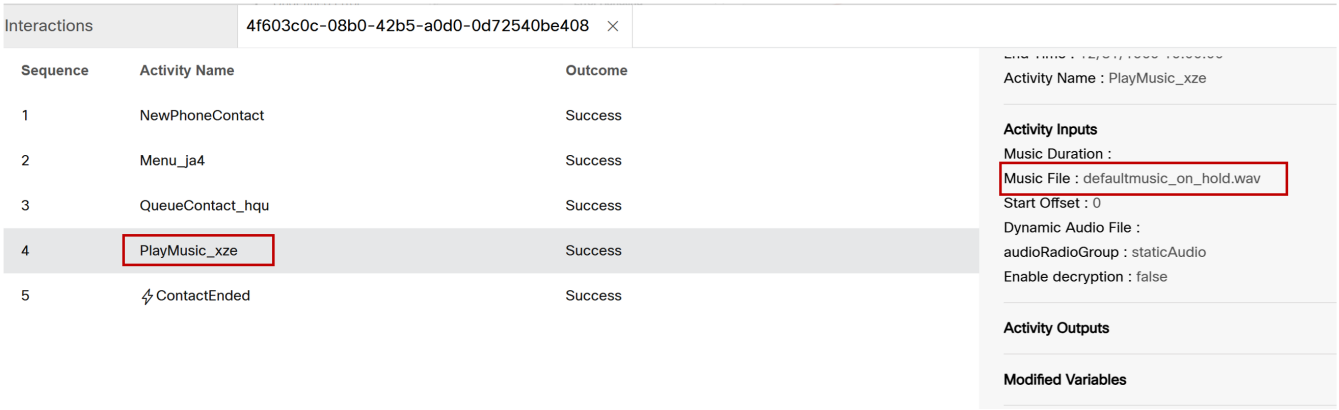
- You will notice that the call passed through the "Menu" node, where option "0" was pressed.



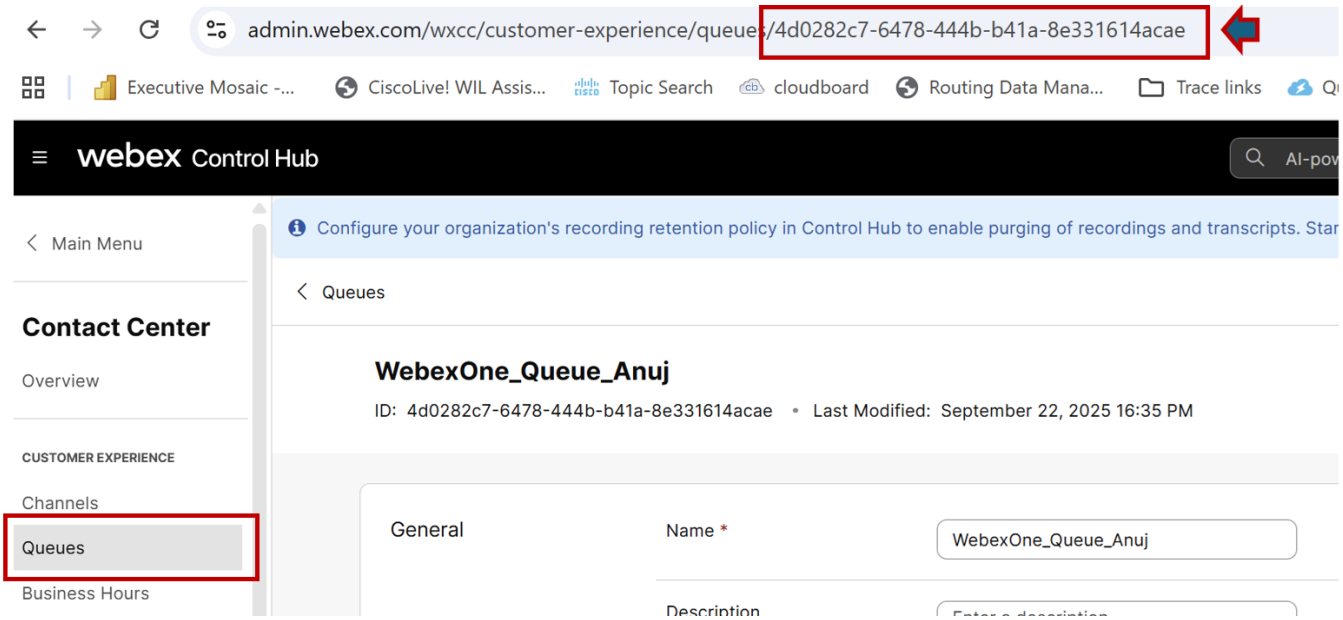
- Then, the call landed in the queue node where the selected queue had the ID "4d0282c7-6478-444b-b41a-8e331614acae".



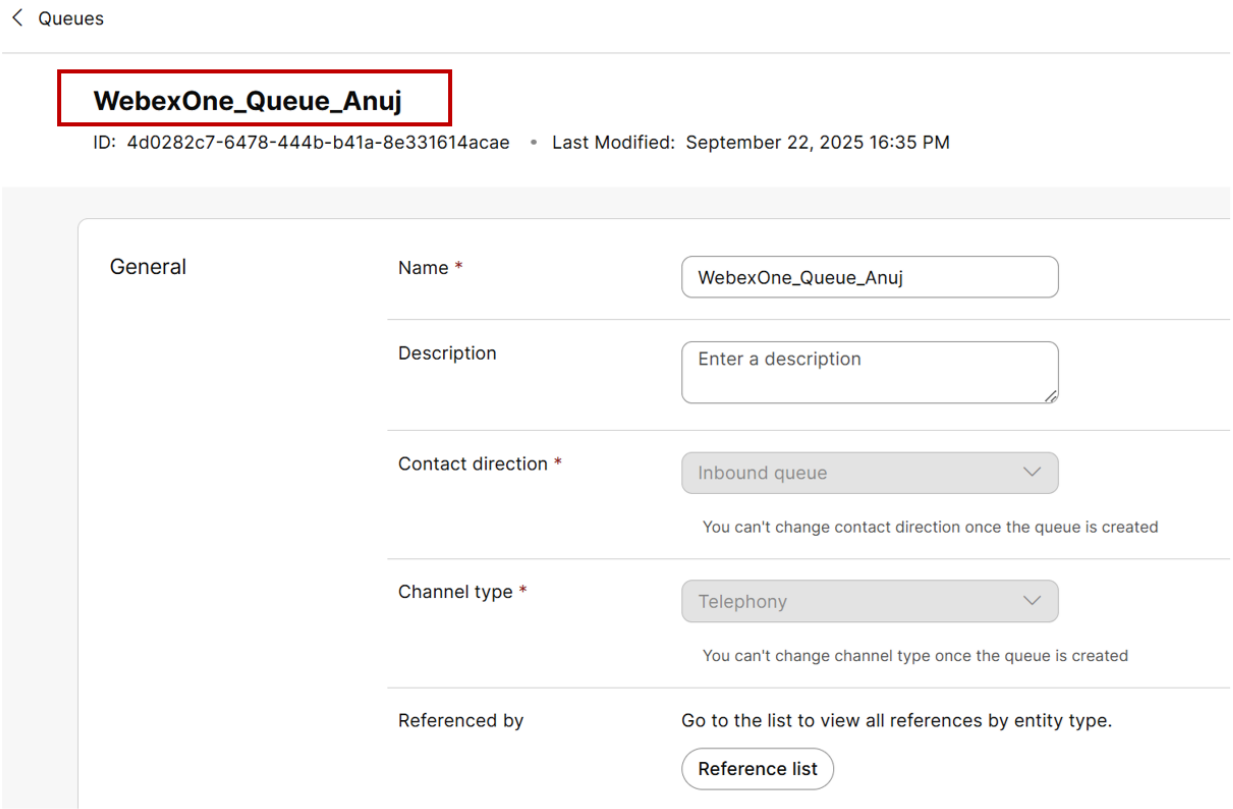
- Now, the call moves to the "Play Music" node, where "defaultmusic_on_hold.wav" is played instead of being presented to the logged-in agent.



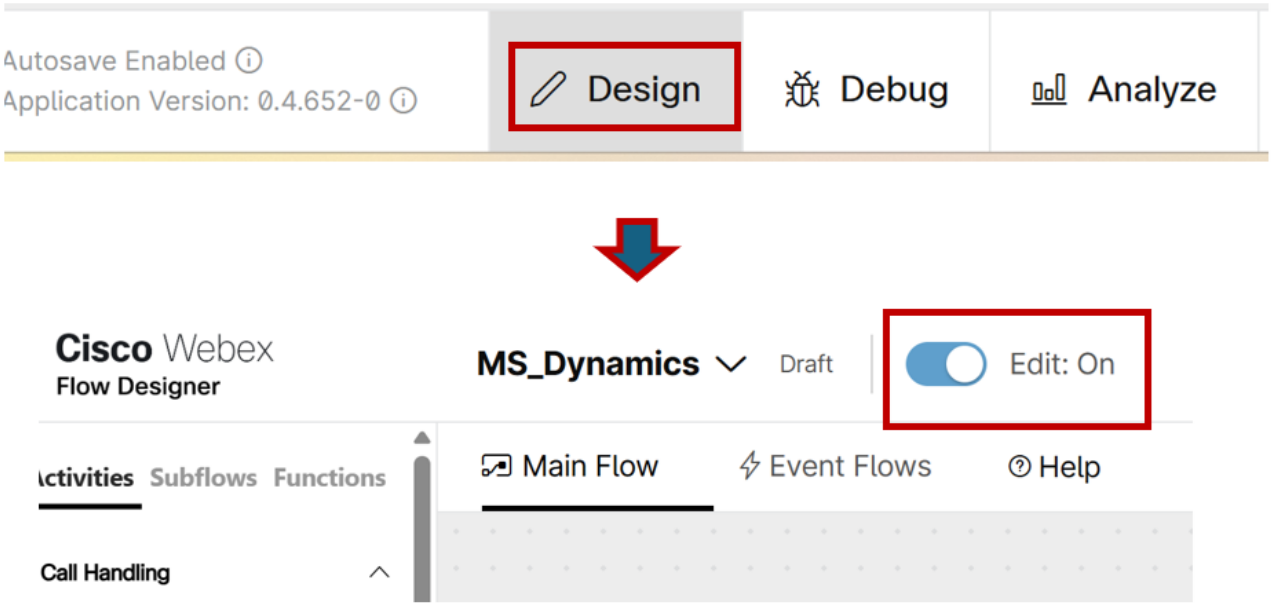
- To figure out what queue it is in the Control Hub, go to the "Queues" section and in the URL, append the queue ID "/4d0282c7-6478-444b-b41a-8e331614acae" noted before and load the page.



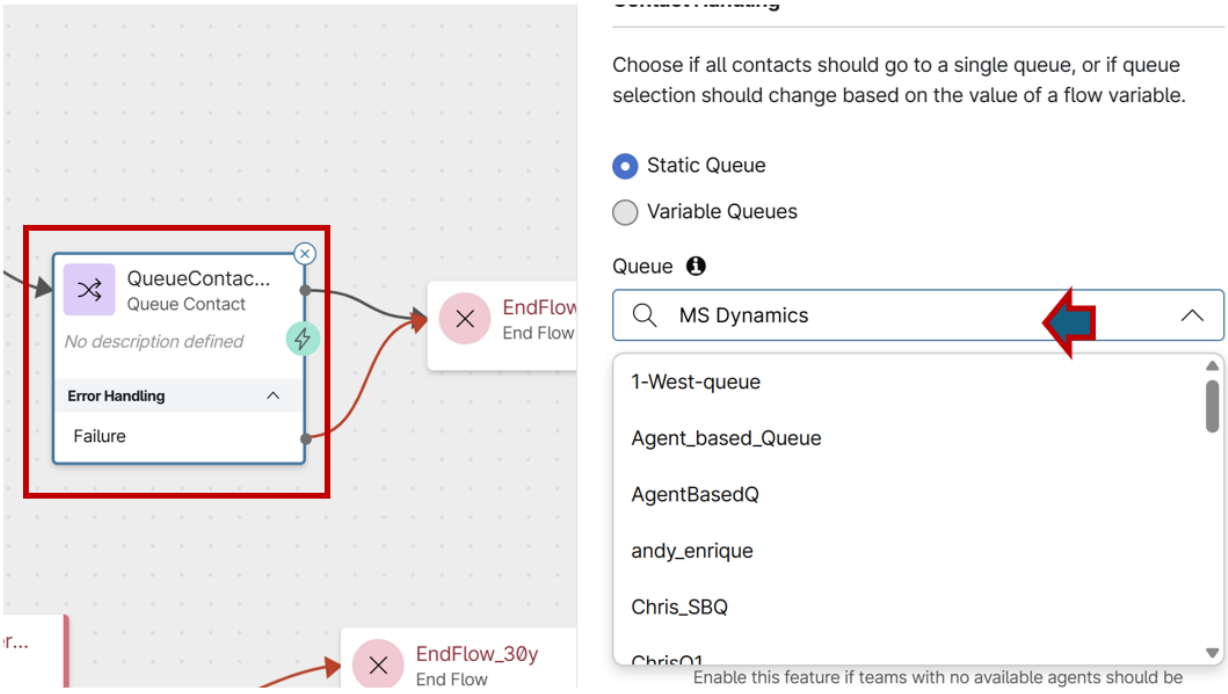
- The queue page that loads is "**WebexOne_Queue_Anuj**" which is not the correct queue for the agent who is logged in.



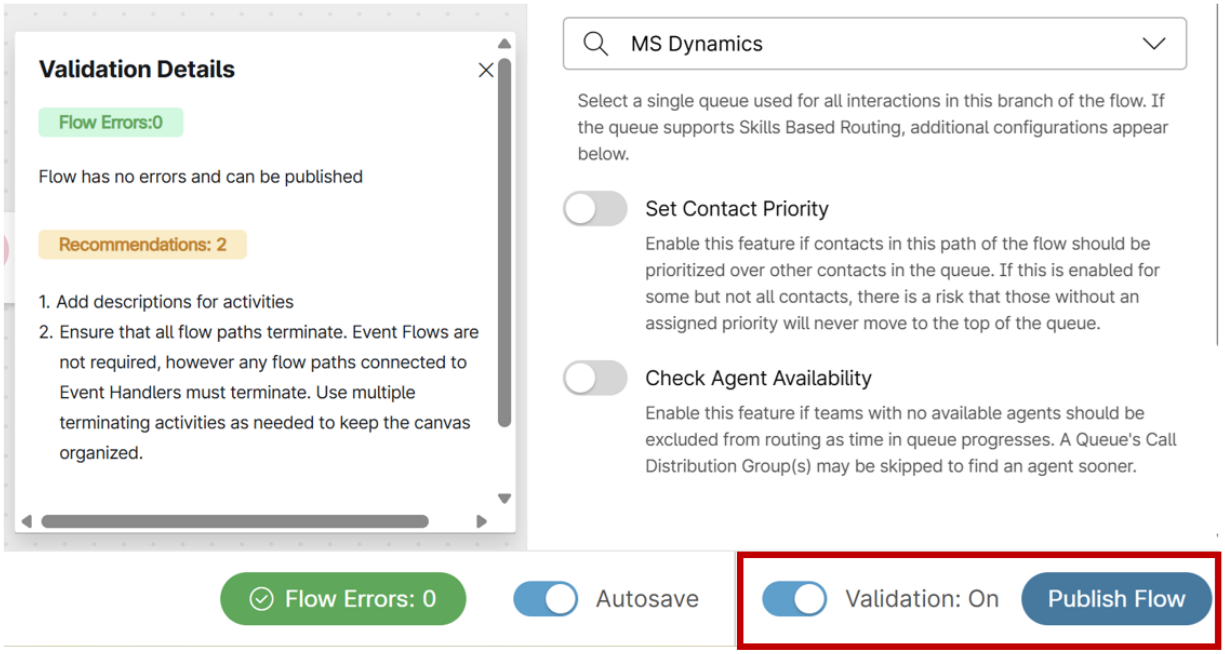
- To correct this, come back into the Design section of the flow and click on the "Edit" option in the flow.



- Select the "Queue contact" node, and from the drop-down menu, select the queue that has your team and the agent i.e **WebexOne_Queue_[num]**

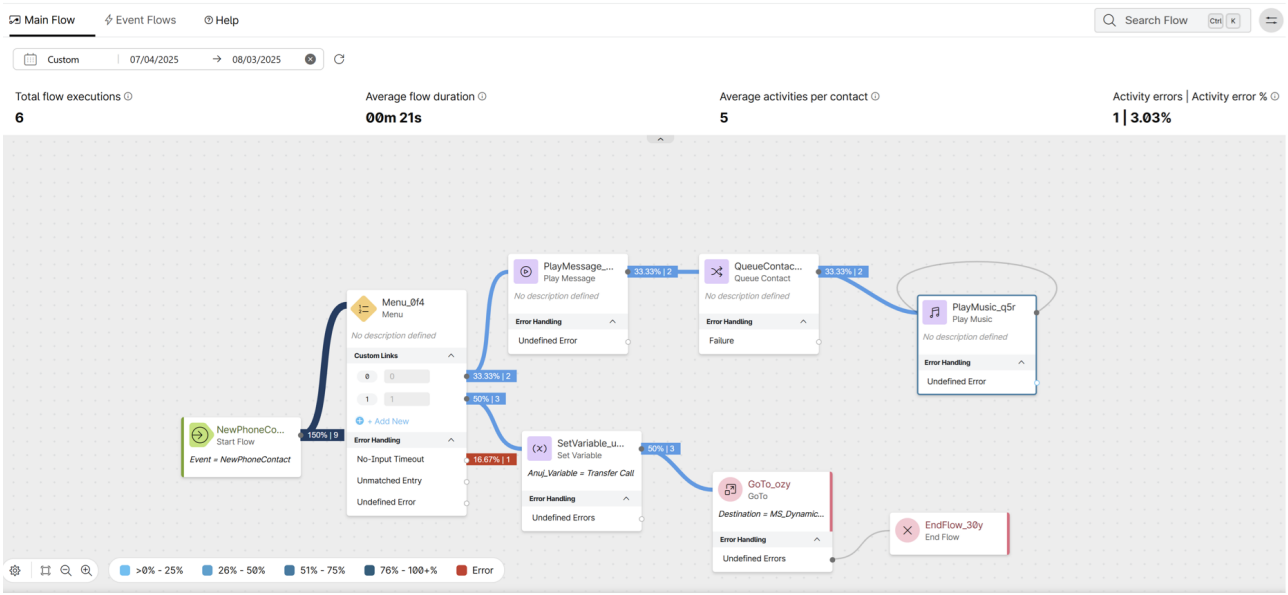


- Toggle "Validation" to "On" to ensure there are no validation errors, and then publish the flow.



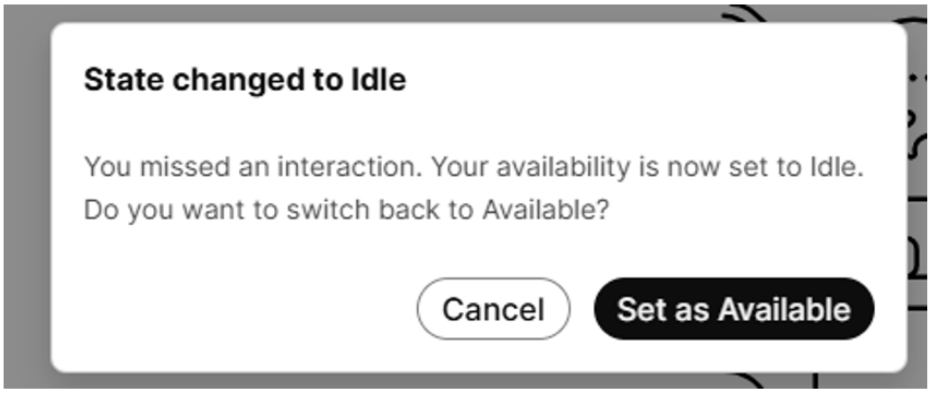
> **Note**
>
> When publishing, it is important to ensure the label you have set is the same one that was set when the entry point was being mapped to the routing flow.

- Place another call from your cell phone to the dialed number you have assigned and select option 0 to succesfully reach the logged-in agent.
- In the Flow Canvas you can now also check how the calls have travered through valrious nodes by cliking on the Analyze capability and selecting the current day.

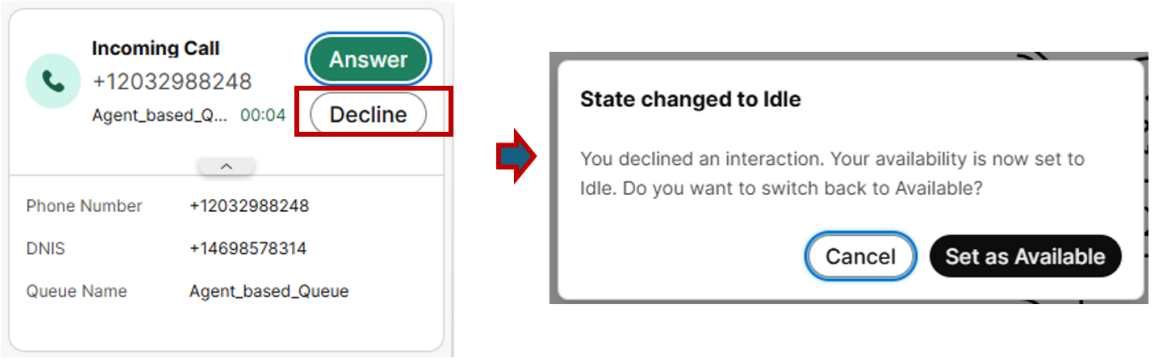


1.2.4 Section 3: Analyze Agent Performance with Call Reports

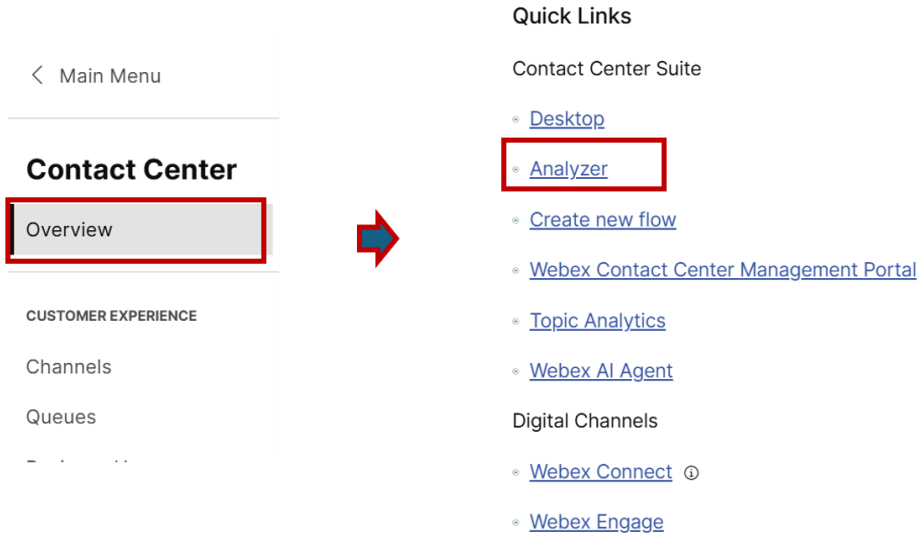
- The goal of this lab section is to review reports to determine what happened with few calls.
- To create a dataset of calls, lets simulate some common agent-side issues.
- First, ensure that the agent is ready and make a call from your cell phone.
- When the call is presented to the agent, do not accept it. Let the call ring for 12 seconds until it goes to a "no answer" state.
- Hangup to disconnect the call.



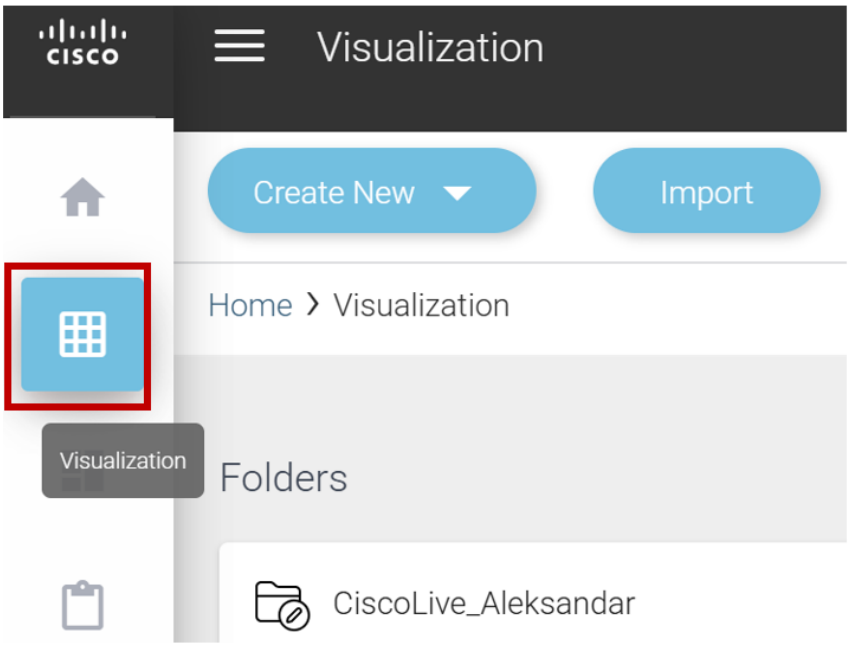
- Next, make another call.
- This time, when the call is presented to the agent, reject the call. Disconnect call from your cell phone.



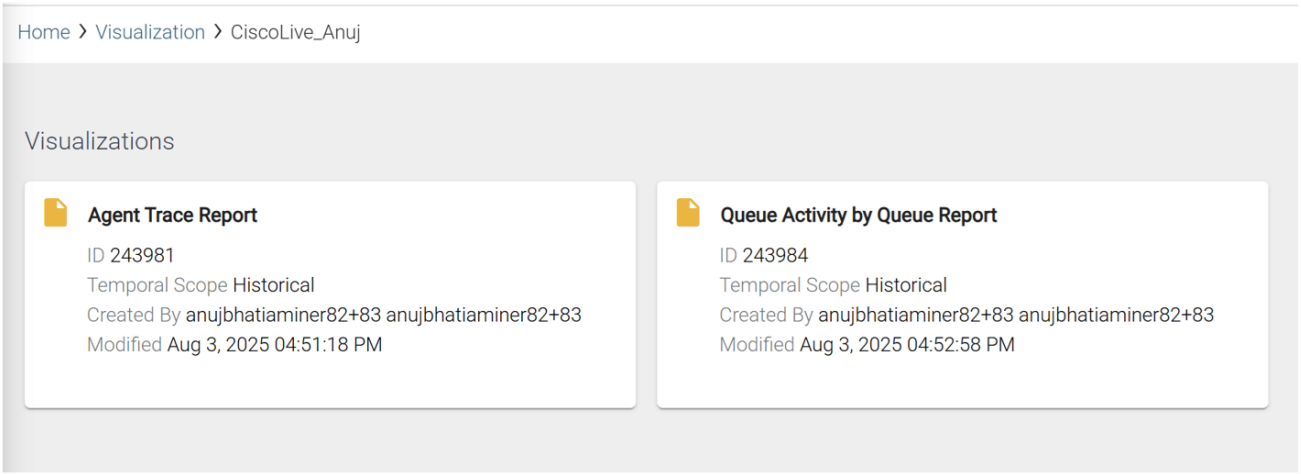
- Now that you have a variety of calls in the system, you can explore the reports to see how to track "**Contact Offer**" and "**Assignment Failures**" to the agent.
- In Control Hub under "Contact Center" select "Overview." In the "Quick Links" section on the right, select "Analyzer."



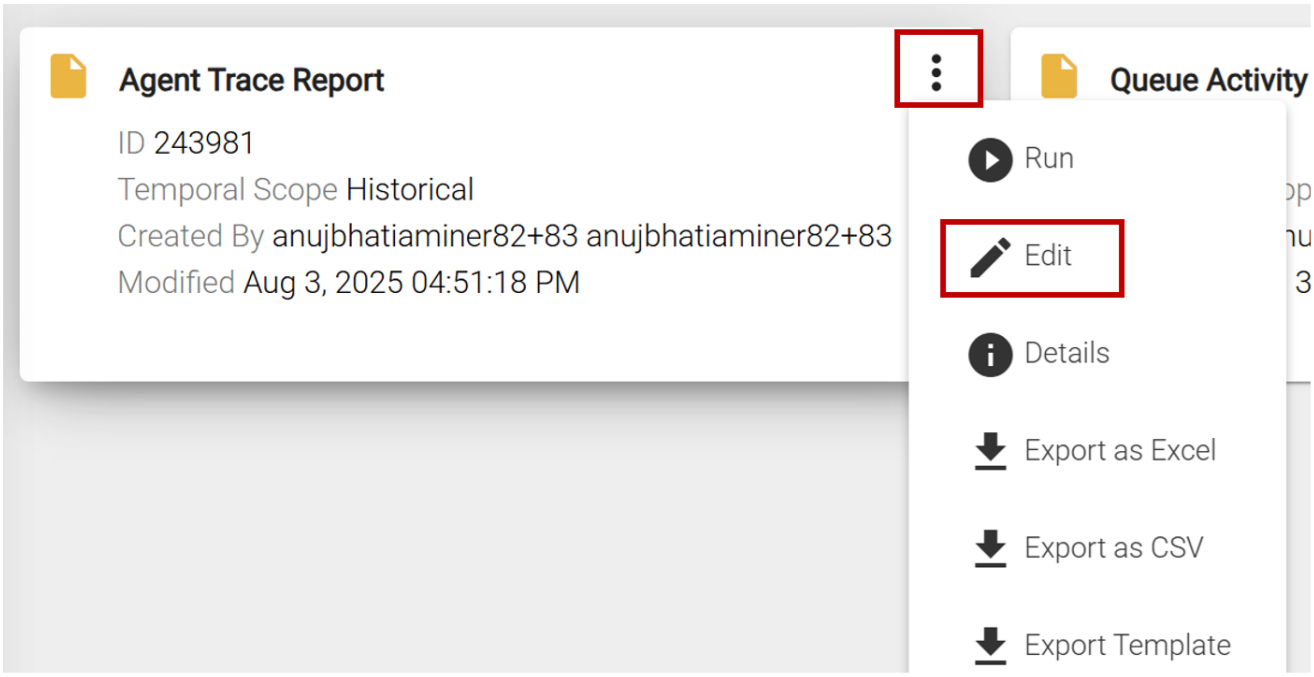
- Click on "Visualization" and double-click the folder **WebexOne_Report_User[num]** containing your user details.



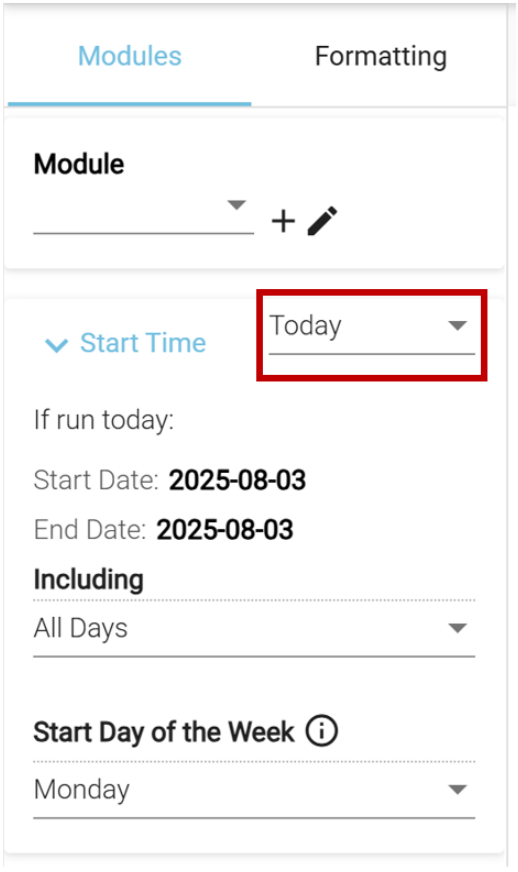
- There are two reports in the folder which are copy of the stock report:
- "**Agent Trace Report**" and
- "**Queue Activity By Queue Report**".



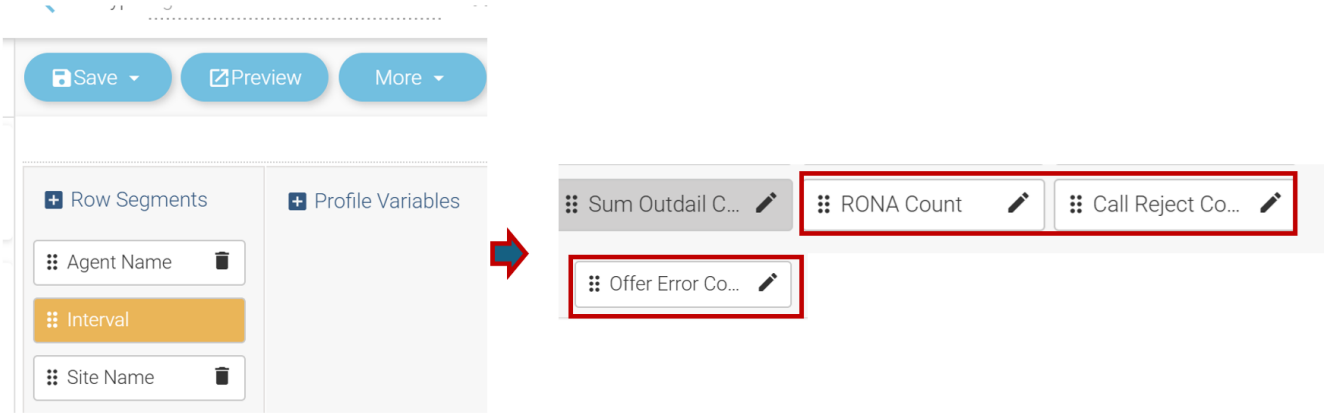
- Each report includes three columns:
- **RONA Count**: This captures all calls that failed because the configured RONA timer expired in WxCC (reasonCode: RONA_TIMER_EXPIRED) or Because of a configured ring timeout on the device (reasonCode: NO_ANSWER_USER).
- **Call Reject Count**: This tracks all calls that failed because the agent explicitly declined the call (reasonCode: USER_DECLINED).
- **Offer Error Counts**: This reflects call offer and assignment failures to the agent caused by any other error.
- To review the "Agent Trace" report, first click on the "Edit" option



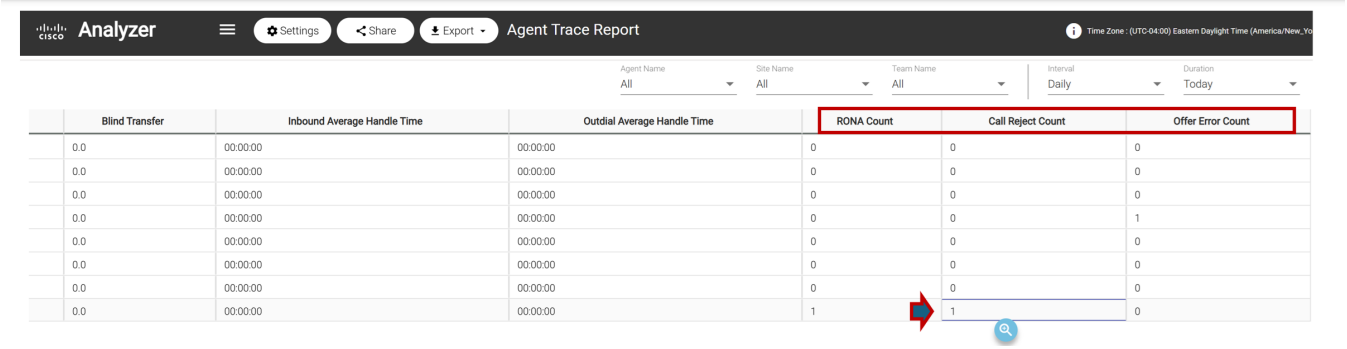
- Ensure that the start time of the report is set for "Today."



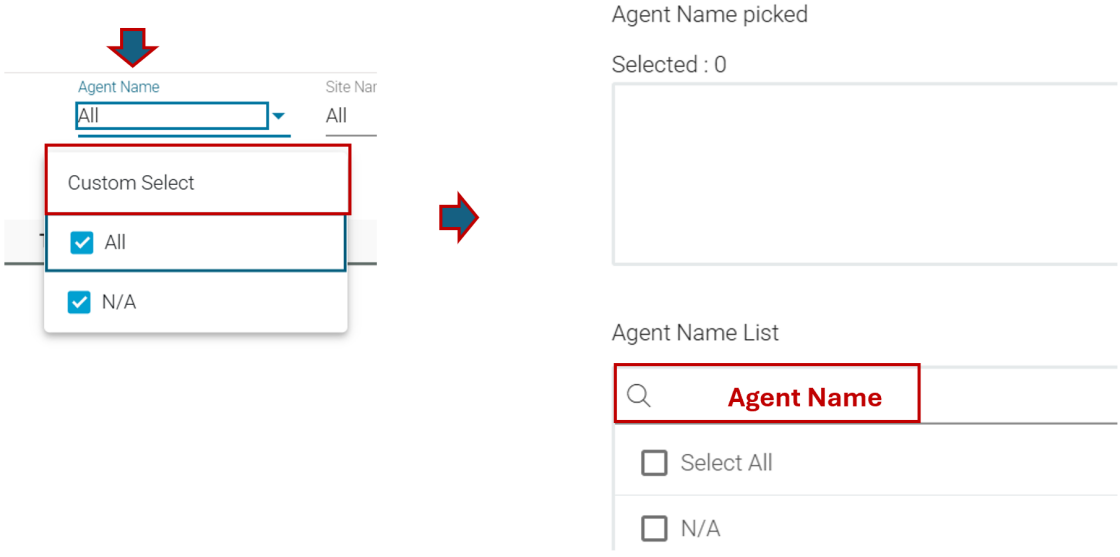
- In the profile variables, you can also confirm that the three variables i.e. **RONA Count**,**Call Reject Count** , and **Offer Error Counts—are** present.



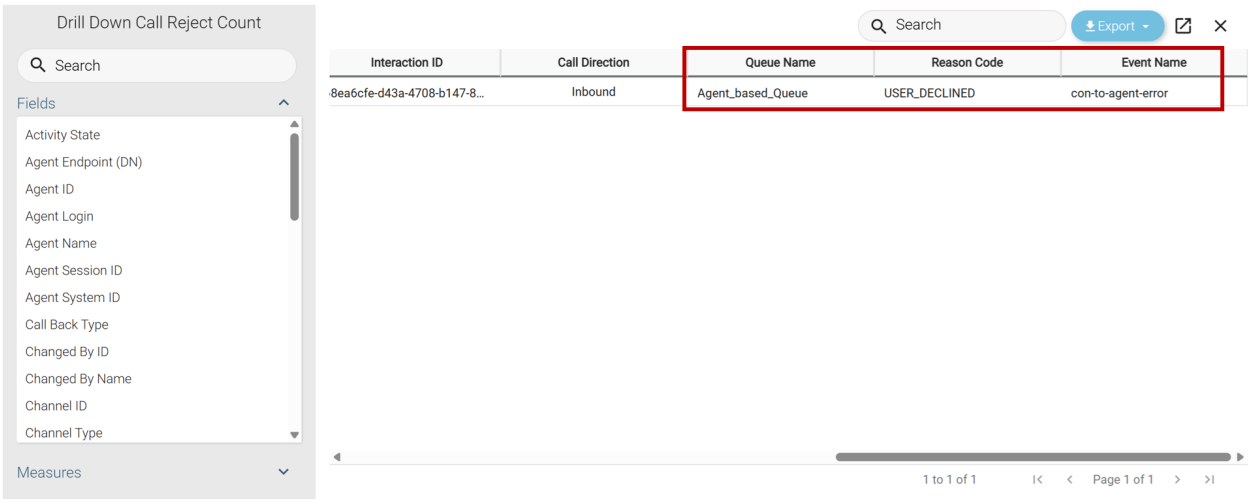
- Save the visualizationand Click the "Preview" option.
- You should now see counts for "RONA" and "Call Reject." Click on the value and the magnification symbol for a further drill-down of the number.



- To easily review the calls in the report, use the Custom Select option in Agent Name to search for the designated user and their calls



- A new window will pop up with all the details of this call. As needed, you can add fields and measures from the available options to get all the desired information related to the call and the agent.



- To test this, you can click on "Agent ID" and it will be appended to the end of the report as a new column.
- The same process can be used to explore the "**Queue Activity By Queue**" report.

Congratulations!! on completing this section of the lab!

You've successfully navigated the complexities of an inbound call flow configuration, debugged common issues, and learned how to analyze agent performance by simulating and tracking different call outcomes in reports.

1.3 Lab 2 - Seamless Call Transfer & Preserving flow Variables

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

`Control Hub URL`	https://admin.webex.com
`Username`	labuser**ID**@wx1.wbx.ai *(where **ID** is your assigned pod number; this ID will be provided by your proctor)*
`Password`	webexONE1!

1.3.1 Objective

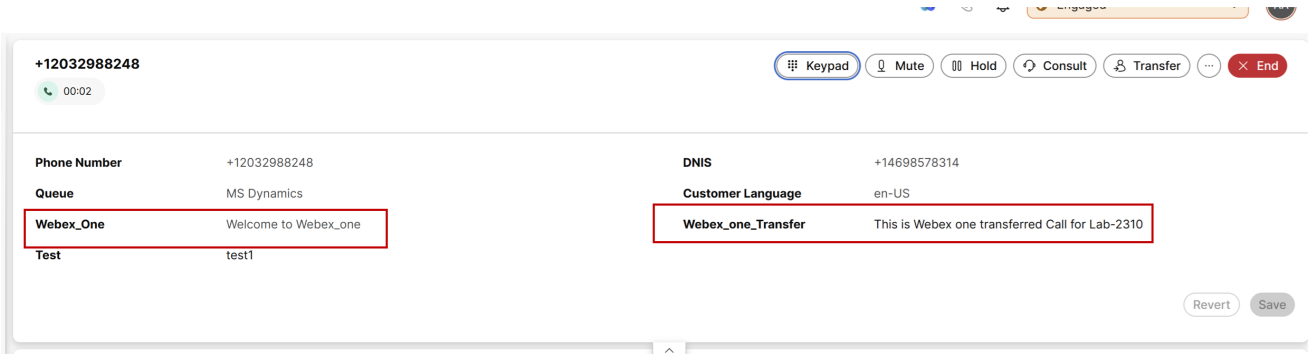
The objective of this exercise is to correct a call flow that currently loses global variables during a transfer.

The current flow is as follows:

- A customer calls and presses option 1, which transfers the call to another department. This transfer causes the call to leave the system and re-enter through a new entry point, creating a new call leg. Within this new flow, the customer presses the user ID option to reach the correct team and agent.
- This process presents two main issues for the business:
- The call is unnecessarily leaving and re-entering the system.
- All original global variables are lost during this transfer, which prevents the agent from seeing crucial information.
- The task is to correct this flow so that the call is transferred seamlessly to the correct destination while preserving all original global variables for the agent.

1.3.2 Section 1 : Experience the Issue

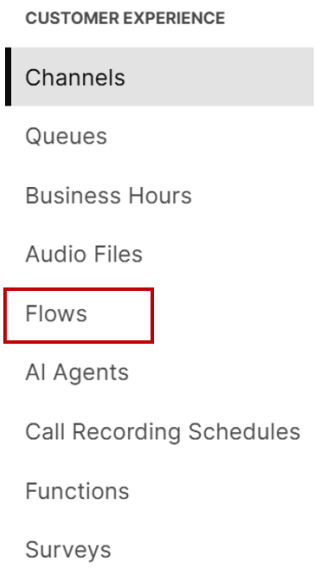
- Ensure that the agent is logged into the desktop and has an "Available" status.
- Call the same entry point number that was configured in Lab 1 for an inbound call and press option 1.
- You will hear a **ringback** and a menu prompting you to enter your user ID.
- The **ringback** you hear is the **first issue**, as it indicates a new call is being made and hitting a transferred entry point (DN).
- Once you enter your user ID, the call will be presented to the agent.
- Accept the call.
- On the desktop, you should see two flow variables, "**Wbex_one_Tansfer**" and "**Webex_one**" However, you will notice that they are not appearing.



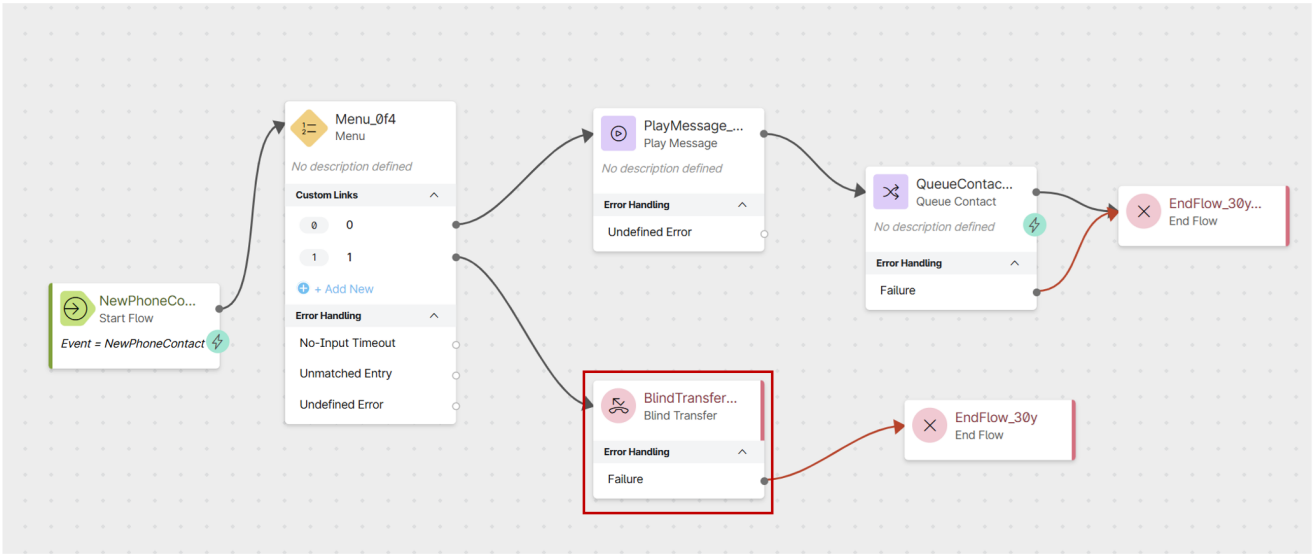
- **This is the second problem**: The original flow variables collected in the first flow are being dropped during the transfer.

1.3.3 Section 2 : Inspect the Flow

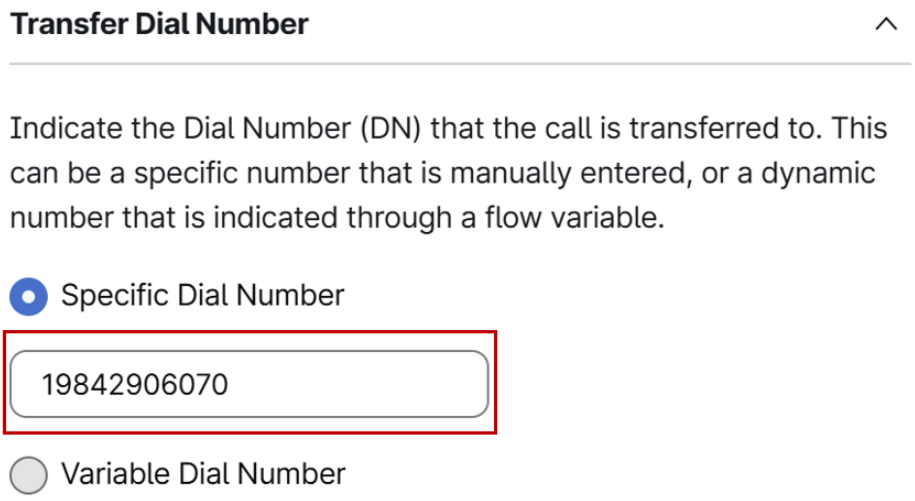
- To understand why this is happening, let's inspect the first flow.
- In the Customer Experience section of Control Hub, select "Flows" and search for the flow you have mapped to your entry point, "**WebexOne_Flow_[num]**."



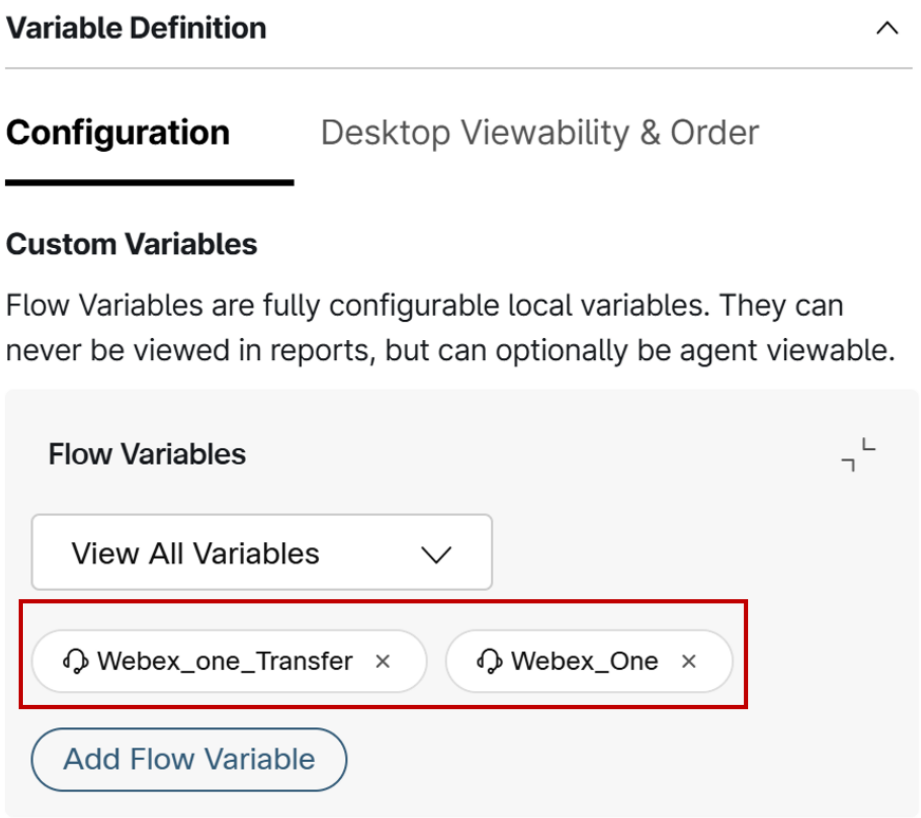
- In the menu node for Option 1, you will notice that it maps to a "**Blind Transfer**" node.



- Select the Blind Transfer node, and you will see it is transferring the call to a number "**19842906070**", which is mapped to a different flow called "**WebexOne_Flow_Transfer**"



- This is the core of the problem. We are instructing the flow to transfer the call to an external number, which creates a new call leg.
- As a result, the variables as well defined in this flow are not being passed to the "**WebexOne_Flow_Transfer**" flow.
- To check these variables, click on any empty space on the flow canvas and review the right side of the flow in **Variable Definition** section.



1.3.4 Section 3 : Correct the Flow

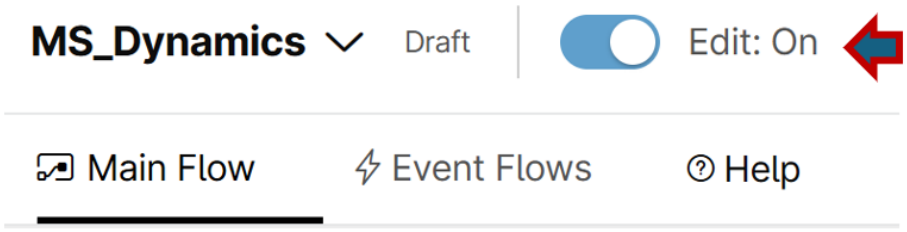
- To rectify this, we need a node that handles internal transfers more effectively.

- The Flow Designer provides a "GoTo Node" specifically for this use case.

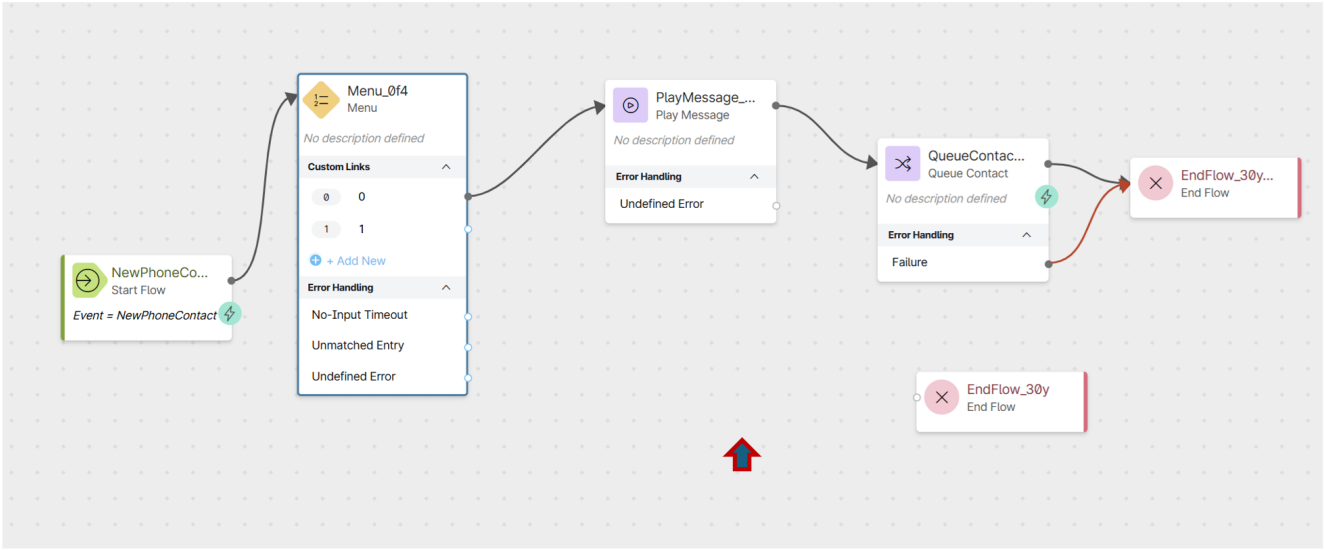
Note

A GoTo Node is used to seamlessly transfer a call to another flow within the same system, preserving variables. A Blind Transfer node is used to transfer a call to an external number.

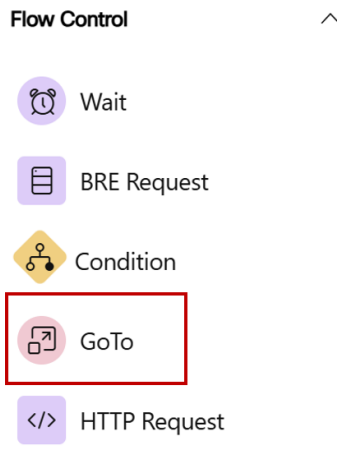
• To replace the Blind Transfer node, first enable the edit mode of the flow designer.



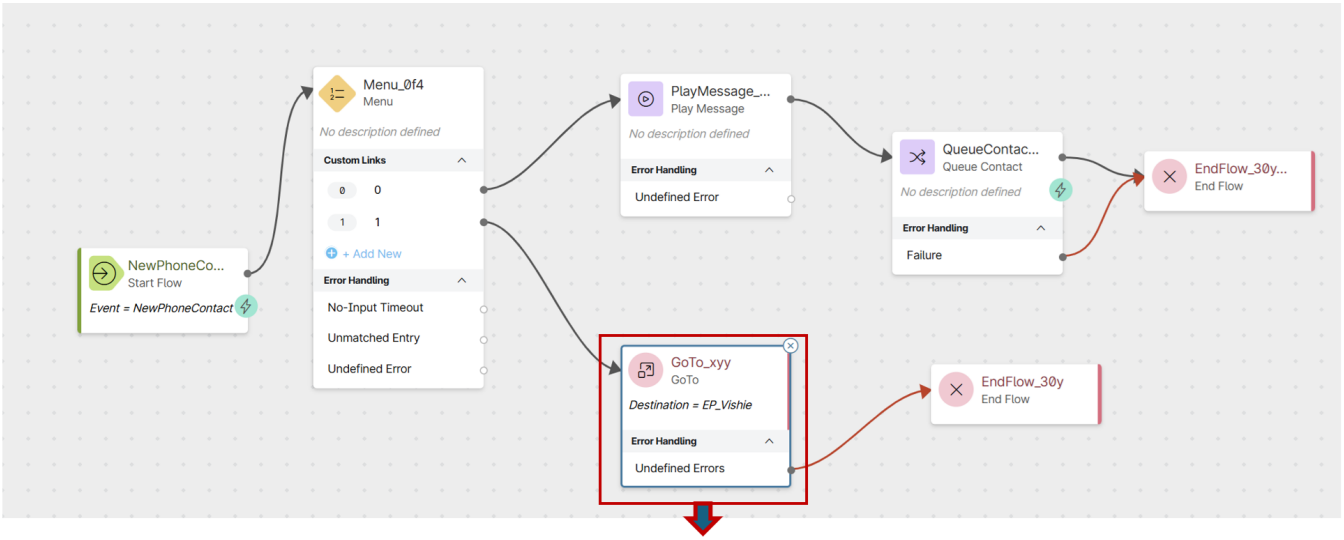
• Delete the Blind Transfer node by selecting it and pressing the delete key.

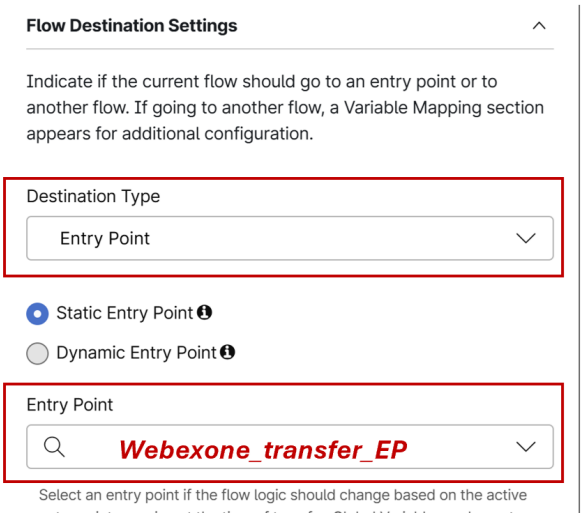


• From the Flow Control section, drag a "GoTo Node" onto the canvas.

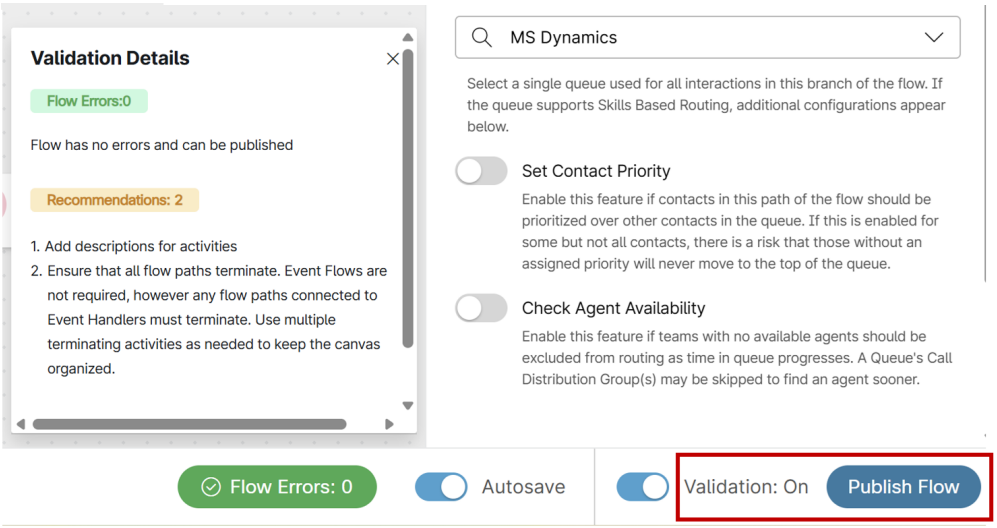


• In the GoTo Node's properties, select "Entry point" as the destination type and map it to "**WebexOne_Transfer_EP**"



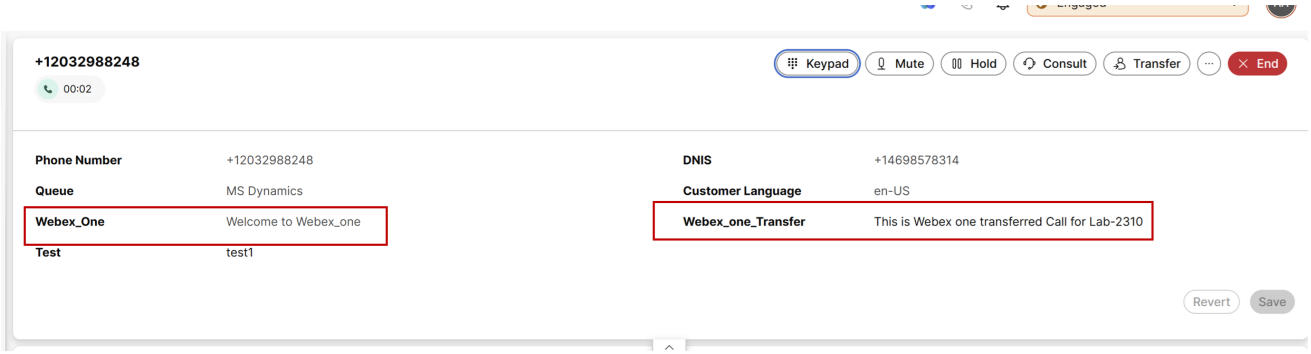


- Connect the Menu node's Option 1 output to the GoTo Node and the "undefined" error output to the "End of the flow" node, as shown in the screenshot above
- Toggle "Validation" to "On" to ensure there are no validation errors, and then publish the flow.



1.3.5 Section 4 : Verify the Solution

- Ensure that the agent is logged into the desktop and has an "Available" status.
- Call your provided number from your cell phone again and press Option 1.
- You should hear no ringback, but instead, be directly presented with the menu option to enter your user ID.
- This resolves the first problem of creating a new call leg for the transfer.
- Once you provide the user ID, the agent should receive the call.
- After the call is accepted, you should now see the two variables defined in the first flow on the agent desktop.



- This confirms that the second problem of variables being lost has also been resolved.

Congratulations !! on successfully completing this exercise!

You've learned how to perform seamless internal transfers within entry points and now understand the crucial difference between a Blind Transfer and a GoTo Node.

1.4 Lab 3 - Set Up Outdial Calls in WxCC

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

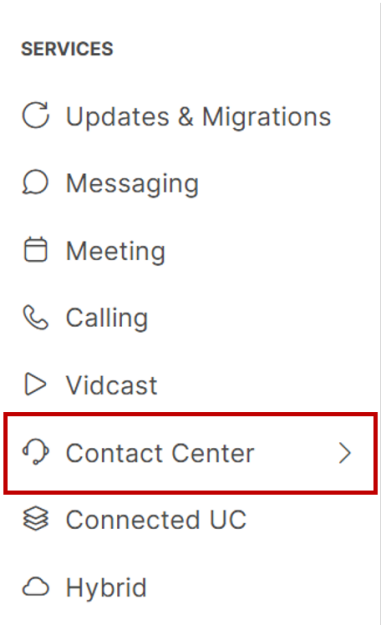
`Control Hub URL`	https://admin.webex.com
`Username`	labuser**ID**@wx1.wbx.ai *(where **ID** is your assigned pod number; this ID will be provided by your proctor)*
`Password`	webexONE1!

1.4.1 Objective

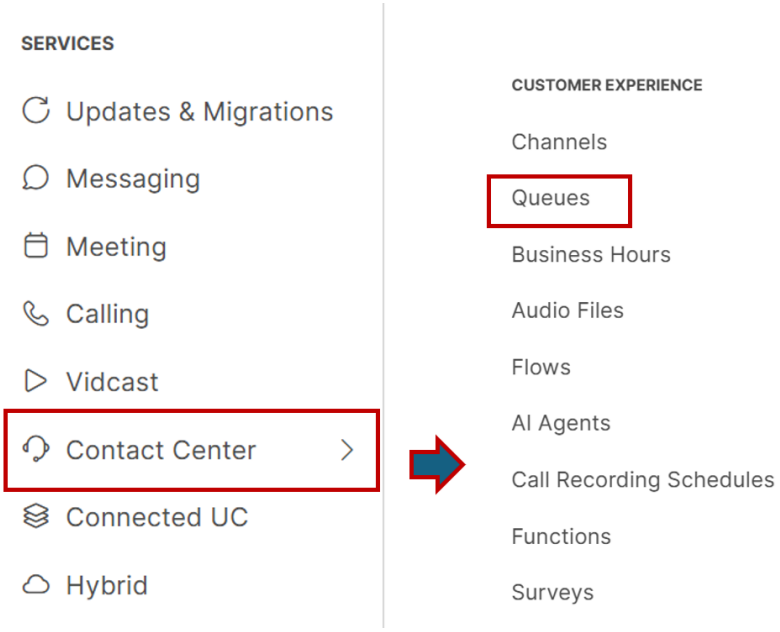
In this lab exercise, the audience will gain hands-on experience configuring WxCC solutions to enable outdial capabilities from scratch. Furthermore, the exercise will also equip participants with essential debugging skills to identify and rectify frequent implementation errors ensuring a robust and functional deployment.

1.4.2 Section 1 : Setup Outdial

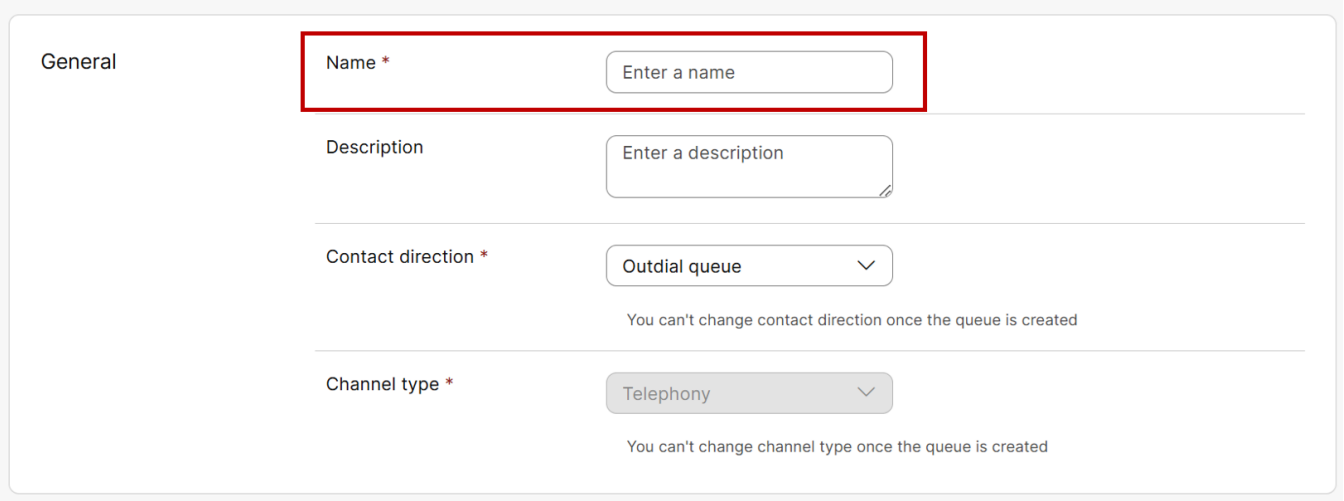
- Log into Webex Control Hub with the provided credentials.
- In Control Hub, navigate to Services and click on Contact Center.



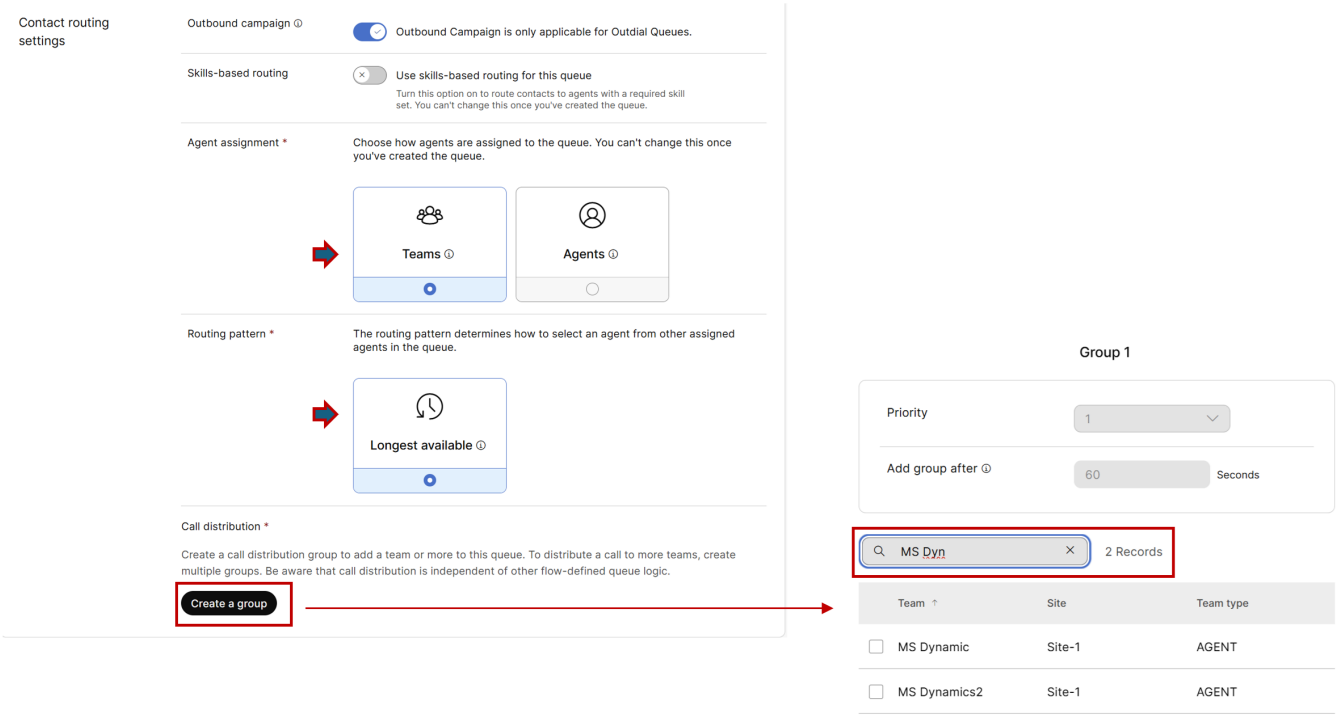
- In the Contact Center navigation pane, under Customer Experience, select **Queues**.



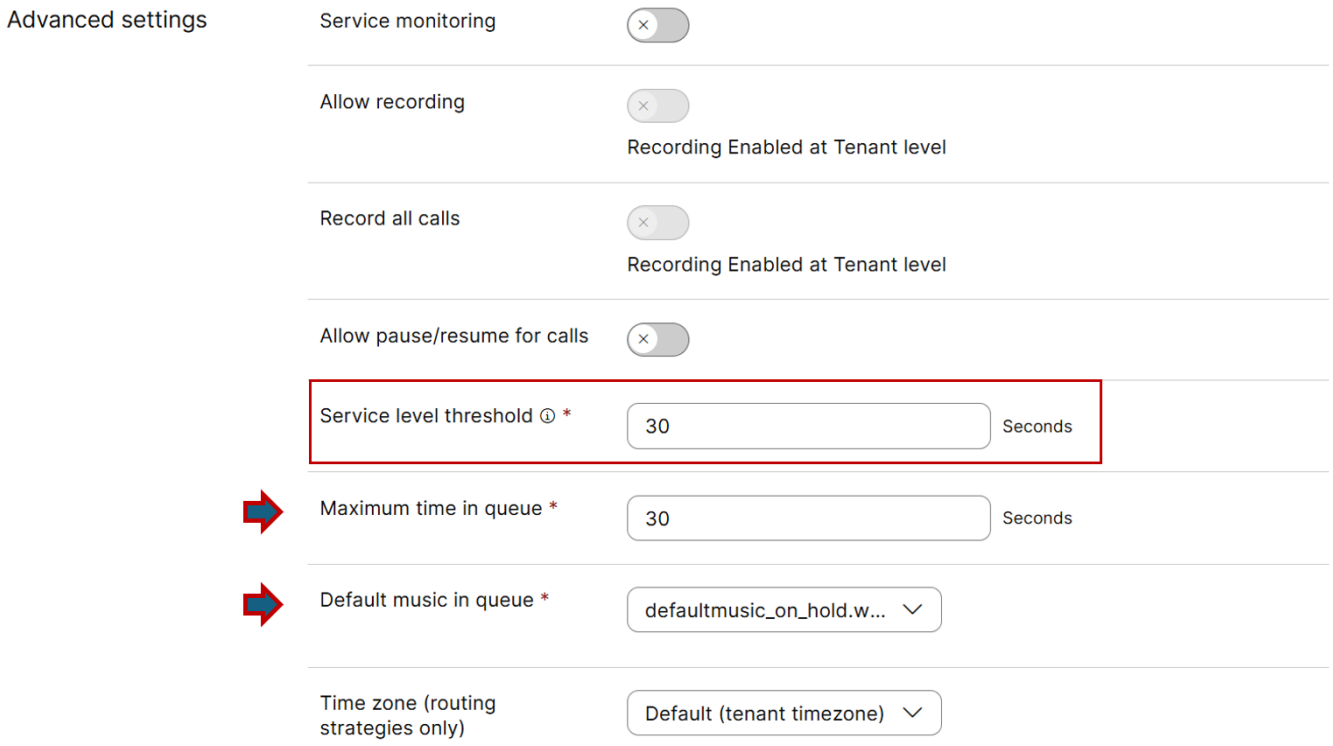
- Create a new queue by clicking on the "Create a Queue" option.
- The Queue Creation Wizard will appear. Provide the following details:
- **General**
- **Name**: [Provide a descriptive name for your queue]
- **Contact direction**: Outdial Queue
- **Channel type**: Telephony



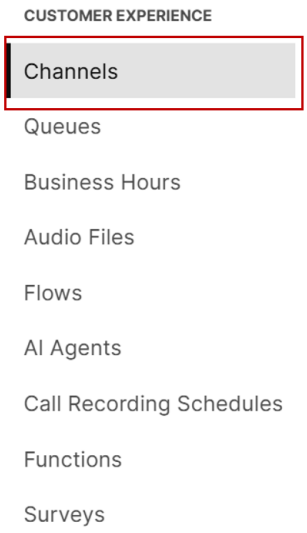
- **Contact Routing Settings**
- **AgentAssignment**: Teams
- **Routing Pattern**: Longest available
- **Call Distribution**: Create a group and add the team **WebeOne_Team_[num]**



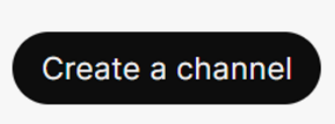
- **Advanced Settings**
- **Service level threshold**: 30 seconds
- **Maximum time in queue**: 30 seconds
- **Default music in queue**: defaultmusic_on_hold.wav



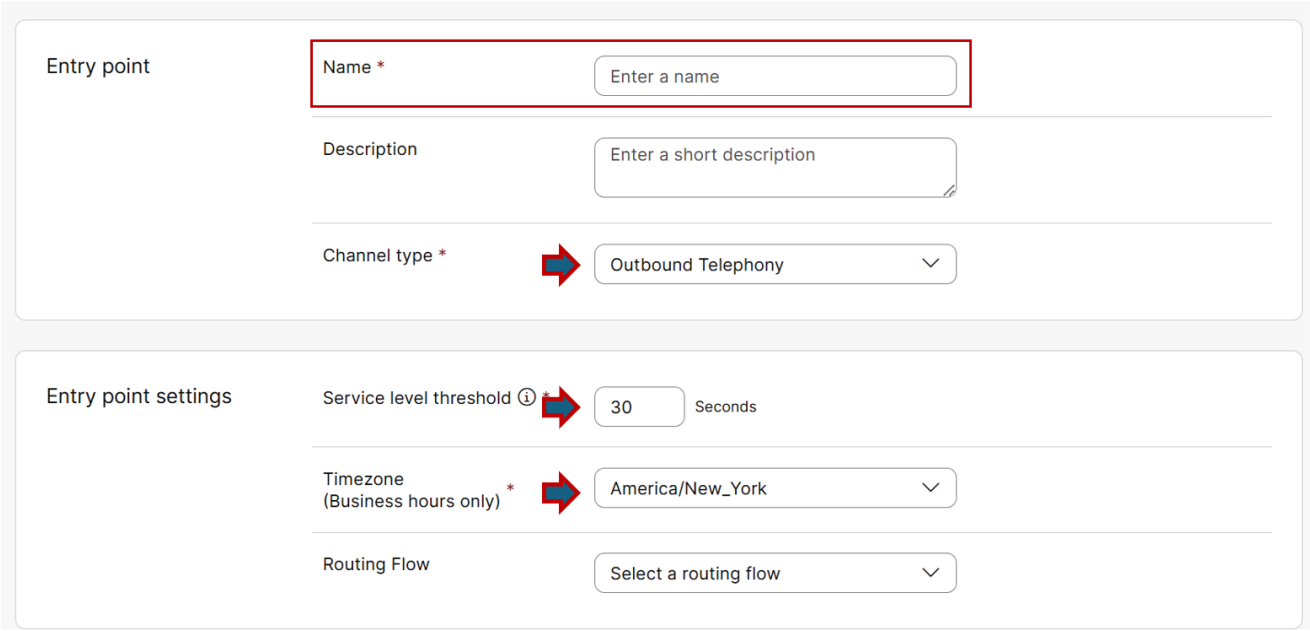
- Once these settings are added, click Create to finalize the queue.
- Lets create a Entry point to map the queue to this entry point.
- Navigate back to Customer Experience in Contact Center and click on **Channels**.



- Create a new channel by clicking on the "Create a channel" option.

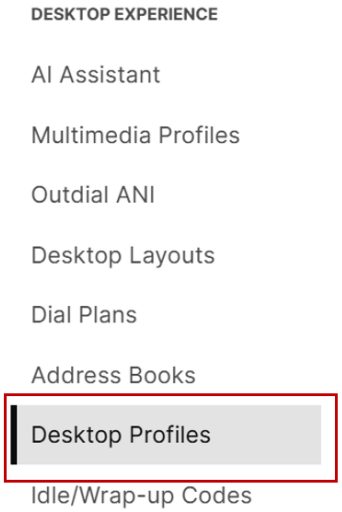


- The Channel Creation Wizard will appear. Provide the following details:
- **Name**: [Provide a descriptive name for your channel]
- **Channel type**: Outbound Telephony
- **Service level threshold**: 30 seconds
- **Timezone**: America/New York

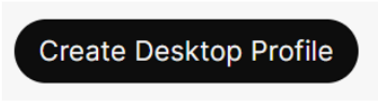


- After these settings are added, click Create to finalize the channel.
- Since outdial is an agent activity, the Agent Desktop should have the capability to call any number outside the WxCC ecosystem.
- To enable this capability, create an agent profile and map it to the agent.

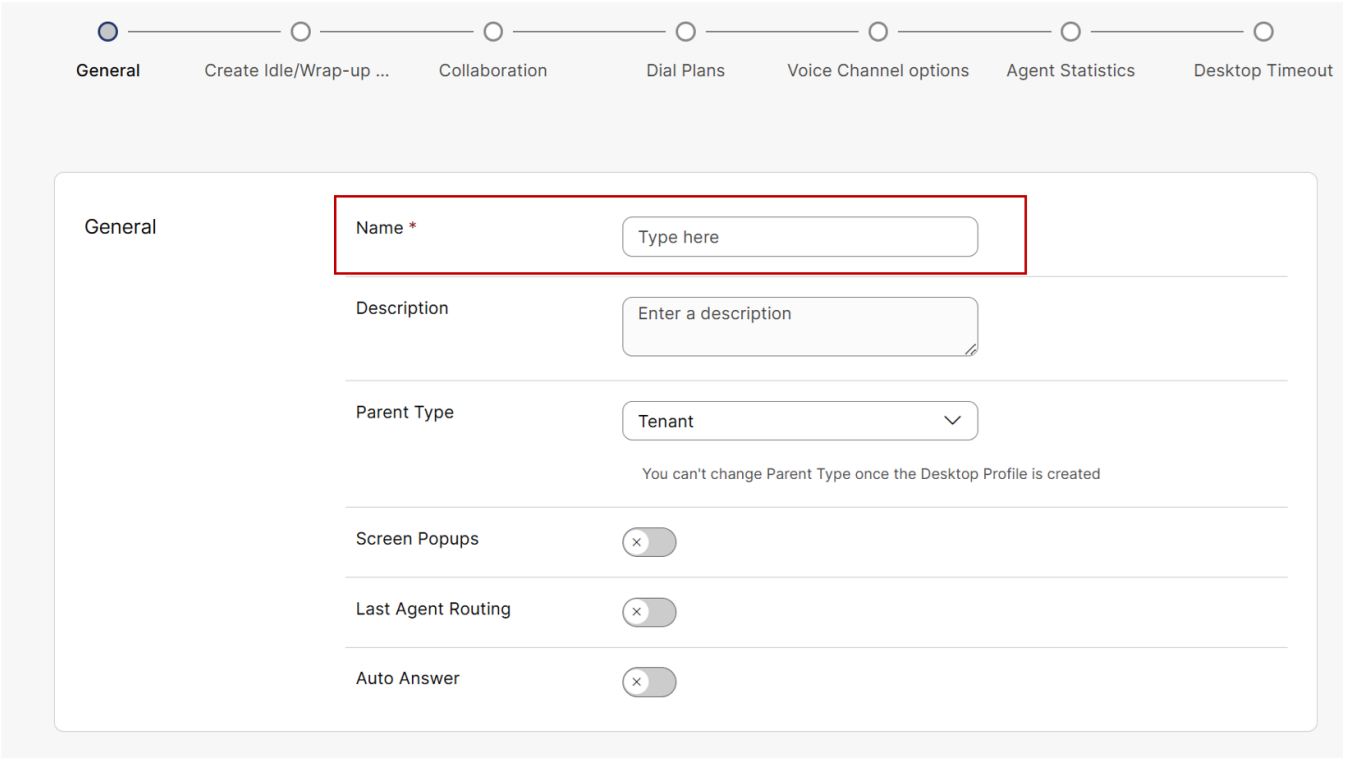
- Navigate to Desktop Experience in Contact Center and click on **Desktop Profiles**.



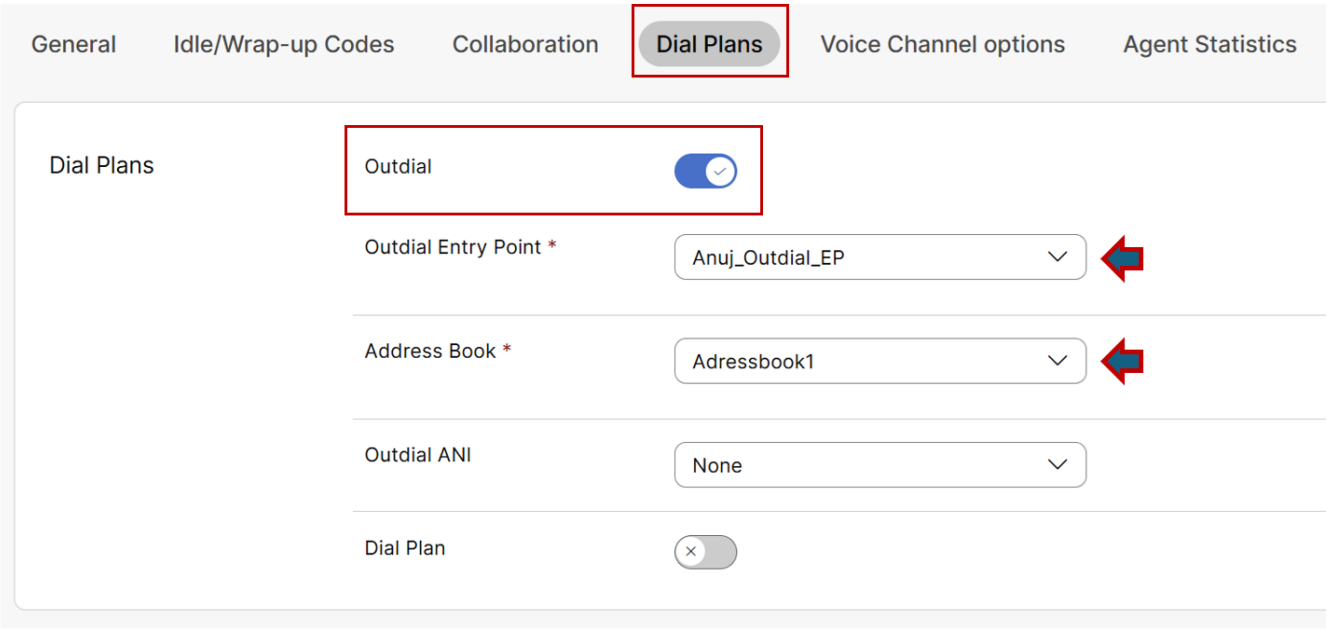
- In Desktop Profiles, create a new profile by clicking on "Create Desktop Profile".



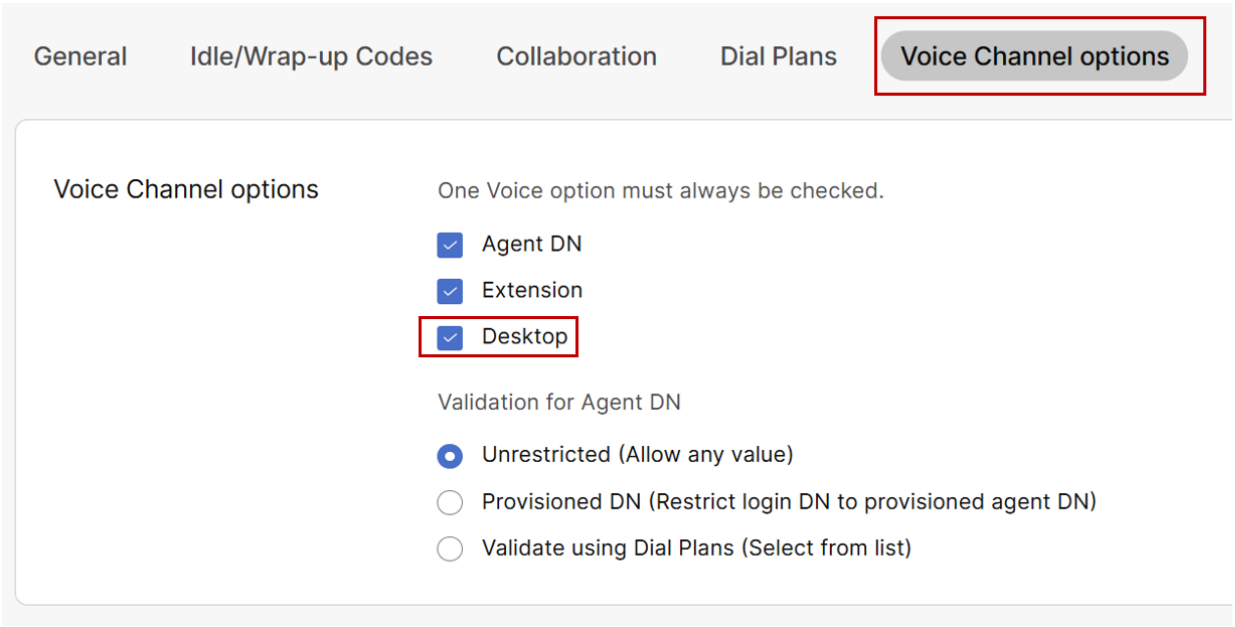
- In the General section, provide the desired name for your profile.



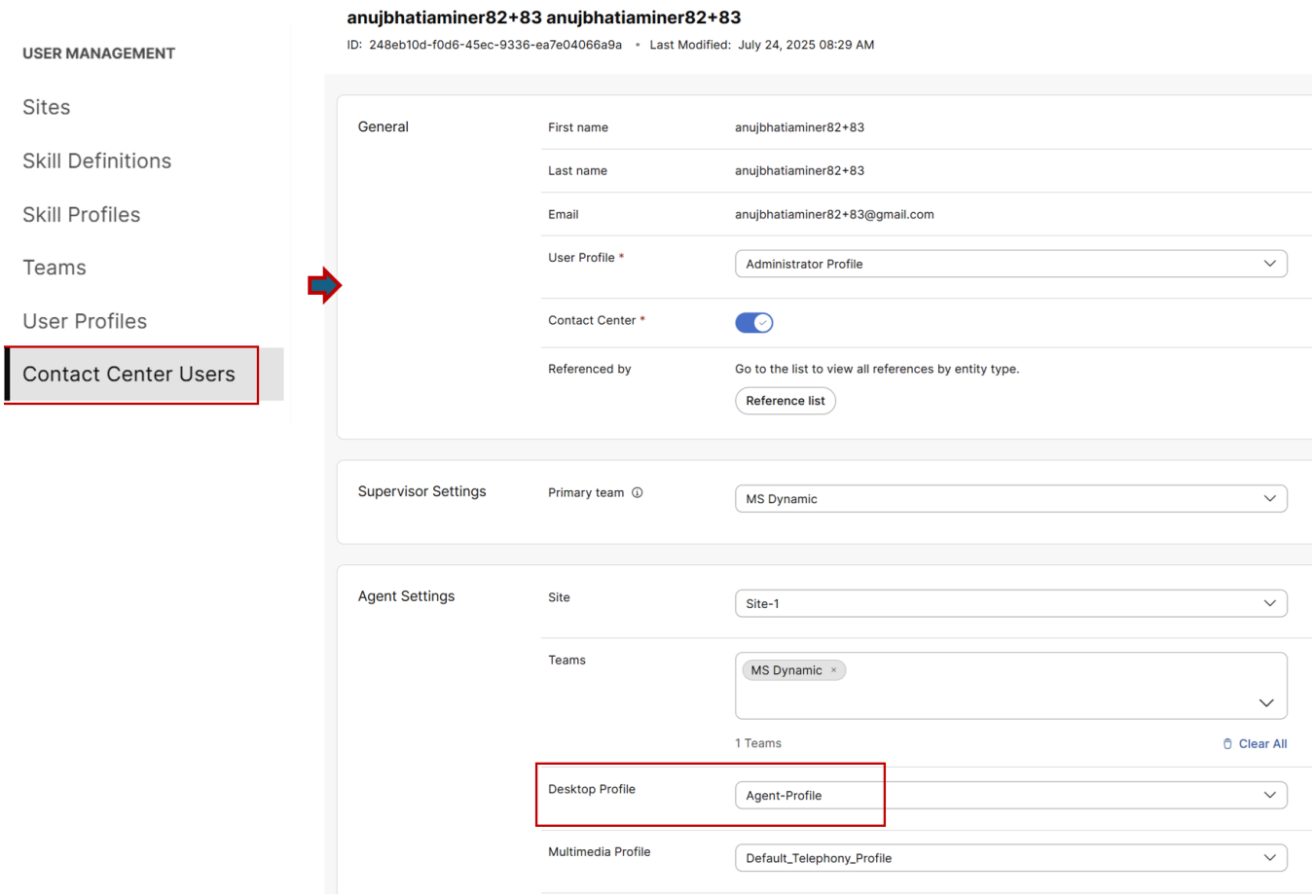
- Move to "Dial Plans" by clicking Next button (at the bottom of the screen) a couple of times.
- Enable "**Outdial**".
- Select the newly create entrypoint as an "**Outdial Entry Point**".
- Select the preconfigured address book "**WebexOne_outdial_AddressBook**" as an "**Address Book**".



- Move to "Voice Channel Options" by clicking Next and ensure that "Desktop" is enabled under "Voice Channels options".

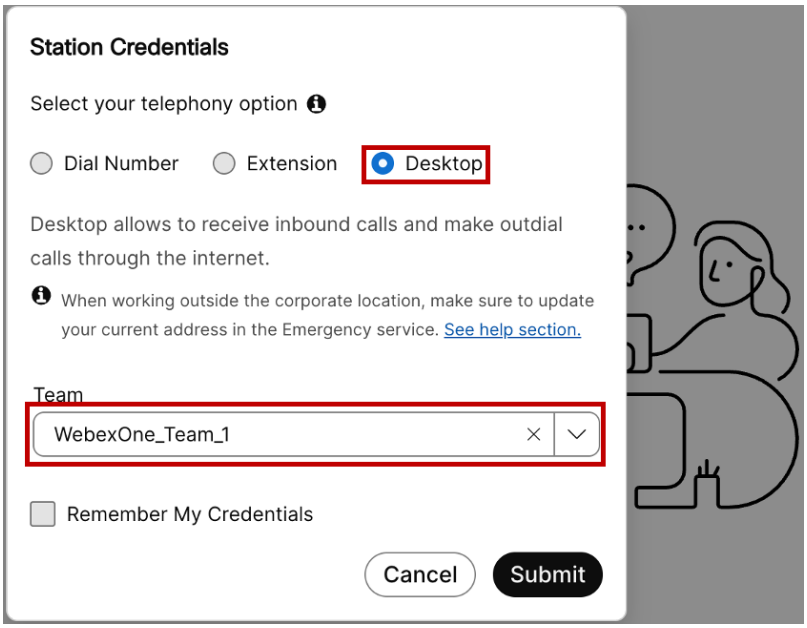


- Proceed to the end of desktop profile creation by clicking Next and finally Create.
- Now, Navigate to the User Management section in Contact Center and click on **Contact Center Users**.
- Bring up your user and assign the newly created desktop profile under "Desktop Profile" and Save changes.



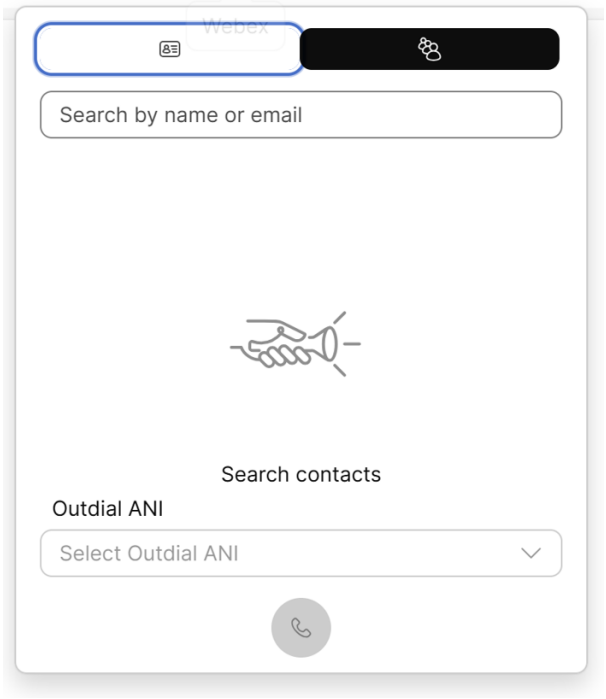
1.4.3 Section 2 : Test Outdial

- Now, log in to the Agent Desktop using the provided credentials.
- **URL**: https://desktop.wxcc-us1.cisco.com/
- **Username**: Contact the lab proctor if information is unavailable.
- **Password**: Contact the lab proctor if information is unavailable.
- Please select desktop as telephony option and set the Team as **WebexOne_Team_[num]** and login.



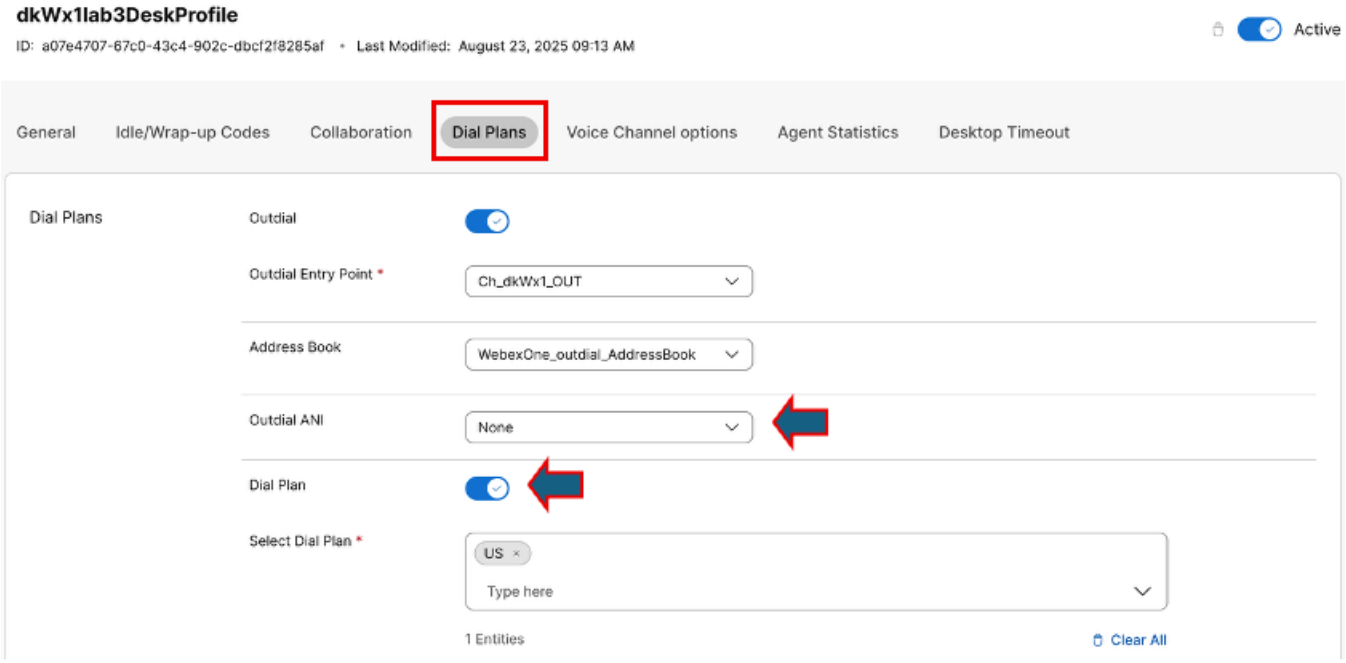
- Present task is to dial your cell phone number.
- First, click the Outdial Call option on the top right corner of the desktop.

- You'll notice that the dial pad is missing; the only available option is to search by name, email, or number within the tenant.

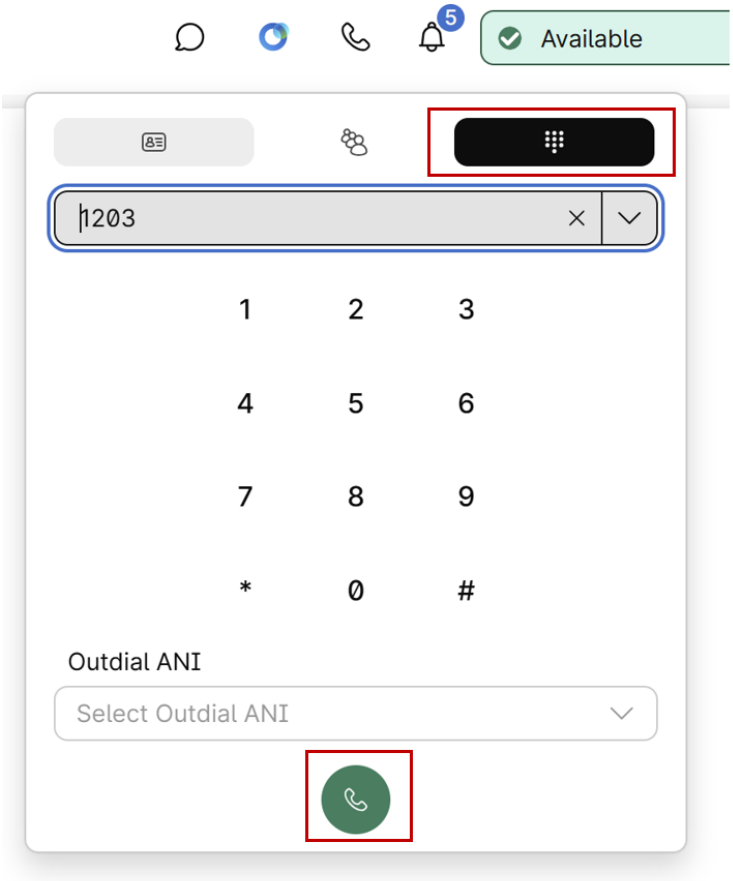


- This prevents us from dialing an individual cell phone number directly.
- To fix this, we need to find where the dial pad setting is controlled.
- Since this is an agent desktop function, we'll check the agent's desktop profile and the dial plan where we enabled the outdial option.

- In Control Hub, go back to the Desktop Profile section.
- Select the profile that's mapped to the agent you are working with.
- Navigate to the Dial Plans tab.
- Enable the dial plan functionality and select US as the dial plan. Then, click Save.



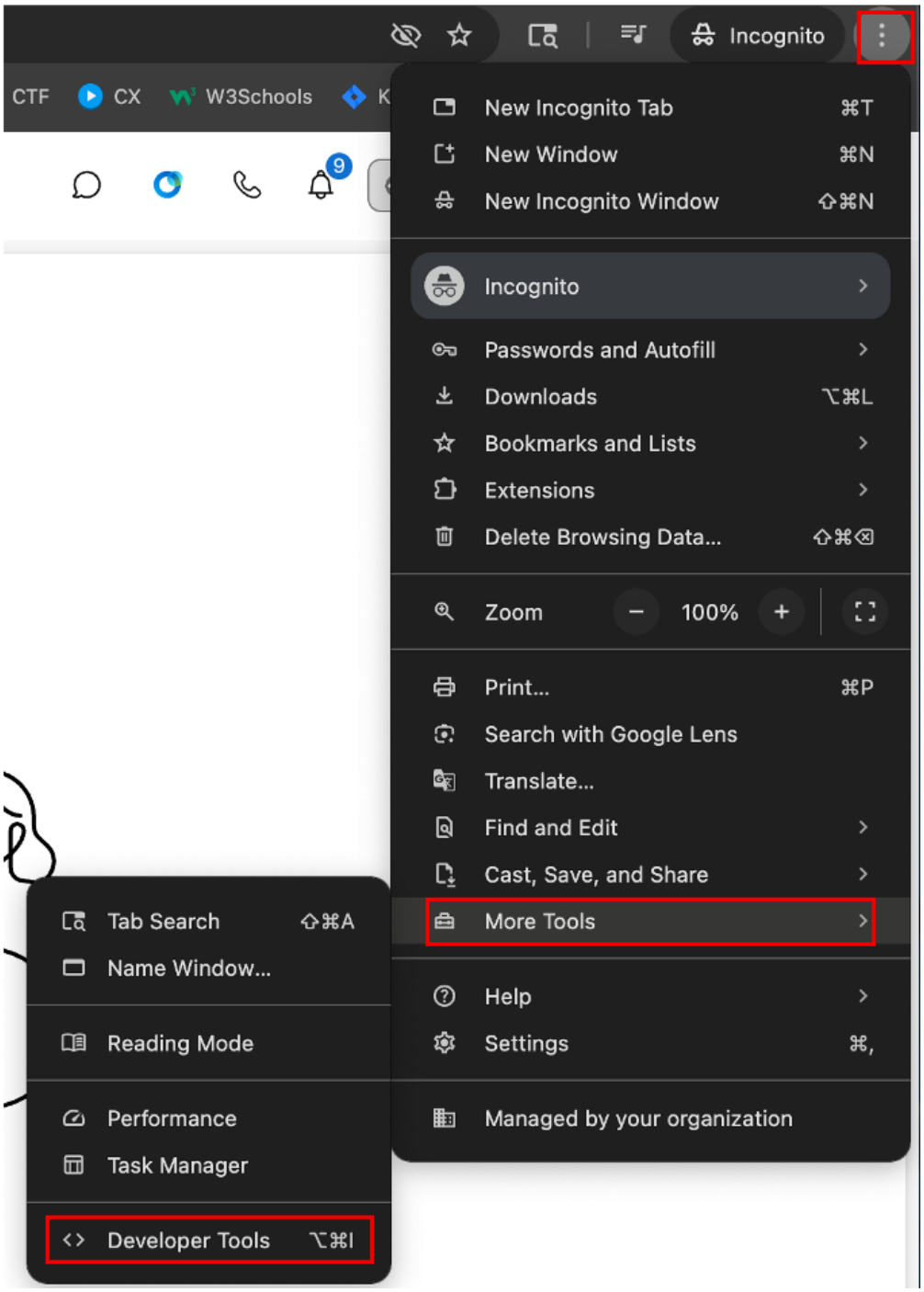
- Refresh the Agent Desktop application browser.
- Click the Outdial option again.
- You should now see the number pad pop up, allowing you to punch in numbers.
- Enter your cell phone number. You may add a "1" before the number, or it will work without it
- Click the Dial button.



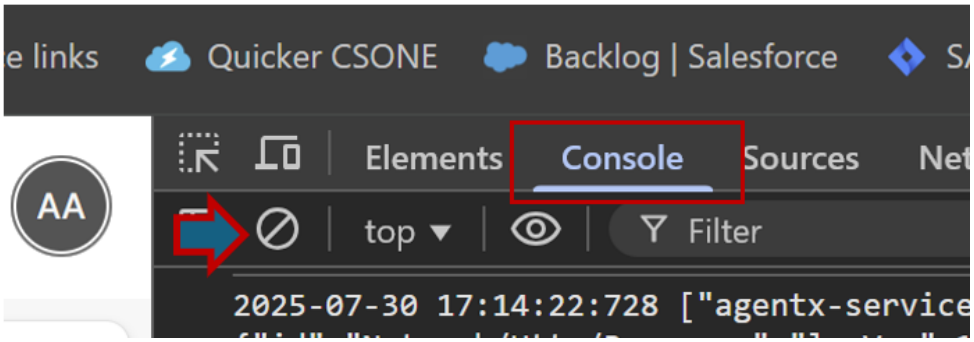
- You should ideally see an agent-initiated call to the cell phone number, but nothing happens.

1.4.4 Section 3 : Troubleshoot Outdial Failure

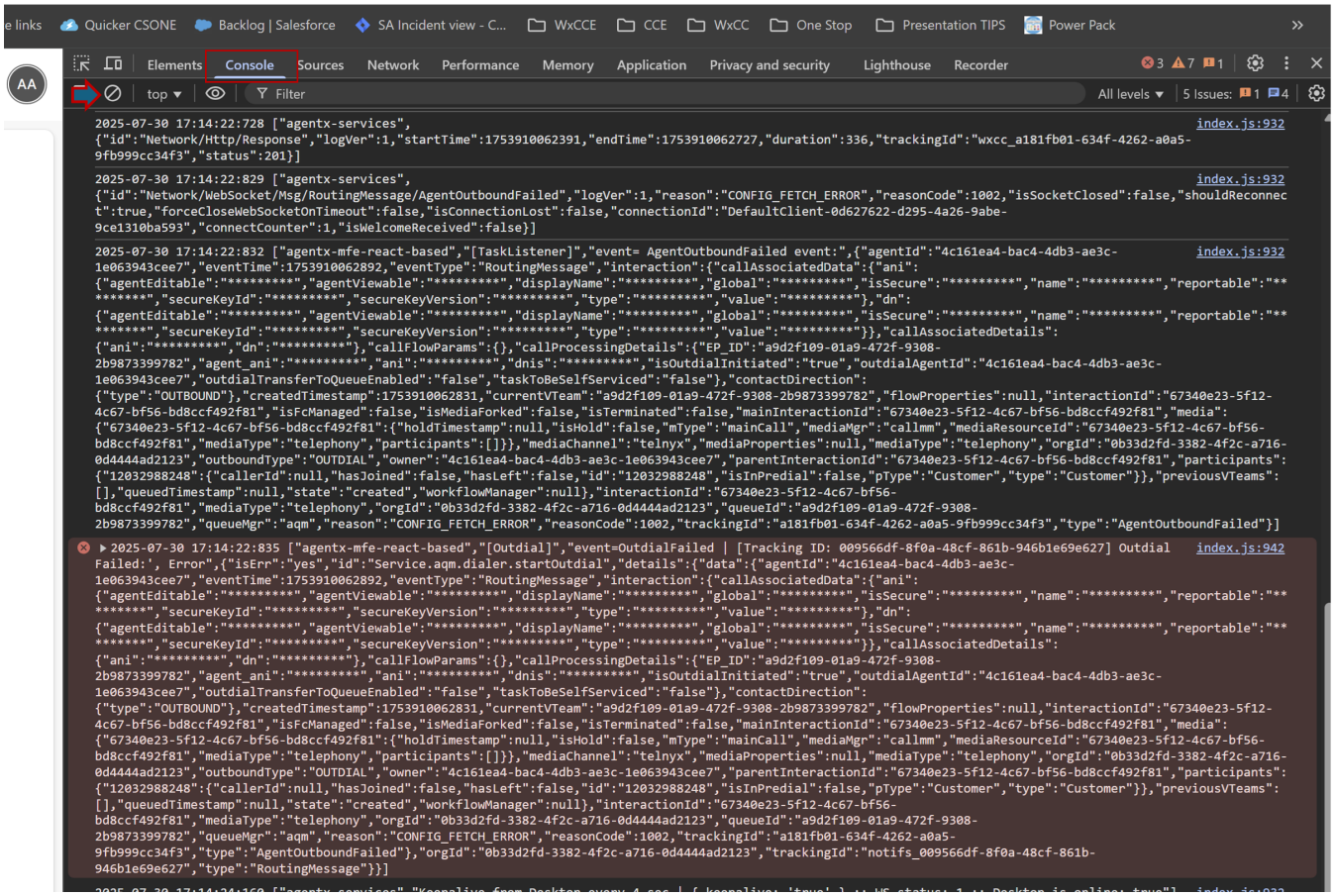
- Let's troubleshoot to see why this is the case.
- To figure this out, bring up the browser developer tool (Windows Shortcut: Press F12 Key)



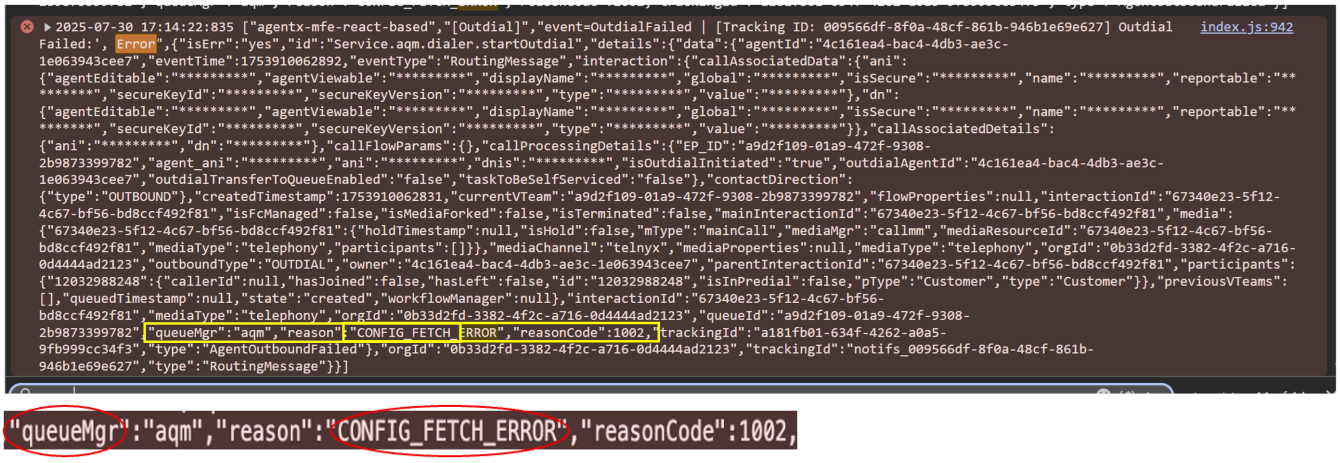
- Once the developer tool is up, ensure that it's on the "Console" tab and clear the console logs by selecting the "Clear Console" button



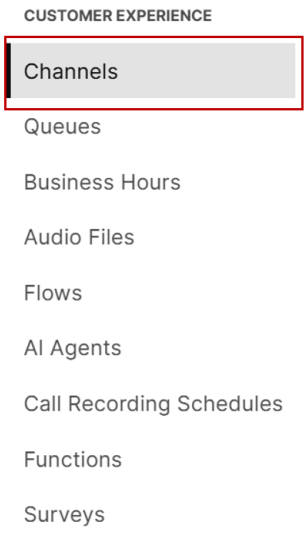
- Using the dial pad, dial the cell phone number again.
- As soon as the call fails, you should see a red error message in the console logs with the message **event=OutdialFailed**



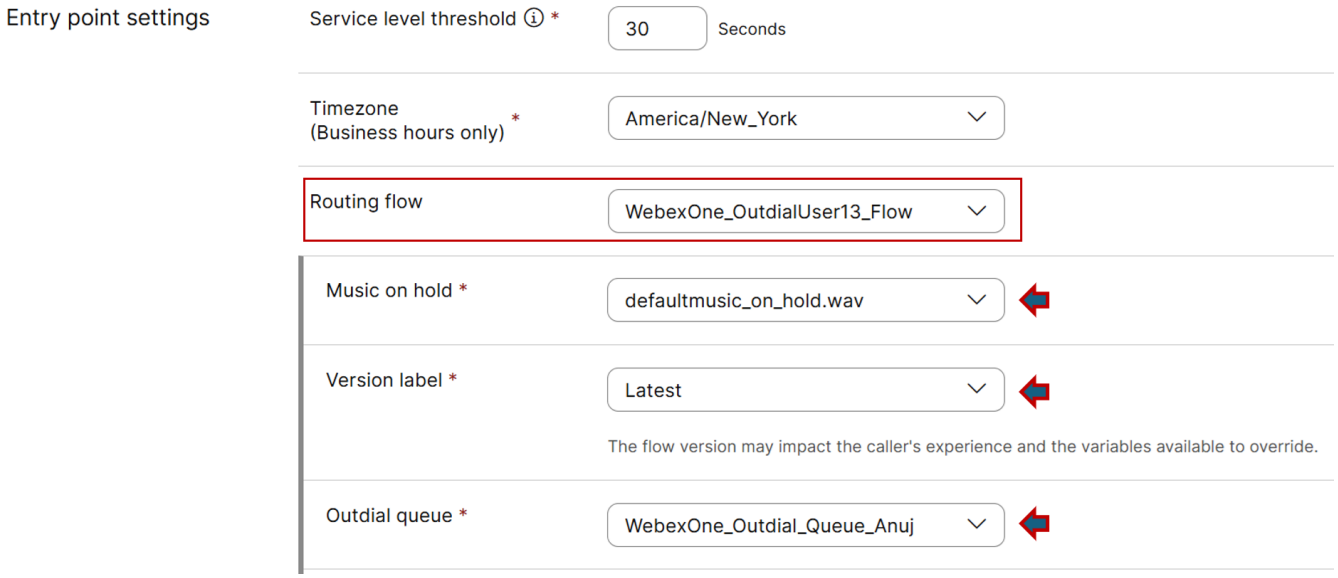
- Now, let's look closer into the failure message and figure out what the issue might be.
- For ease, one can copy the error message into a Notepad or Notepad++ application.
- Search for "error," and at the bottom of the error message, you will notice there is a fetch error on "**Config**" – "**Config_fetch_error**." The exact config it's talking about is "**queuemgr**" which basically means queue.



- This overall means the system is not able to fetch the team details from the queue perspective where agent resides.
- In WxCC, a queue is always mapped to an entry point via routing flows, so let's go back to the Entry Point for outdial and check the configuration again.
- Via Control Hub, under "Customer Experience," go back via "Channel" to the Outdial Entry Point that was configured.



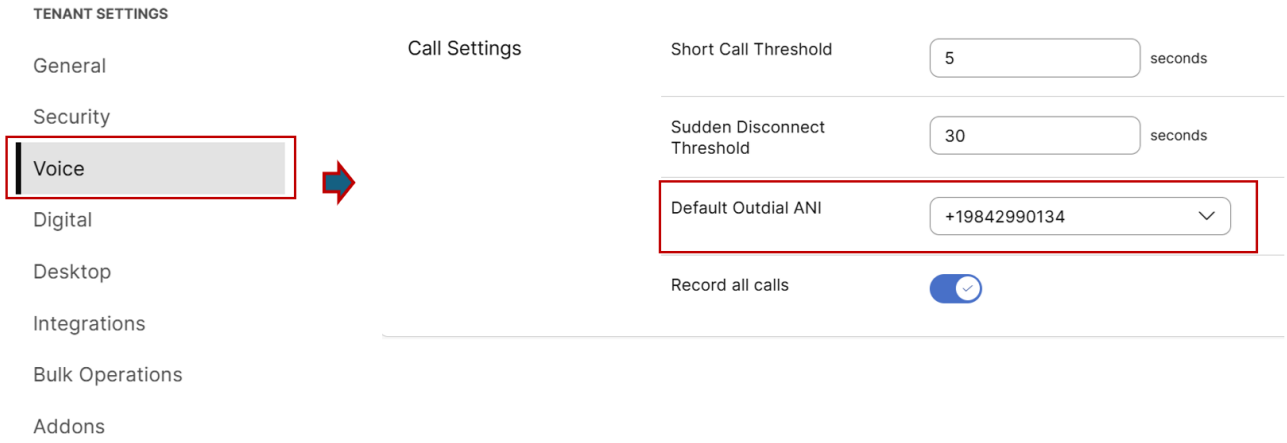
- Under "Entry Point Settings," you will notice that there is no routing flow mapped.
- From the dropdown, select the flow "**WebexOne_OutdialUser[num]_Flow**" and fill in these fields:
- Music on hold: "defaultmusic_on_hold"
- Version label: Latest
- Outdial Queue: Select the queue that was created in step 1 of your initial setup.



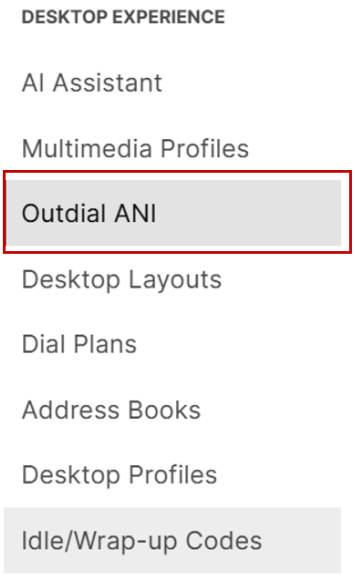
- Save the settings.
- Refresh the Agent Desktop browser and perform the outdial to the cell phone number; the call should now be successful.
- If you observe the browser debug console logs, you should see a message that will clearly show case its fetching the config and have the team details via the queue ID.

1.4.5 Section 4 : Custom Outdial ANI

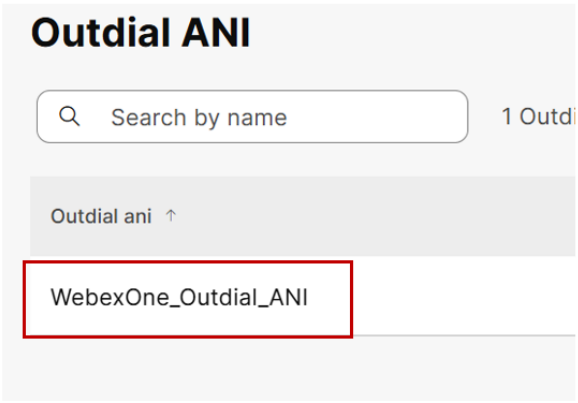
- In many cases, business requirements dictate that the Outdial ANI displayed on customer devices should be set to a specific toll-free or departmental number.
- Here, the outdial ANI noticed on the cell phone is "**+19842906065**" which is the default configuration set on the tenant level.
- To, review the Tenant-Level Outdial ANI setting in tenant Settings navigate to the Voice tab and note the existing Outdial ANI which is "**+19842906065**".



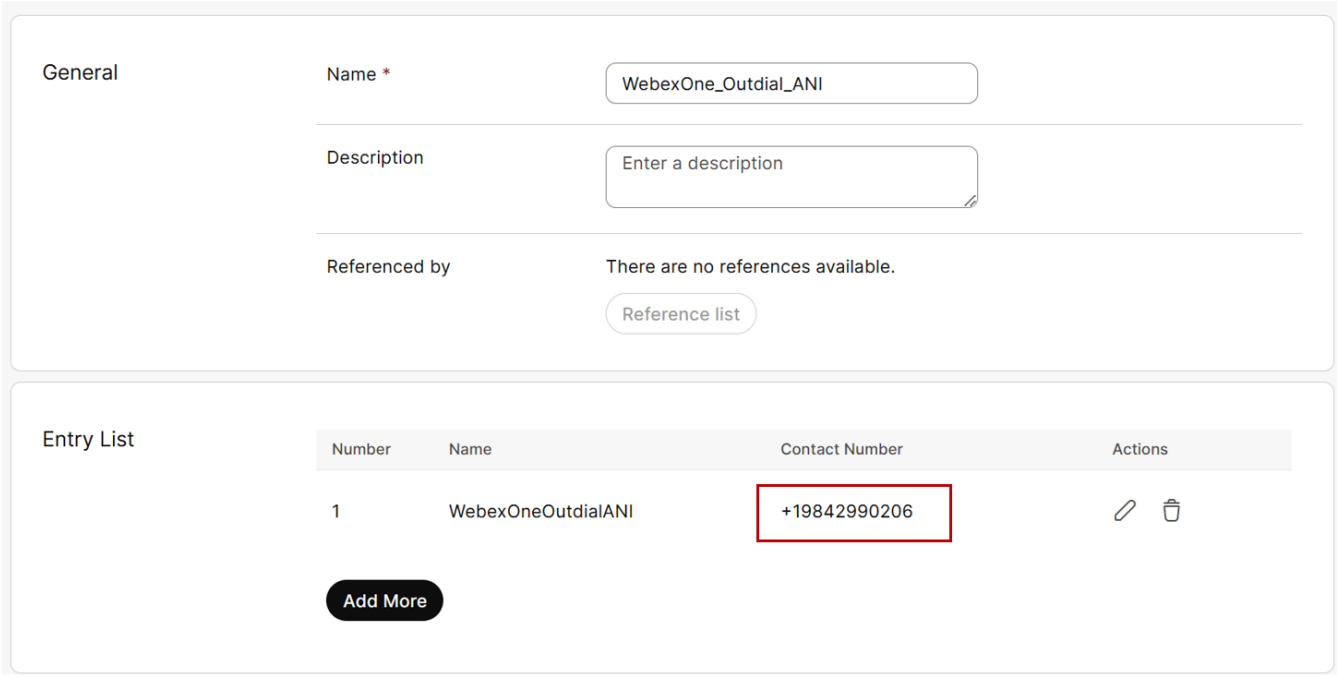
- To change to a custom ANI, administrator can create there own outdial ANI.
- For ease here outdial ANI has already been pre-configured and to review in Desktop Experience, go to Outdial ANI settings.



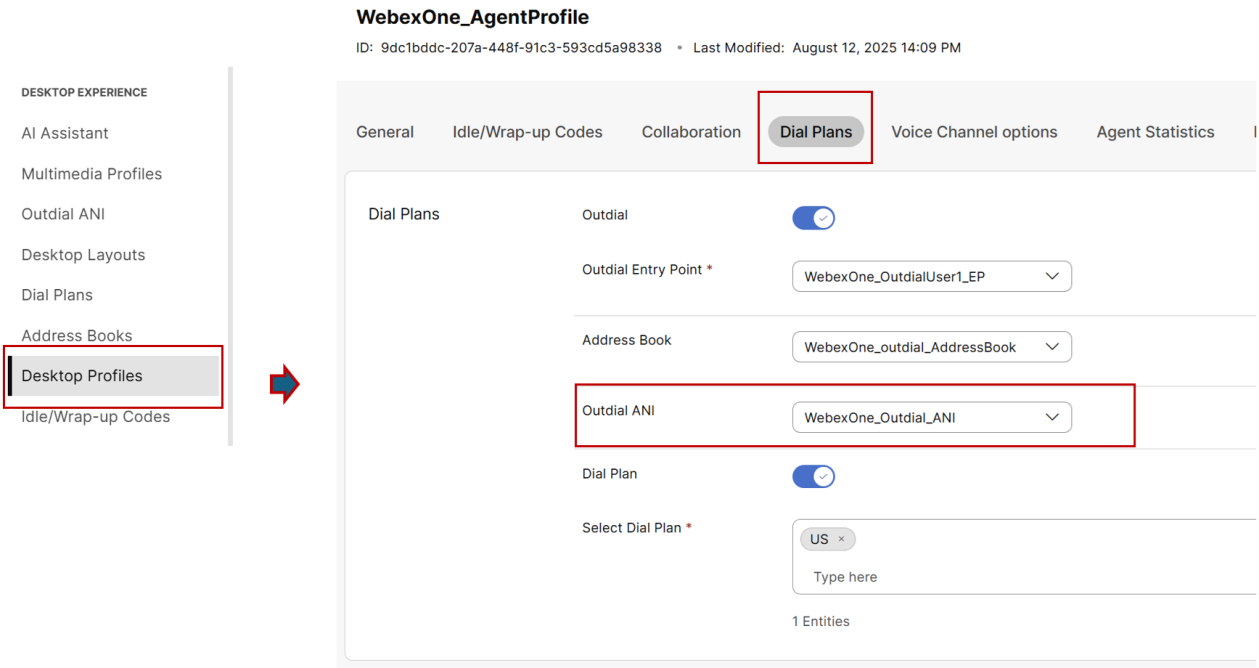
- Select **WebexOne_Outdial_ANI**.



- Confirm that it is mapped to the number "**+19842906070**".



- A custom Outdial ANI allows an agent to select the ANI on the desktop during an outdial call, provided the agent's desktop profile is mapped to the new Outdial ANI.
- To check in Desktop Experience, open the configured Agent Desktop Profile.
- Navigate to the Dial Plans section.
- In the Outdial ANI field, select **WebexOne_Outdial_ANI** and save your changes.



- Now, Perform an Outdial Call Using the Custom ANI.
- Log in or refresh the agent desktop.
- Initiate an outdial call.
- Verify on the recipient's device that the displayed ANI is the custom number.

Congratulations !! on completing this exercise!

You've not only set up the outdial feature from scratch but also learned how to identify and fix common errors, ensuring your deployments are both functional and reliable.

1.5 Task 4 - Exploring WebRTC data and statistics

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

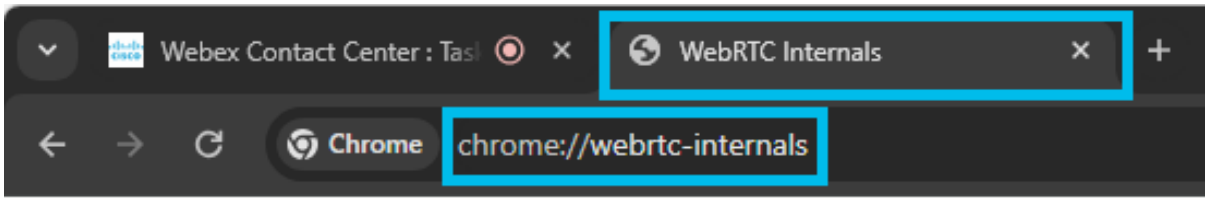
`Control Hub URL`	https://admin.webex.com
`Username`	labuser**ID**@wx1.wbx.ai *(where **ID** is your assigned pod number; this ID will be provided by your proctor)*
`Password`	webexONE1!

Info

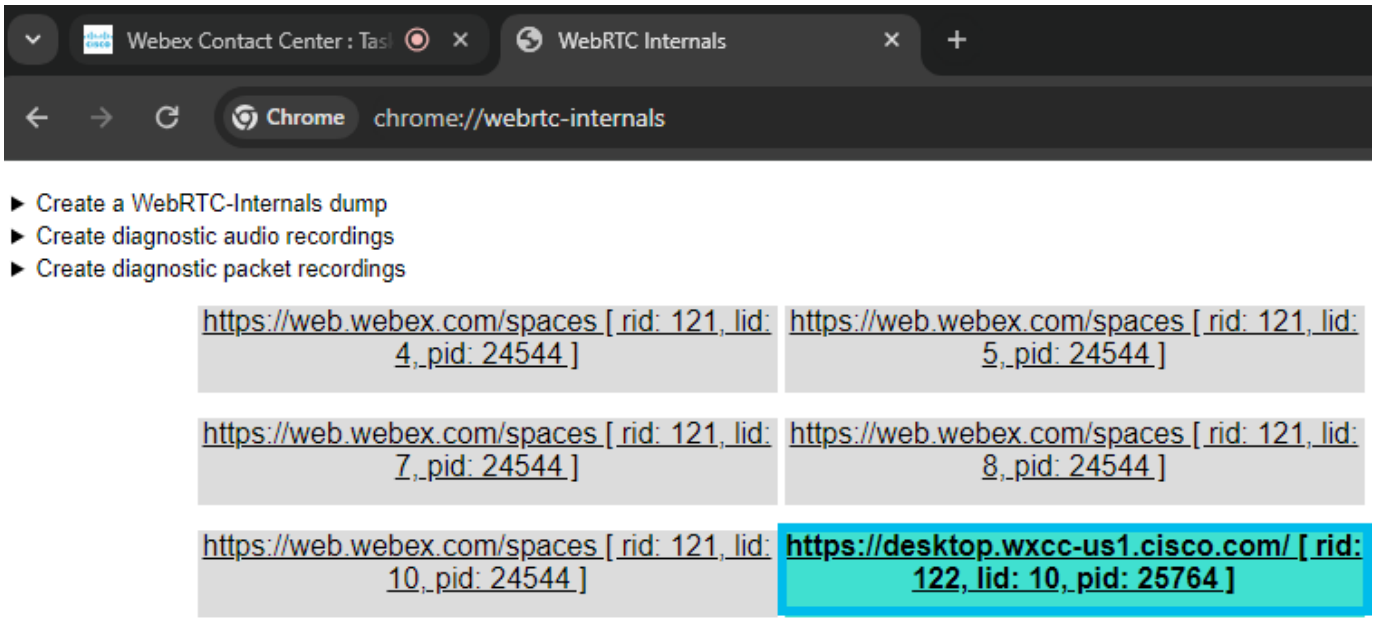
This task showcases available WebRTC call data and statistics using built-in browser tools and simulates missing audio during the call.

1.5.1 Section 1 - Exploring WebRTC internals

- Set your state to **'Available'** on the Agent Desktop.
- Make an inbound call and answer via Agent Desktop when the call is presented.
- Open a new tab in **Google Chrome** web browser.
- Navigate to **chrome://webrtc-internals**.



- In the **WebRTC-internals** tab select the active WebRTC session part (**desktop.wxcc-us1.cisco.com** in its name).

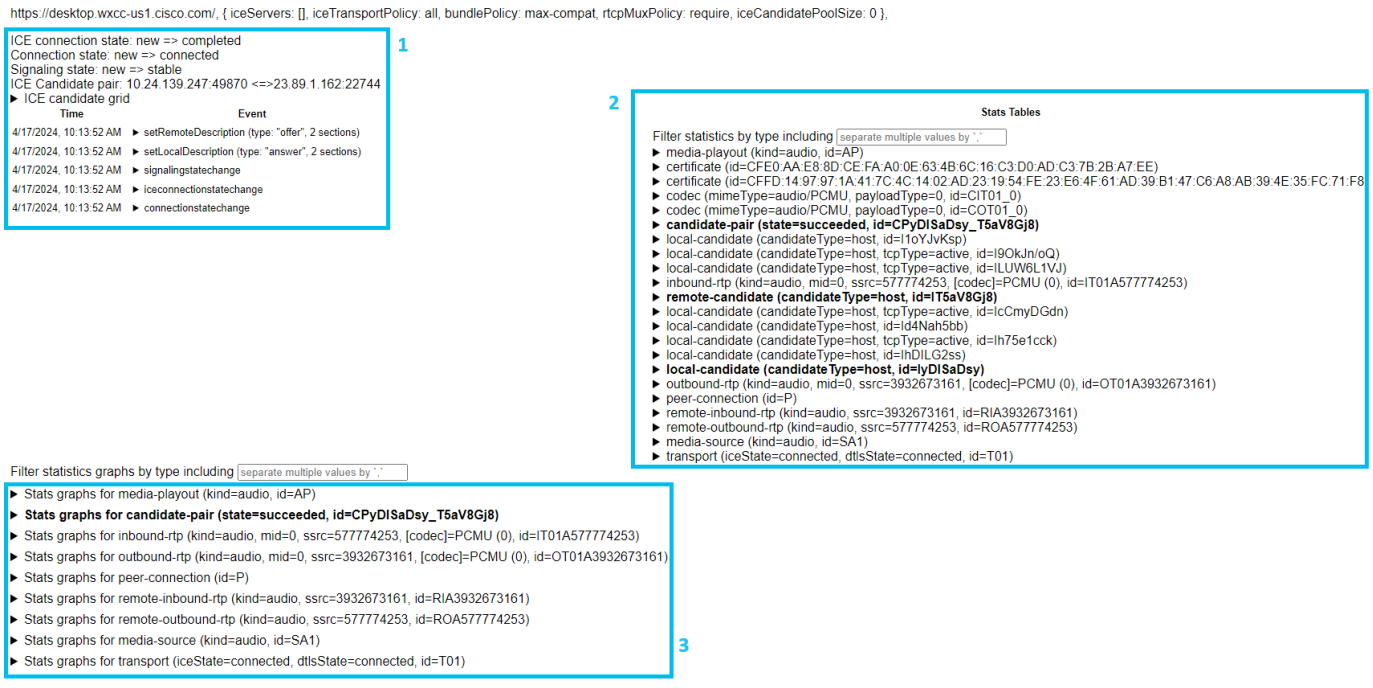


• The following information can be seen:

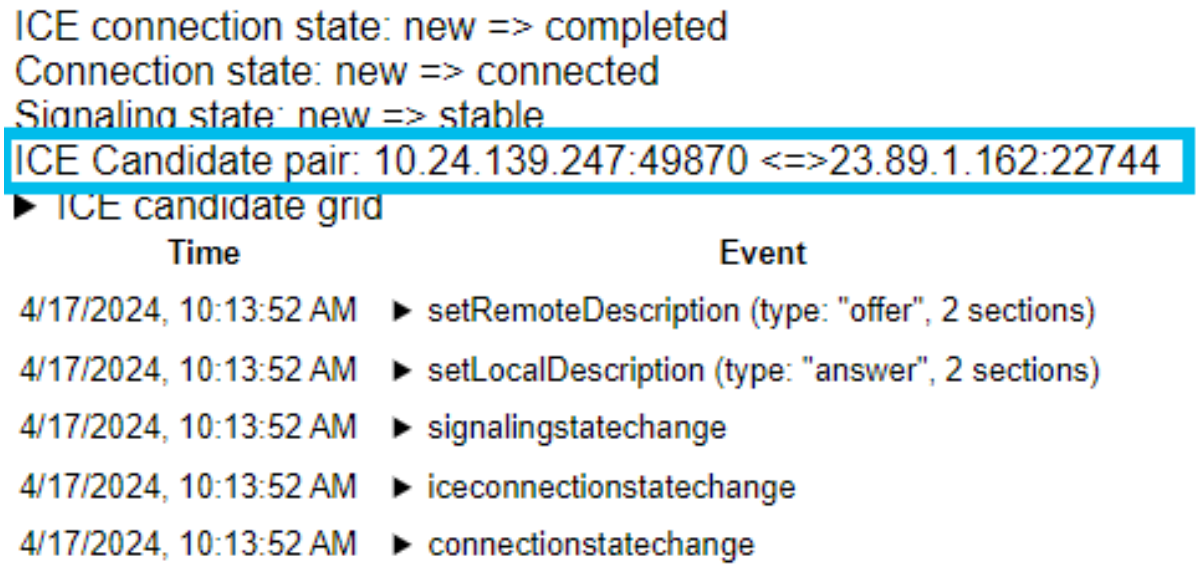
a. On the left side **(1)**, you'll find a record of the PeerConnection API activities. The RTCPeerConnection is the central interface in the WebRTC API. It represents the connection between the local and remote peer and provides all the functions and events necessary to establish the connection (more details can be found at https://webrtc.org/getting-started/overview). These traces **(1)** display interactions with the RTCPeerConnection object and parameters used.

b. On the right side **(2)**, you'll find the metrics collected via the getStats API about the current WebRTC connection.

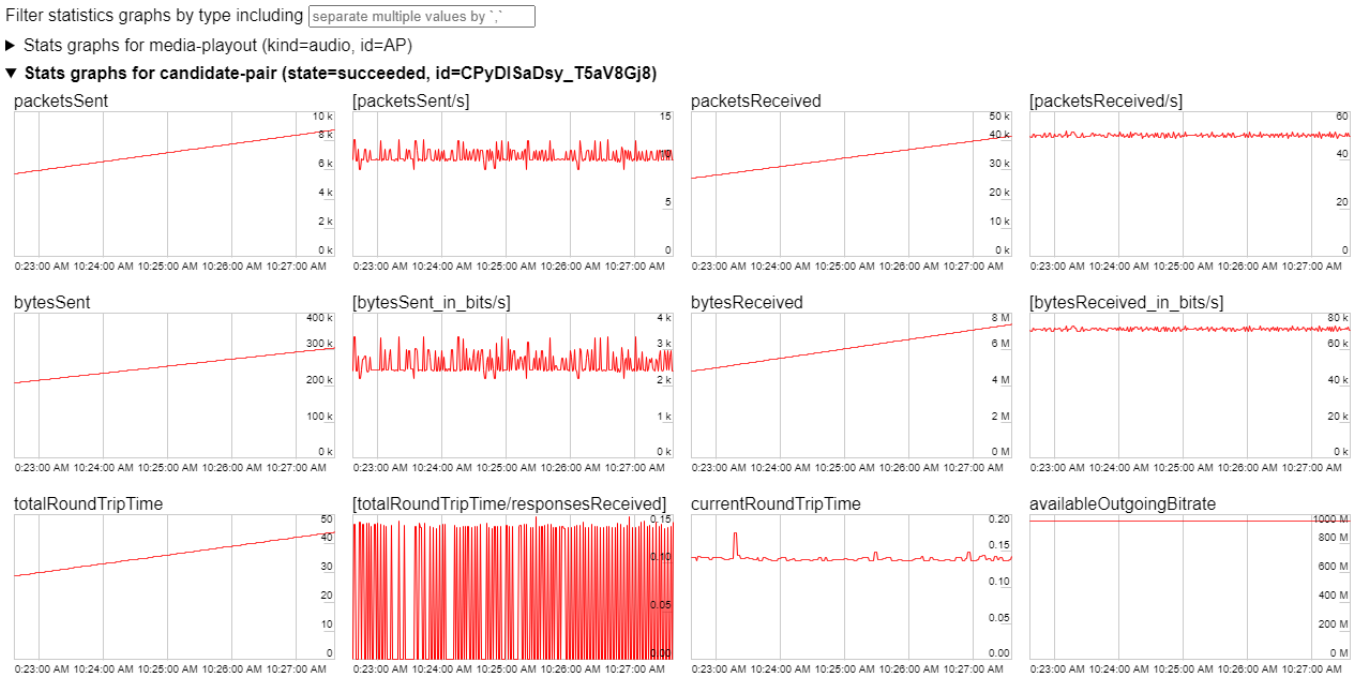
c. On the bottom **(3)**, you'll find graphs produced by utilizing the getStats API **(2)** about the current WebRTC connection.



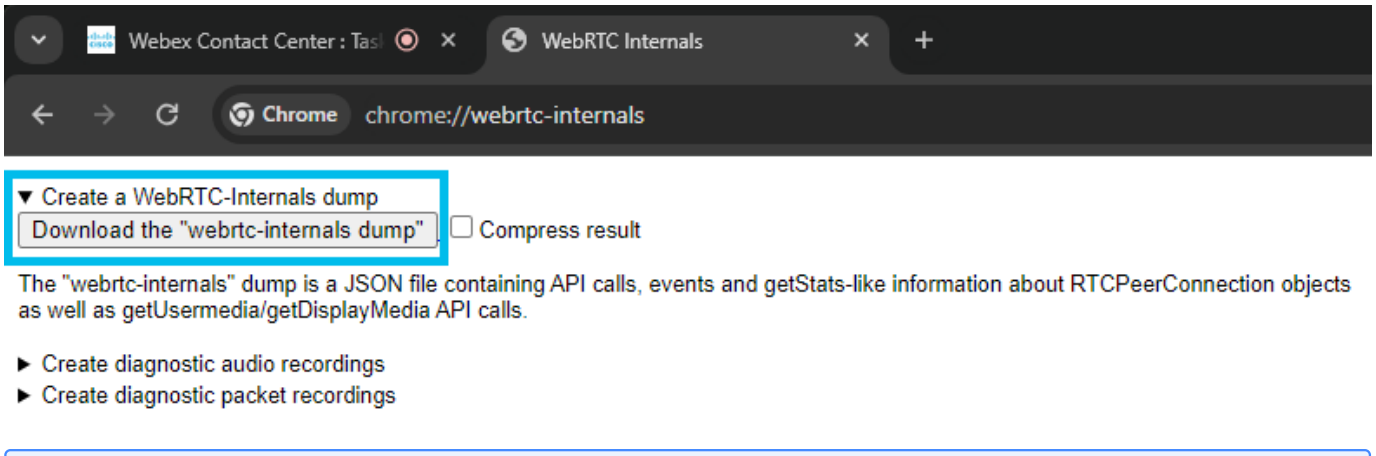
• Local and remote IPs and ports in use for this particular WebRTC connection can be seen under traces of the PeerConnection API calls.



- Various live call statistics (i.e. sent packets, received packets, latency, dropped packets etc.) related graphs can be seen at the bottom.



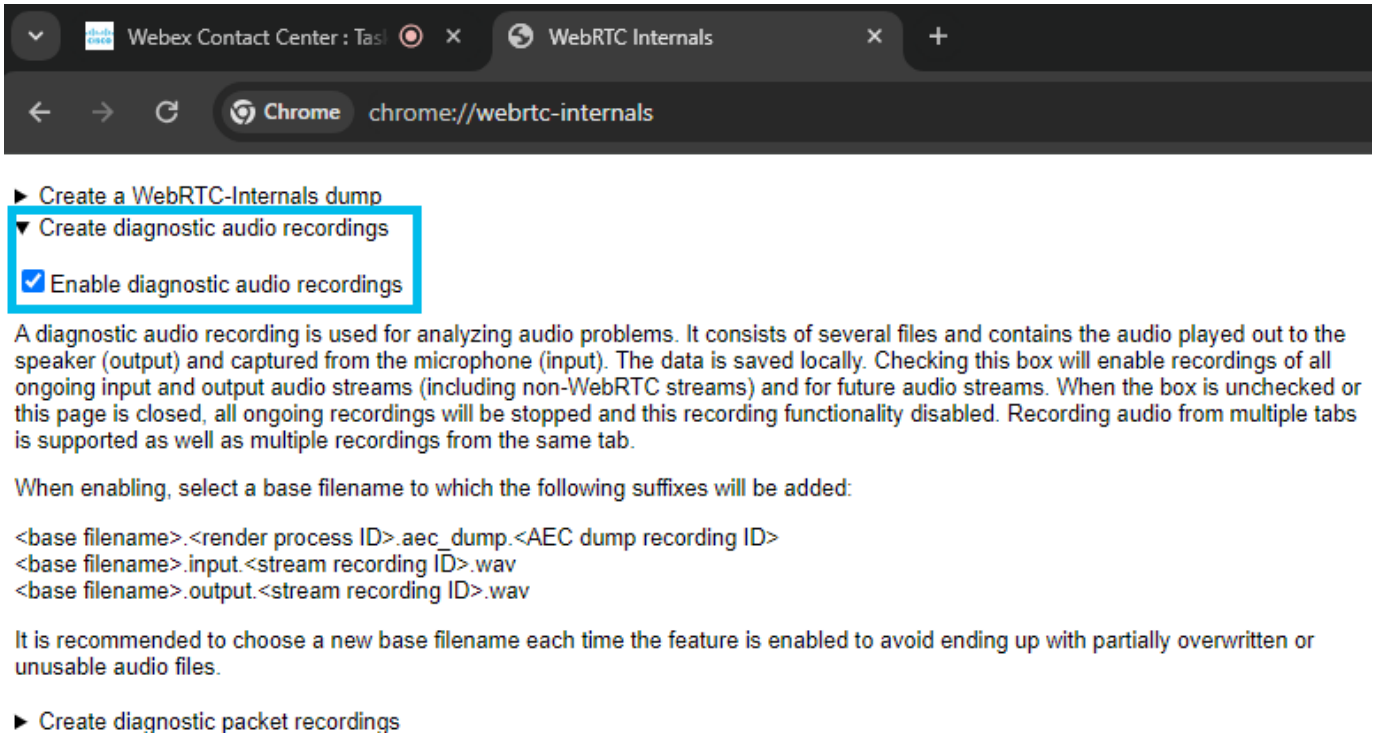
- WebRTC log dump can be retrieved by clicking **'Create a WebRTC-Internals dump'** and clicking **'Download the "webrtc-internals dump"** option.



Note

WebRTC log dump file can be parsed and analyzed using specific tools, i.e. https://fippo.github.io/webrtc-dump-importer

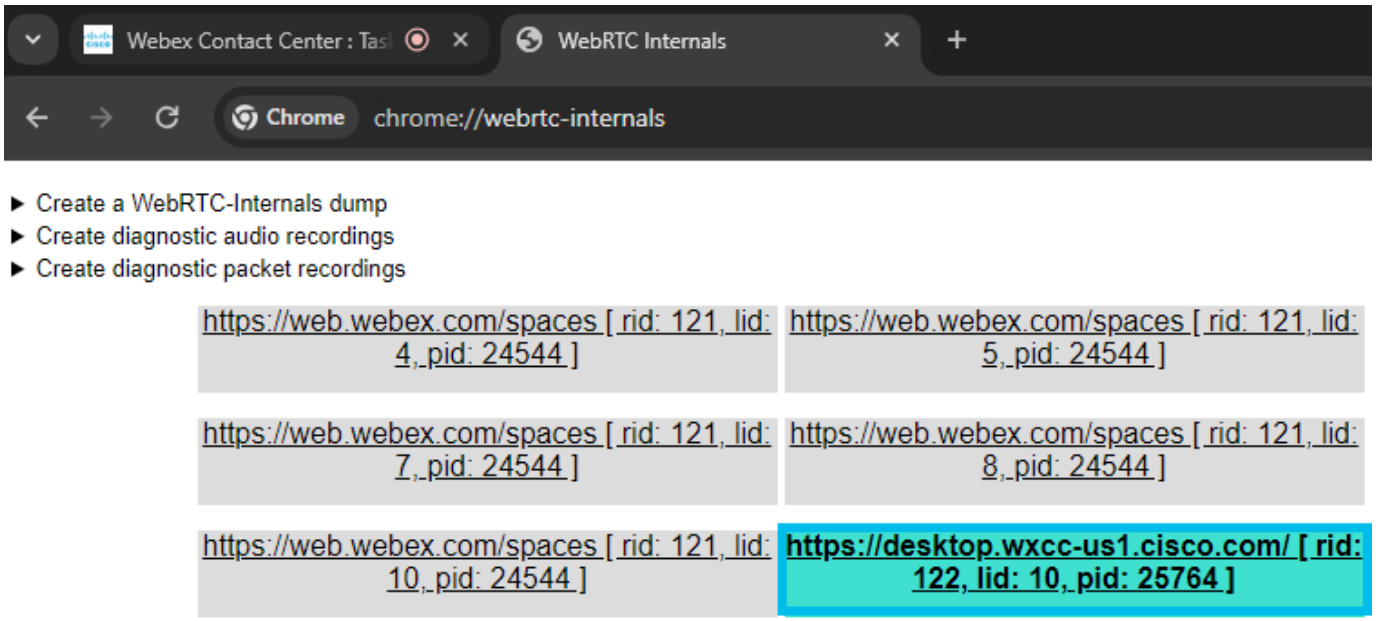
- WebRTC audio can also be captured for diagnostic purposes.



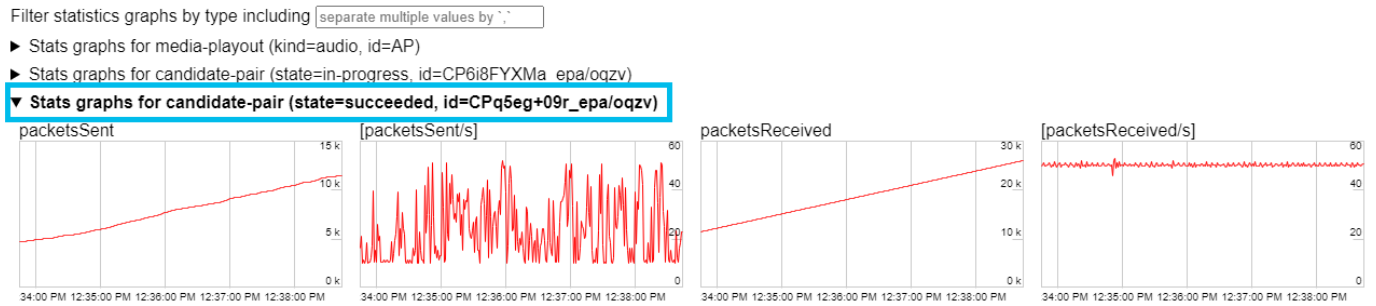
- End the call.
- Change your state to the idle **'Meeting'** state on the Agent Desktop.

1.5.2 Section 2 - Investigating missing audio

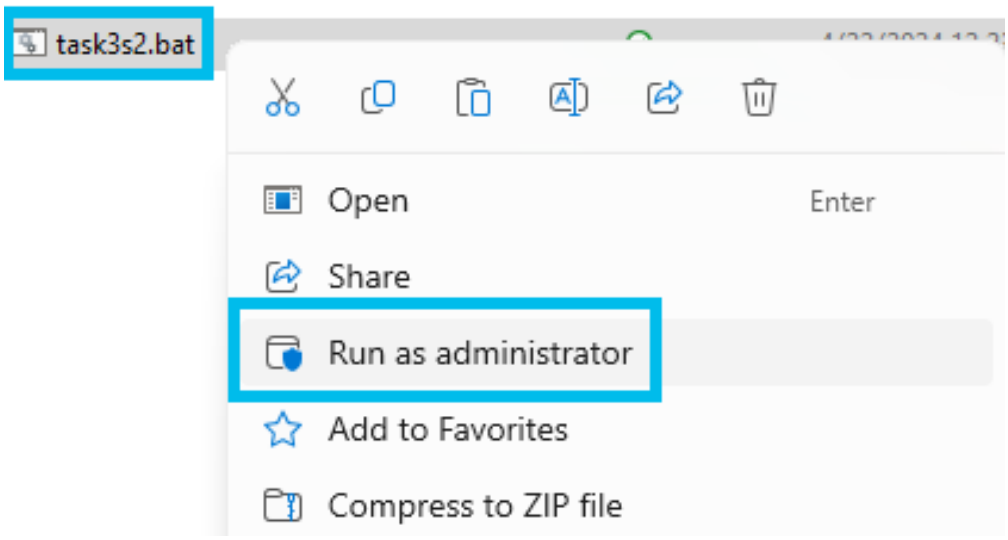
- Download **task3s2.zip** file from https://github.com/asuchank/clus24/raw/main/task3s2.zip and extract it.
- Make a new call to **+19842906065** using WebRTC via Agent Desktop and press **0** (music should be playing).
- Navigate to **chrome://webrtc-internals** in a new tab.
- In the **WebRTC-internals** tab select the active WebRTC session part (it will have **desktop.wxcc-us1.cisco.com** in its name).



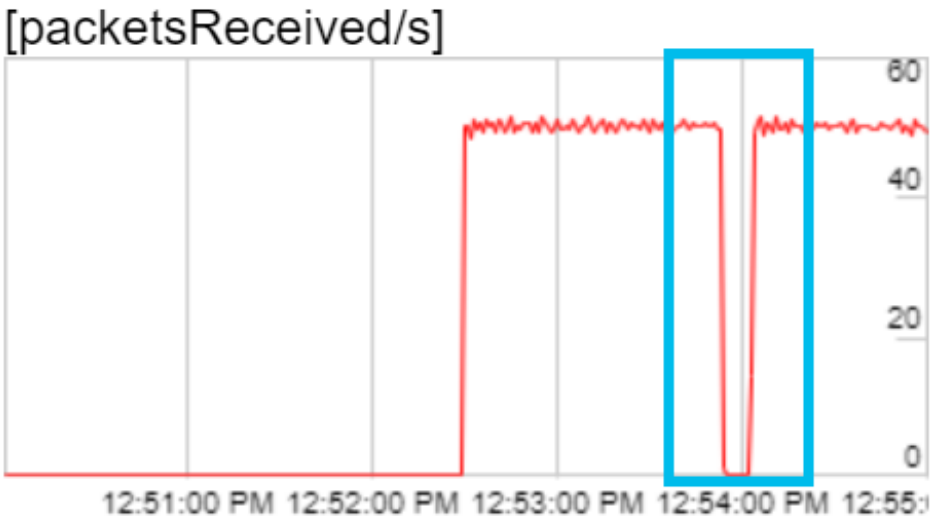
• At the bottom of the **WebRTC-internals** diagnostic page expand bolded text to uncover graphs for an ongoing session.



• Right-click on the **task3s2.bat** file and choose **'Run as administrator'** (audio will be lost for 10 seconds).



• Observe **packetsReceived/s** go down to zero on the expanded graph from before.



- Based on the observation above - missing audio was caused by the lack of media packets.
- End the call.
- Congratulations, you have completed this task!

1.6 Related Sessions at Webex One

- BRKXXX-1111 My friend's breakout session

2. How to use this guide

2.1 Setting up and using the tool

2.1.1 Initial Setup

This lab guide template uses MKDocs to take your markdown documentation and transform it into an interactive web based lab guide. You will need to install some software on your PC or Mac in order to take full advantage of the tool which will allow you to view your changes as you save them and ensure that your formatting is exactly how you want it.

Prerequisite Software to Install

1. Python
2. Visual Studio Code
3. Git

Setting up your lab guide on your computer

CLONE YOUR REPOSITORY WITH GIT

Open Visual Studio Code

Click the Source Control button in the left menu 

Click Clone Repository

Enter the repository source: {{config.extra.repo}}

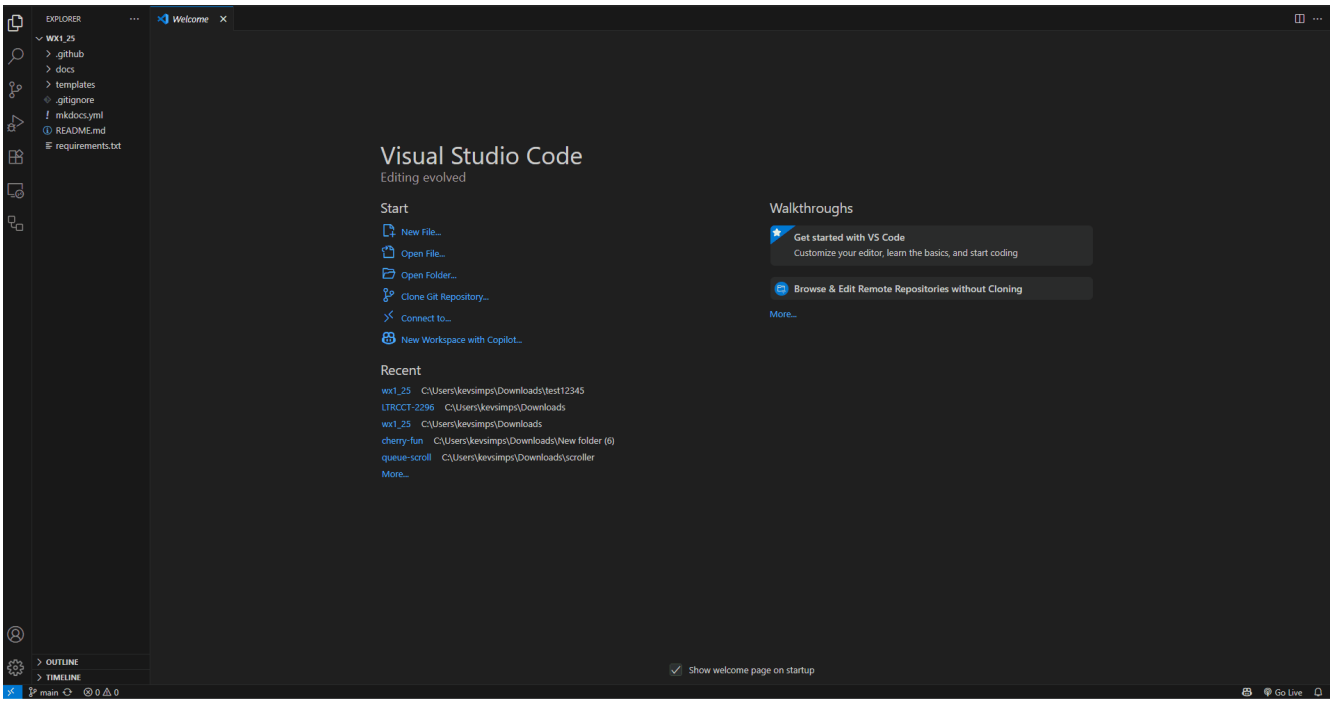
Select or create a new folder to clone the repository into.

BUILD THE ENVIRONMENT

When prompted to open the cloned repository, select open.

Drag open the terminal at the bottom of the Visual Studio Code window

Show Me



If you are on a PC If you are on a Mac

In your terminal enter the following commands:

python -m venv venv

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

venv\Scripts\activate.ps1

pip install -r requirements.txt

mkdocs serve

Open a browser to http://127.0.0.1:8000

In your terminal enter the following commands:

python -m venv venv

source venv/bin/activate

pip install -r requirements.txt

mkdocs serve

Open a browser to http://127.0.0.1:8000

2.1.2 Using the tool after the initial setup

After the initial setup you will only need to take the following steps:

Open Visual Studio Code

Click the Source Control button in the left menu 

Click Open Folder and navigate to the folder where you cloned the repository

If you are on a PC **If you are on a Mac**

In your terminal enter the following commands:

venv\Scripts\activate.ps1

mkdocs serve

Open a browser to http://127.0.0.1:8000

In your terminal enter the following commands:

source venv/bin/activate

mkdocs serve

Open a browser to http://127.0.0.1:8000

2.1.3 Publishing Your Changes

As you progress with the creation of your lab guide, you should sync your changes periodically to GitHub.

Syncing your changes to GitHub

Make sure that you have saved all of the files you want to sync up to the repository

Click the Source Control button in the left menu 

Click the plus sign next to each file you want to sync as you hover over the file names or alternatively, click the plus sign next to changes

Enter a note about the changes you are making in the message text box

Click the down chevron on the commit button to reveal more options

Select Commit & Sync

AFTER YOU SYNC YOUR CHANGES, IT WILL TAKE A FEW MINUTES FOR GITHUB ACTIONS TO BUILD THE WEBSITE SO THAT YOU CAN VIEW ALL OF THE CHANGES YOU MADE.

2.2 Styling and Advanced Features

2.2.1 Markdown and Other References

MK DOCS FEATURE REFERENCE:

https://squidfunk.github.io/mkdocs-material/reference/

MARKDOWN CHEAT SHEETS:

https://www.markdownguide.org/cheat-sheet/

https://github.com/lifeparticle/Markdown-Cheatsheet

2.2.2 Custom Features

Adding a Copy Button Without a Code Block

```
This will copy the following text: <copy>Text to copy</copy>
```

This will copy the following text: Text to copy

Adding user specific variables into your lab guide

There may be times in which you want to embed some attendee specific information into the instructions of your lab guide, like credentials or phone numbers, which will be used on multiple pages of your lab guide. You can gather the information via a form or you can pass a pre-encoded JSON string as a URL parameter to populate the variable values into the browser's session storage. Then you can use a simple HTML tag with a special class name to update the values in the lab guide. This feature can be combined with the copy button method above.

FORM METHOD

ow me the code

```
<form id="info">


<label for="Admin">Admin Login:</label>
<input type="text" id="Admin" name="Admin"><br>

<label for="PW">Admin Password:</label>
<input type="text" id="PW" name="PW"><br>

<label for="EP">Inbound Channel Name:</label>
<input type="text" id="EP" name="EP"><br>

<label for="DN">Inbound Channel Phone Number:</label>
<input type="text" id="DN" name="DN"><br>

<label for="Queue">Queue 1 Name:</label>
<input type="text" id="Queue" name="Queue"><br>

<label for="Queue2">Queue 2 Name:</label>
<input type="text" id="Queue2" name="Queue2"><br>

<label for="Team">Team 1 Name:</label>
<input type="text" id="Team" name="Team"><br>

<label for="Team2">Team 2 Name:</label>
<input type="text" id="Team2" name="Team2"><br>
<br>
<button onclick="setValues()">Update Lab Guide</button>
</form>
```

Example Input Form
Admin Login:

Admin Password:

Inbound Channel Name:

Inbound Channel Phone Number:

Queue 1 Name:

Queue 2 Name:

Team 1 Name:

Team 2 Name:

Update Lab Guide

Login: Provided by proctor

Password: Provided by proctor

Assigned Inbound Channel Name: Provided by proctor

Assigned Inbound Channel Number: Provided by proctor

Assigned Queue Name 1: Provided by proctor

Assigned Queue Name 2: Provided by proctor

Assigned Team name 1: Provided by proctor

Assigned Team name 2: Provided by proctor

URL METHOD

If you have a lot of attendee variables in your lab, you may choose to precompile and encode them so that you can simply provide a URL link which will load all of their required information.

To see this in action, add this string at the end of the URL for any page on this site: ?
eyJBZG1pbiI6ImFkbWluQHh5ei5iaXoiLCJQVyI6InNVcGVyU2VjcmV0MTIzISIsIkVQIjoiRVAxIiwiRE4iOiIrMTkxMDU1NTEyMTUyIiwiUXVldWUiOiJRdWV1ZTEiLCJRdWV1ZTIiOiJRdWV1ZTIiLCJUZWFtIjoiVGVhbTEiLCJUZWFtMiI6IlRlYW0yIn0=

2.2.3 Custom Admonitions

In addition to the stock admonitions which are available out of the box we have added some additional branded admonitions which were created by Bobby McGonigle.

Blank

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Blank!')
```

Vidcast

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Vidcast!')
```

Download

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Download!')
```

Cedeploy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Ce-Deploy!')
```

Webex

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Webex!')
```

Gif

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('How should I pronounce Gif?')
```

Important

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Important!')
```

Challenge

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Challenge!')
```

Tool

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Tool!')
```

Curious

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Curious!')
```