



## Amazon Connect PCI Pal Integration

23-February-2021

**Partner:** AWS

**Version:** 1.0

**Reference:** Amazon Connect configuration guide

**Sensitivity:** *Public*

## 1 Table of Contents

2	Purpose .....	2
3	Architecture overview .....	2
4	Integration Configuration .....	5
4.1	Amazon Web Services .....	5
i.	AWS Lambda .....	5
ii.	DynamoDB .....	5
4.2	AWS Amazon Connect.....	6
iii.	Generic configurations .....	6
iv.	Contact Flows .....	7
v.	Create the contact flows .....	7
vi.	DIDs .....	13
vii.	Quick Connects.....	14
viii.	Queues.....	15
ix.	Routing Profiles .....	15
x.	Agents (User Management) .....	16
5	Amazon Connect Agent Operation during PCI Pal Payments .....	16
5.1	How to take a secure payment .....	17

### Document History

Version	Modified by	Modified On	Checked On
1.0	Cesar Branco	23 <sup>rd</sup> February 2021	

### Controlled Distribution

Version	Sent by/Date	Sent to	Method
1.0			

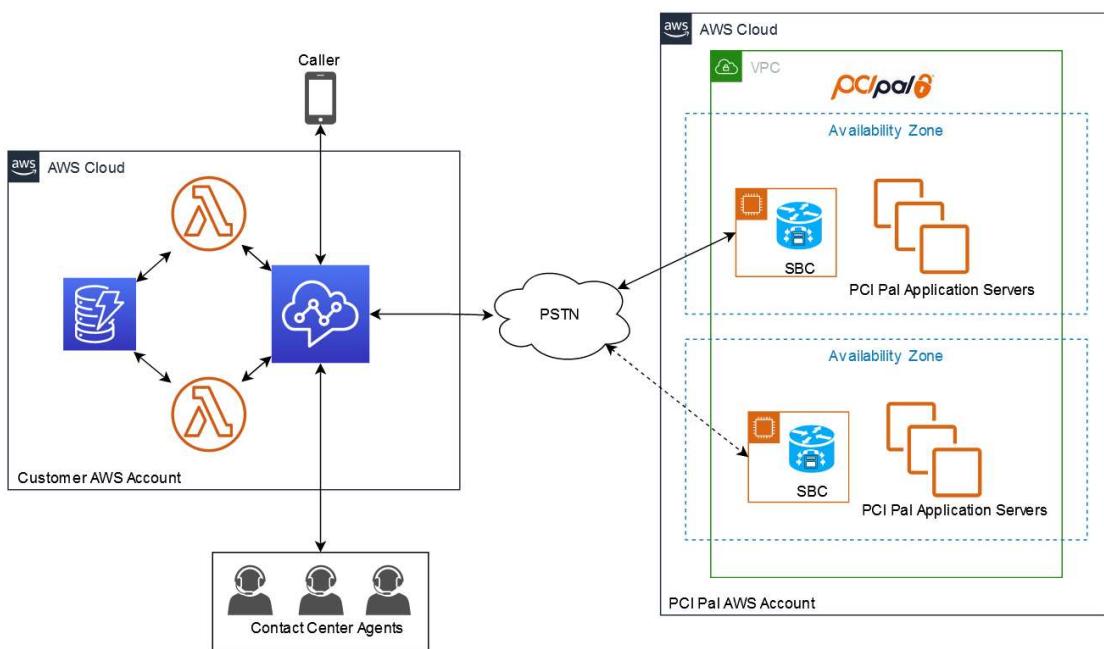
## 2 Purpose

This document describes the architecture and necessary configurations on the Amazon Connect tenant (the common PCI Pal and Amazon Connect customer) to integrate with the PCI Pal service and allow the agents to take secure (PCI) payments over Amazon Connect calls, maintaining constant communication with the cardholder (end customer on the phone), without putting that end customer on hold during payment or transfer to IVR.

PCI Pal will intercept and remove the credit card digits from the audio stream, in real time and without delays, during the agent and cardholder conversation, execute the payment on your (the customer) behalf, while informing the agent of what is happening throughout so she/he can guide the cardholder during the payment.

## 3 Architecture overview

The integration requires adding the Lambda functions and DynamoDB table to the merchant's AWS account, making the functions available for Amazon Connect and some other simple configurations, and creating the route(s) on PCI Pal's portal. Calls to PCI Pal are routed via PSTN connectivity. Following displays a high level architecture diagram.



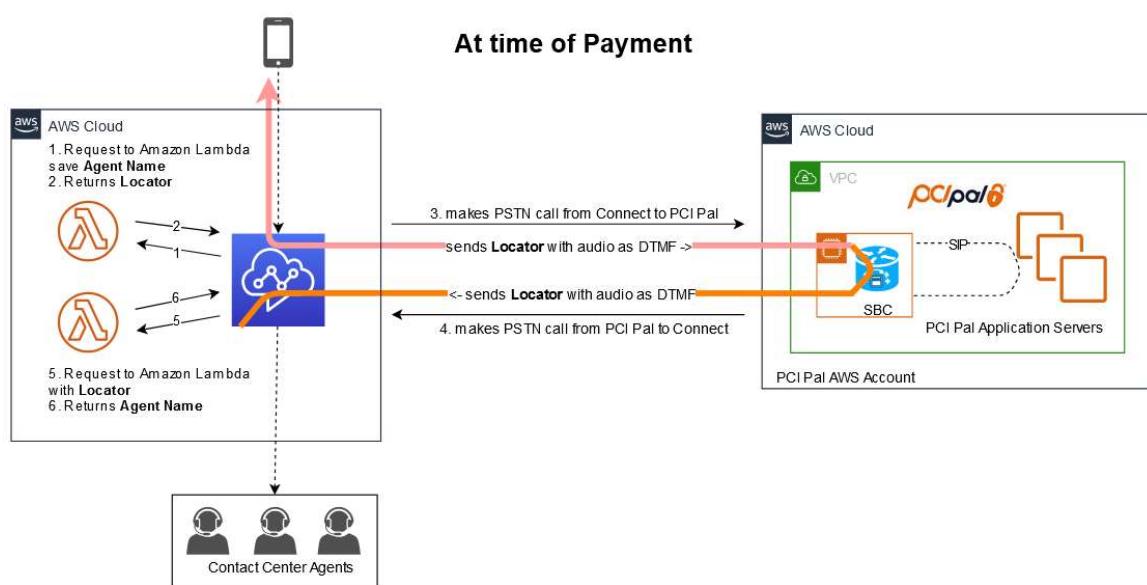
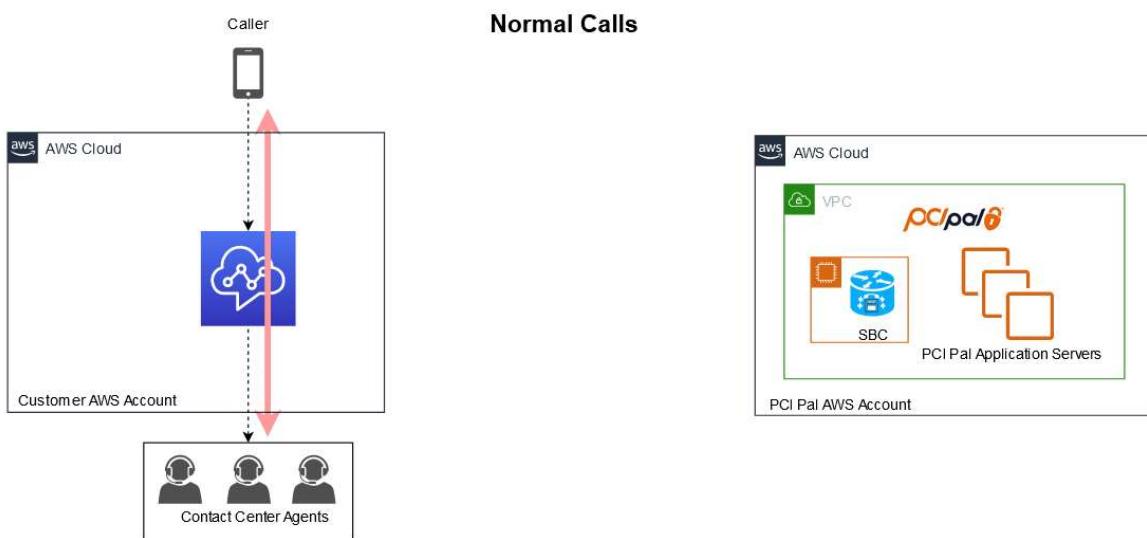
For this integration, it is relevant to be aware that during normal call operation PCI Pal does not have any visibility of the Amazon Connect calls. PCI Pal is only involved when a payment needs to be taken.

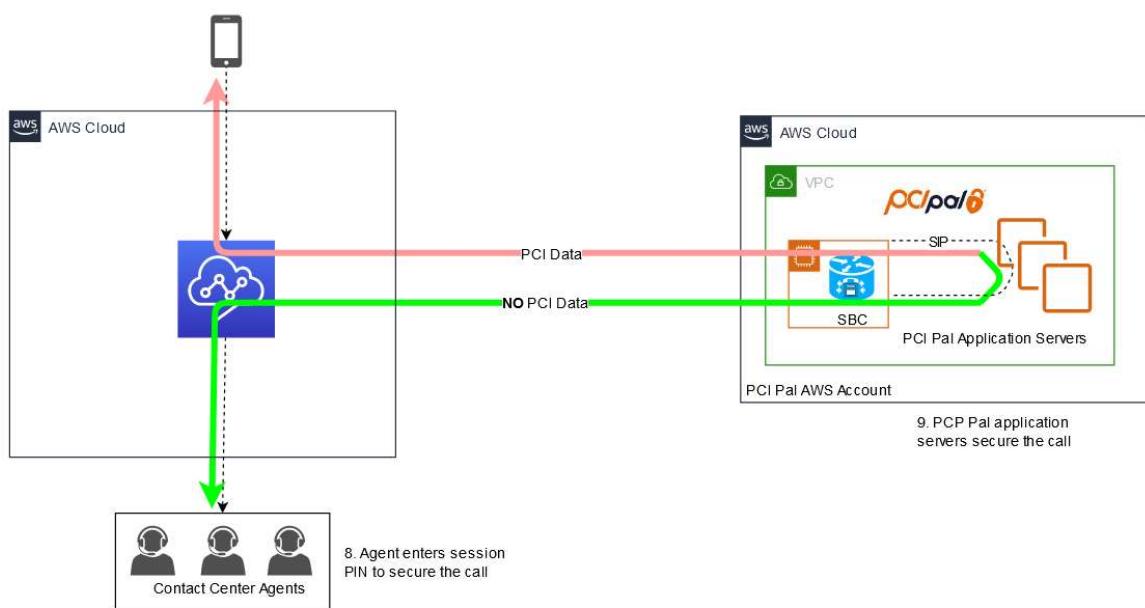
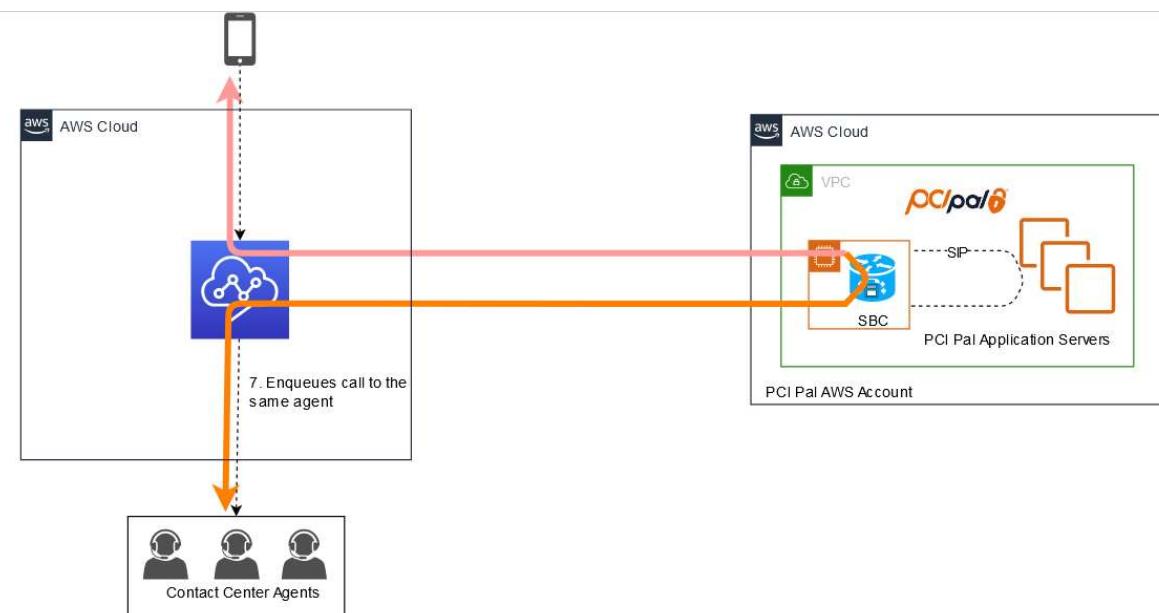
At time of payment, the agent will transfer the call to PCI Pal the “PCI Pal secure payment” “quick connect” contact on the agent’s soft phone.

1. Invoke an AWS Lambda function that will pass the agent name and receive a locator (4 digits).
2. Make a PSTN call to a PCI Pal DID associated with the customer (your tenant). PCI Pal will communicate this DID.
3. PCI Pal will make a PSTN call to an Amazon Connect DID associated with the customer payment service and an Amazon Connect **merchant** “contact flow”, creating a 2-call loop through PCI Pal.
4. When the call is answered by the **merchant** flow, the **cardholder** flow will send the locator PIN as DTMF tones.

5. The merchant “contact flow” will read the locator PIN and use that PIN to invoke an AWS Lambda function that will return the agent’s username.
  6. The merchant flow will use the agent name to enqueue the call to the same agent the cardholder was talking to.
  7. The agent disconnects the original call on its side and is connected to the new, secure, call - where it can continue speaking with the cardholder who was never disconnected.
  8. Now the agent and the cardholder can continue as an “agent assist” call, where the credit card details are cleaned from the merchant leg using the PCI Pal payment flows and gateway integrations as usual.

See sequence below:





## 4 Integration Configuration

The integration requires adding the Lambda functions and DynamoDB table to the merchant's AWS account (using the Quick Start), making the functions available for Amazon Connect and some other simple configurations, and creating the route(s) on PCI Pal's portal (done by PCI Pal and communicated to the customer during the deployment project).

### 4.1 Amazon Web Services

The required AWS services are deployed automatically by the AWS Amazon Connect Quick Start CloudFormation script provided by PCI Pal on the Quick Start page.

#### i. AWS Lambda

There are 2 lambda functions to be used:

AddAgent: saves the agent name before the 2-step transfer and provides a locator.  
input: the agent's Amazon Connect username.  
returns: a 4-digit locator (or error)

GetAgent: retrieves the agent name using the locator after the call loops through PCI Pal.  
input: a 4-digit locator  
returns: the agent's Amazon Connect username (or error)

#### ii. DynamoDB

There is 1 DynamoDB table that stores agents' names, locators and timestamps for the secure call request (this is not visible to the agent or Amazon Connect – described here just for information purposes):

	PIN	AgentName	CreatedTimestamp	
1	0027	cesar.branco	2020-12-17T21:27:23.284Z	
2	0014	francisco	2020-12-17T21:26:03.500Z	
3	0018	cesar.branco	2020-12-17T21:26:18.795Z	
4	0006	adolfo	2020-12-17T21:25:47.595Z	
5	0033	cesar.branco	2020-12-17T21:27:24.793Z	
6	0045	cesar.branco	2020-12-17T21:27:27.713Z	
7	0053	cesar.branco	2020-12-17T21:27:29.327Z	
8	0039	cesar.branco	2020-12-17T21:27:26.130Z	
9	0056	cesar.branco	2021-01-21T19:32:36.006Z	
10	0002	cesar.branco	2020-12-17T21:25:24.499Z	
11	0057	cesar.branco	2021-01-21T19:32:57.926Z	

These rows only live for a few seconds, the time to transfer the call through PCI Pal for secure payment and that call being answered by Amazon Connect before enqueueing to the same agent.

## 4.2 AWS Amazon Connect

### iii. Generic configurations

Open the Amazon Connect system configurations:

Amazon Connect virtual contact center instances

We are upgrading the Amazon Connect console  
To maintain your current level of access, make sure that you have required permissions. Learn more

Select a virtual contact center instance to manage its directory, administrator(s), telephony options, data storage, and advanced features.

Add an instance	Remove			
Instance Alias	Access URL	Channels	Create Date	Status
<input type="checkbox"/> pcipaldev	https://pcipaldev.awsapps.com/conne...	Inbound, outbound telephony	1/3/2020	Active

And click **telephony** to enable inbound and outbound calls:

Amazon Connect > pcipaldev

Overview

**Telephony** Data storage

Data streaming

Analytics tools

Tasks

Customer profiles

Approved origins

Contact flows

We are upgrading the Amazon Connect console  
To maintain your current level of access, make sure that you have required permissions. Learn more

Telephony Options

Amazon Connect offers the ability to accept inbound calls, make outbound calls, or both. You will claim a telephone number later.  
Note: You will not be able to place or receive phone calls if you don't select the corresponding telephony options.

Incoming calls

I want to handle **incoming** calls with Amazon Connect

Outbound calls

I want to make **outbound** calls with Amazon Connect  
Note: You can set which users can place outbound calls in user permissions.

Enable early media  
Agents can hear pre-connection audio such as busy signals, failure to connect errors, or other informational messages. When agents can't reach a contact, early media audio helps them understand why. Learn more.

Cancel Save

Then click contact flows and select two of the pre published Lambda functions, that will make them available while configuring the call centre logic:

```
<something>AddAgent<some GUID>
<something>GetAgent<some GUID>
```

and enable the logs (Contact Flow Logs) for debugging.

Contact flows

Add key Remove

**Amazon Lex**

Integrate Amazon Lex bots into your contact flows to take advantage of the same speech recognition and natural language understanding technology that powers Alexa.

Note: By adding Lex bots, you are granting Amazon Connect permission to interact with them [Create a new Lex bot](#)

Region EU: London Bot Select + Add Lex Bot

No Lex bots found for your account in the selected region. Create new.

**AWS Lambda**

Amazon Connect can interact with your own systems and take different paths in IVR dynamically. To achieve this, invoke AWS Lambda functions in contact flows to interact with your own systems or other services, then build personalized and dynamic experiences based on data returned.

Note: By adding Lambda functions, you are granting Amazon Connect permission to invoke them [Create a new Lambda function](#)

Function Select + Add Lambda Function

**Lambda Functions**

amplify-webapp-pcipal-161225-AddAgent-1UDR1GIOCB4RS	arn:aws:lambda:eu-west-2:854203033348:function:amplify-webapp-pcipal-161225-AddAgent-1UDR1GIOCB4RS	Remove
amplify-webapp-pcipal-161225-GetAgent-0BWVCI9D80V7	arn:aws:lambda:eu-west-2:854203033348:function:amplify-webapp-pcipal-161225-GetAgent-0BWVCI9D80V7	Remove

**Contact flow logs**

Enable Contact flow logs  
Your logs will be stored here /aws/connect/pcipaldev

#### iv. Contact Flows

Contact flows define the contact centre logic and behaviour for ACD, queues, IVR, transfer and hold. The PCI Pal integration requires only three flows as defined in this section.

<https://docs.aws.amazon.com/connect/latest/adminguide/create-contact-flow.html>

Type	When to use
Inbound contact flow	This is the generic contact flow type that is created when you choose the <b>Create contact flow</b> button, and don't select a type using the drop-down arrow. It creates an inbound contact flow. This contact flow works with voice, chat, and tasks.
Agent whisper flow	Use to manage what the agent experiences as part of an inbound call immediately before being joined with a customer. The agent and customer whispers are played to completion, then the two are joined. This contact flow works with voice, chat, and tasks.
Transfer to queue flow	Use to manage what the agent experiences when transferring to another queue. This type of flow is associated with transfer to queue quick connects, and often plays messaging, then completes the transfer using the <b>Transfer to queue</b> block. This contact flow works with voice, chat, and tasks.

#### v. Create the contact flows

##### 4.2.1.1 Save agent name - Agent Whisper Flow

This flow will store the username of the agent handling the call just before she/he answers it. We will use the default, however it can be changed from another contact flow, so in that case change the non-default flow.

Amazon Connect Contact flows

Name	Type	Description	Status
Default agent hold	Agent hold	Audio played for the agent when on hold	Published
<b>Default agent whisper</b>	Transfer to agent	Default flow to transfer to an agent.	Published
Default agent whisper	Agent whisper	Default whisper played to the agent.	Published
Default customer hold	Customer hold	Default audio the customer hears while on hold.	Published
Default customer queue	Customer queue	Default audio played when a customer is waiting in queue.	Published

Amazon Connect Contact flow designer

```

graph LR
    EP[Entry point] --> SCABlock[Set contact attributes  
Output: QueueAgentName ...]
    SCABlock -- Success --> PPBlock[Play prompt  
Text: $QueueName, you a...]
    PPBlock -- Okay --> EFBlock[End flow / Resume]
    
```

We'll store the agent username as a user defined Attribute, named **QueueAgentName** that we will refer to when the call is 2 step transferred to PCI Pal, using the "Set contact attributes" block:

Set contact attributes

Define and store key-value pairs as contact attributes. Learn more

Contact attributes are accessible by other areas of Amazon Connect, such as the Contact Control Panel (CCP) and Contact Trace Records (CTRs).

Attribute to save

Destination Type: User Defined

Destination Attribute: QueueAgentName

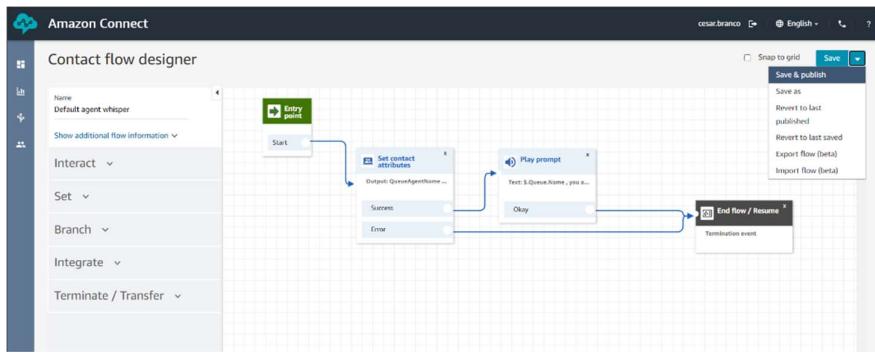
Use attribute

Type: Agent

Attribute: User name

Add another attribute

The play prompt contact block is not necessary, it is just for debugging purposes. Save and publish:



Exported flow (link to github):



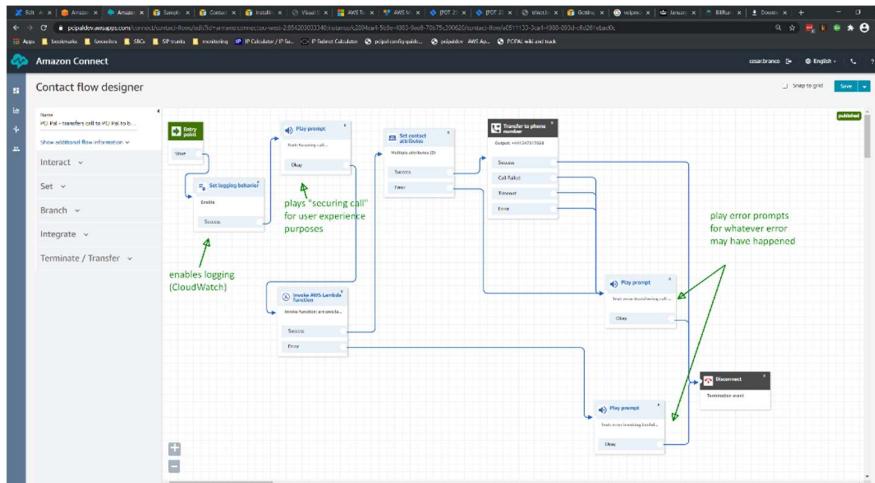
<https://docs.aws.amazon.com/connect/latest/adminguide/default-agent-whisper.html>  
<https://docs.aws.amazon.com/connect/latest/adminguide/set-whisper-flow.html>

#### 4.2.1.2 Transfer Call to PCI Pal - Transfer to Queue Flow

The Queue Transfer Flow will be associated with the “quick connect” the agent will use for the 2 step transfer that loops the call through PCI Pal, and will store the agent user name, retrieve a locator, **transfer** the call to a PSTN DDI (that resides at PCI Pal’s platform) **and** send the locator as a DTMF sequence to be retrieved by the Amazon Connect flow that answers the call as it loops.

Name	Description	Last modified
Default agent whisper	Agent whisper	Default whisper played to the agent.
Default customer hold	Customer hold	Default audio the customer hears while on hold.
Default customer queue	Customer queue	Default audio played when a customer is waiting in queue.
Default customer whisper	Customer whisper	Default whisper played to the customer.
Default outbound	Outbound whisper	Default flow for outbound calls.
Default queue transfer	Transfer to queue	Default flow used to transfer to a queue.
Original Default agent transfer	Transfer to agent	Default flow to transfer to an agent.
PCI Pal - transfers call to PCI Pal in a loop second after sending agent name	Agent whisper	Default whisper played to the agent.
Contact flow	Transfer to queue	Accept call from PCI Pal and deliver to the same agent by calling a lambda function.
PCI Pal - transfers call to PCI Pal ...	Transfer to queue	Loop Transfer to PCI Pal saving the agent name with a lambda functio...

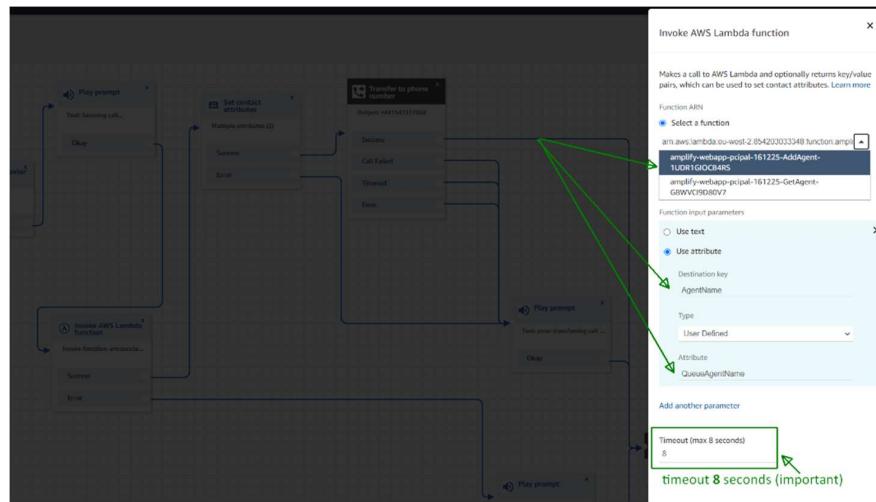
Full flow (name and description set on the upper left corner):



“Invoke Lambda Function”: select the **AddAgent** function, then pass the “**QueueAgentName**” attribute saved on the whisper flow as the Lambda function parameter “**AgentName**”.

#### NOTE:

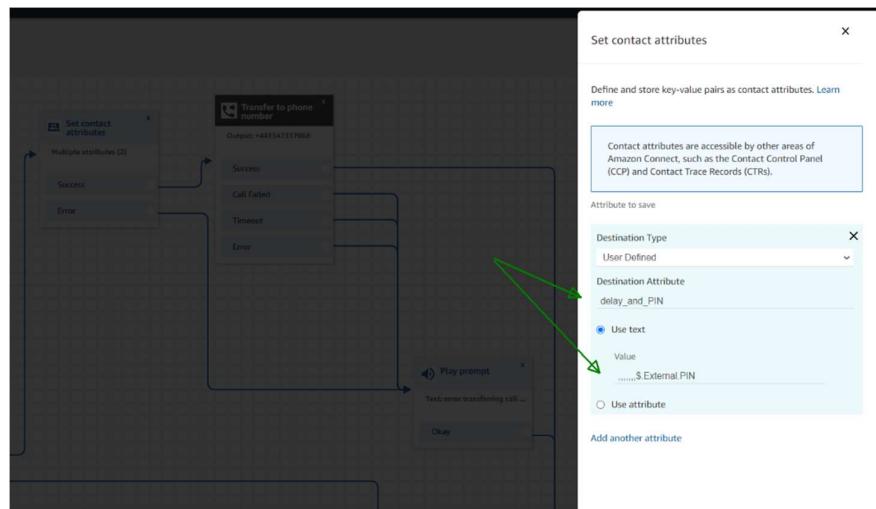
Set the timeout to **8** seconds, that is important because sometimes the function takes 3-4 seconds to “wake-up” if it has not been called in a while and so will timeout and raise an error, typically the response time is < 200 ms.



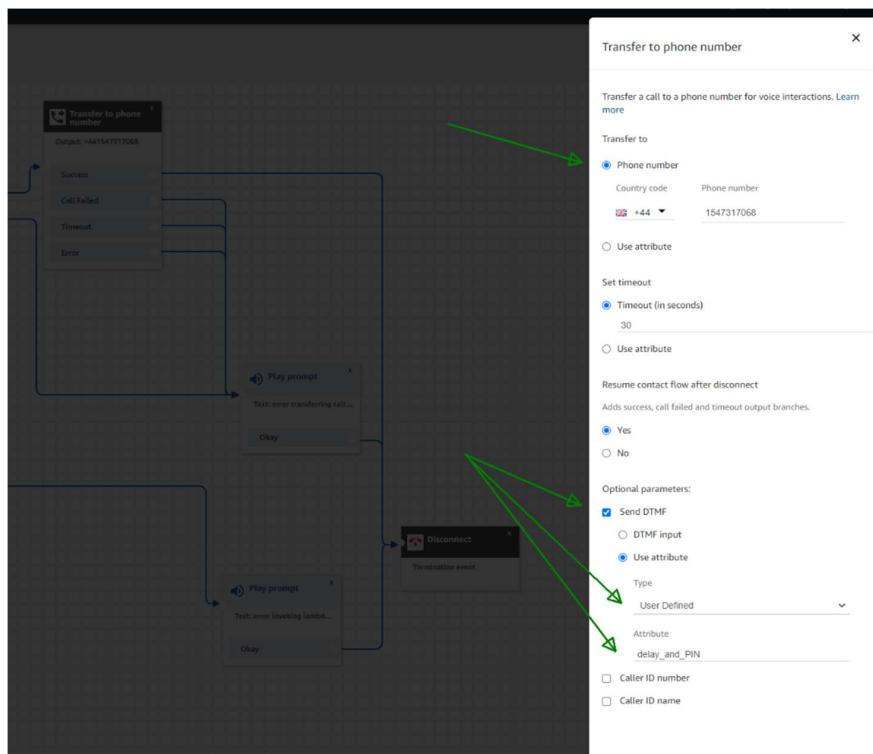
“Set contact attributes”: creates an **User Defined** attribute named “**delay\_and\_PIN**” to be used by the transfer node to send as DTMF to be read by the Contact Flow that answers the looped call. The value is:

**,,,,,,,\$.External.PIN**

where the commas represent 1 second of delay each (,,,,,, it can/should be reduced) and **\$.External.PIN** refers to the **PIN** attribute returned by the Lambda function invoked before.



“Transfer to phone number”: this block executes the transfer to PCI Pal’s platform and the route DDI created for the customer’s integration. The **Phone number** is the PCI Pal route DDI (configured ahead) that our team will supply the customer. For the “Optional parameters” enable “**Send DTMF**”, “**Use attribute**”, Type “**User Defined**” and Attribute “**delay\_and\_PIN**” as created before:



And this is the extent of the transfer to PCI Pal flow. Save and publish.  
Exported here ([link to github](#)):



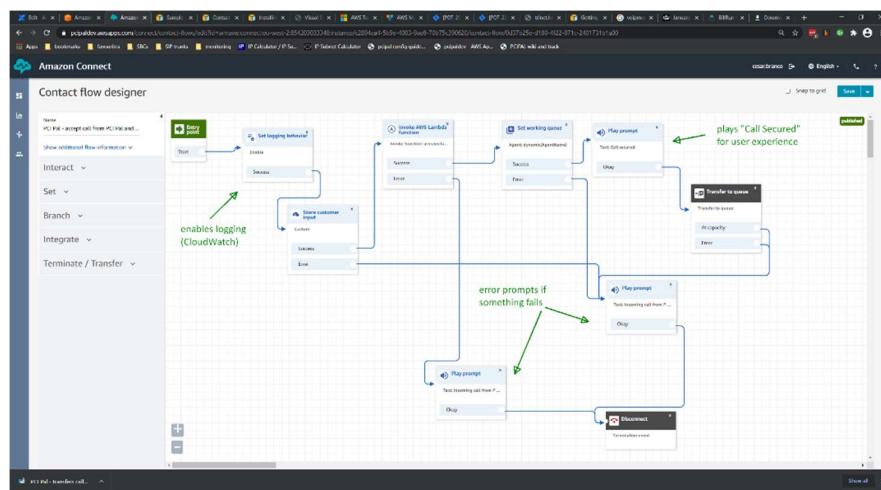
PCI Pal - transfers  
call to PCI Pal to be s

#### 4.2.1.3 Answers Call from PCI Pal - Contact Flow

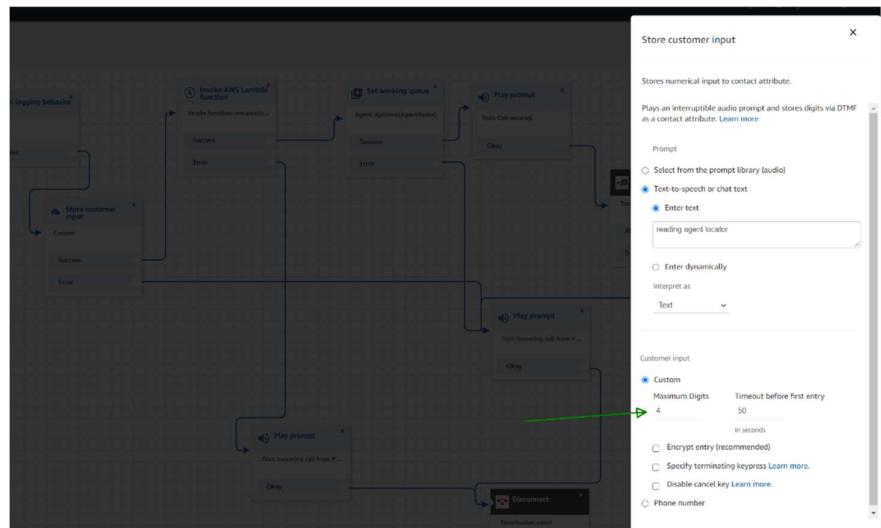
The “Answers Call from PCI Pal” Contact Flow will be answering the returning call (outgoing) from PCI Pal, completing the loop and will be associated with an Amazon Connect DID (as explained in the **DIDs** section ahead). This flow listens for the DTMFs sent by the “Transfer Call to PCI Pal” Transfer to Queue Flow, uses another Lambda function to get the agent username and then enqueue the call **directly** to that agent - taking **precedence** over any other call that may be on queue.

Amazon Connect			
Contact flows			
<input type="text"/> Search by name			
Name	Type	Description	Status
Default agent hold	Agent hold	Audio played for the agent when on hold	Published
Default agent transfer	Transfer to agent	Default flow to transfer to an agent.	Published
Default agent whisper	Agent whisper	Default whisper played to the agent.	Published
Default customer hold	Customer hold	Default audio the customer hears while on hold.	Published
Default customer queue	Customer queue	Default audio played when a customer is waiting in queue.	Published
Default customer whisper	Customer whisper	Default whisper played to the customer.	Published
Default outbound	Outbound whisper	Default flow for outbound calls.	Published
Default queue transfer	Transfer to queue	Default flow used to transfer to a queue.	Published
Original Def. agent transfer	Transfer to agent	Default flow to transfer to an agent.	Saved
Original Def. agent whisper	Agent whisper	Default whisper played to the agent.	Saved
PCI Pal - accept call from PCI Pal and deliver to the same agent	Contact Flow	Accept call from PCI Pal and deliver to the same agent by calling a lambda function after reading the incoming DTMF.	Published
PCI Pal - transfers call to PCI Pal to be secured after say...	Transfer to queue	Loop Transfer to PCI Pal saving the agent name with a lambda function that returns a PIN than will be used to find t...	Published

Full flow:

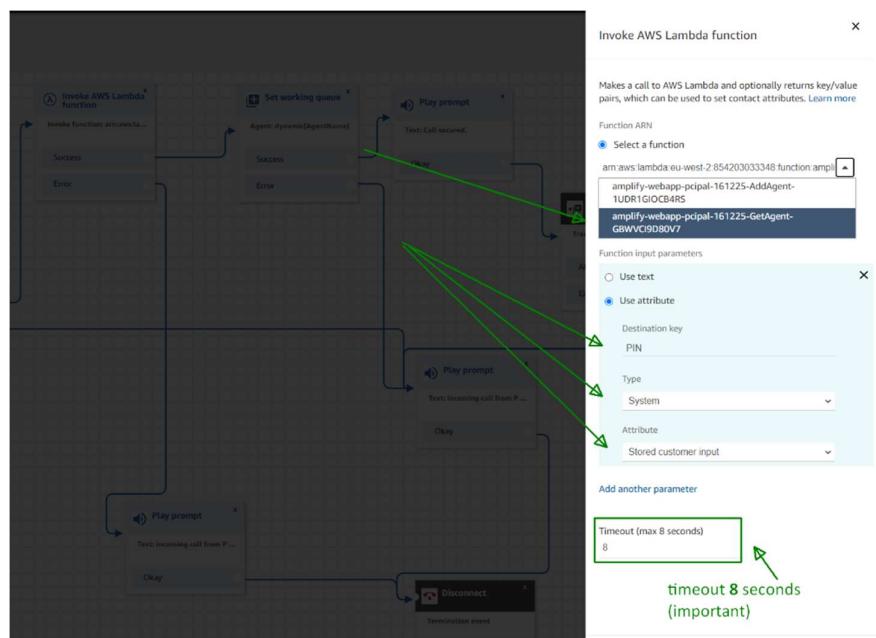


"Store customer input": listens and stores the DTMFs sent by the transfer flow. Set "**Maximum Digits**" to **4** and timeout to 30-50 seconds depending on the PSTN connection times. Add some announcement saying for user experience purposes if required like "...reading agent locator". It helps debug and keeps the agent informed at the beginning. But likely will not be used in production.

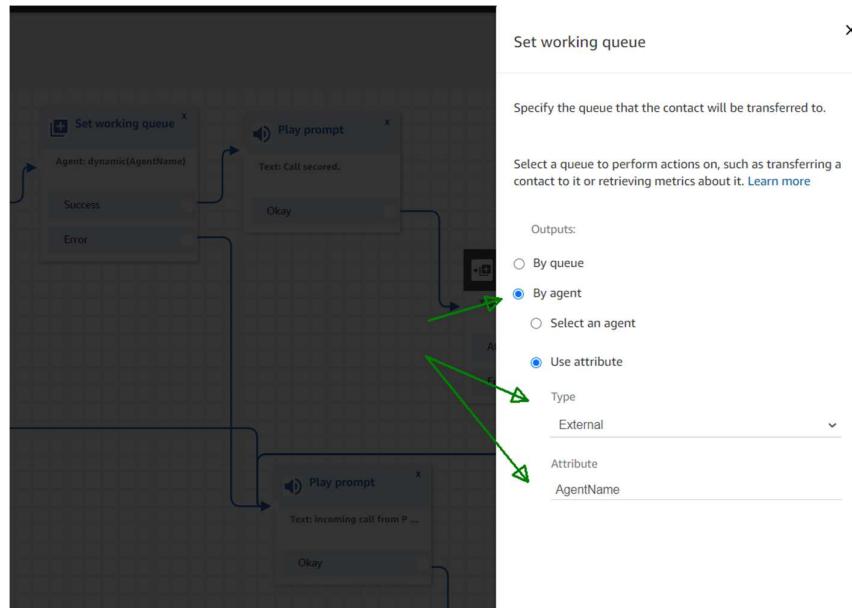


"Invoke Lambda function": this block uses the locator retrieved on the previous block and sends it to the Lambda function to find the agent username to enqueue the transferred call to (i.e. the same).

For the function select **GetAgent**, for the function input parameters select "use attribute", then "Destination key" "**PIN**", which is the name of the input variable recognized by the function. Type "**System**" and "Attribute" is "**Stored customer input**" which is the DTMF sequence just collected. Timeout **8** seconds, same as on the transfer flow.



"Set working queue": and finally set the working queue which will not be a regular queue, but the agent herself/himself, so the looped call has priority over every other call. Select "**By agent**" and then "Use attribute", "Type" "**External**" and "Attribute" "**AgentName**" which is the name of the attribute returned by the Lambda function.



Save and publish.

Exported here (link to github):

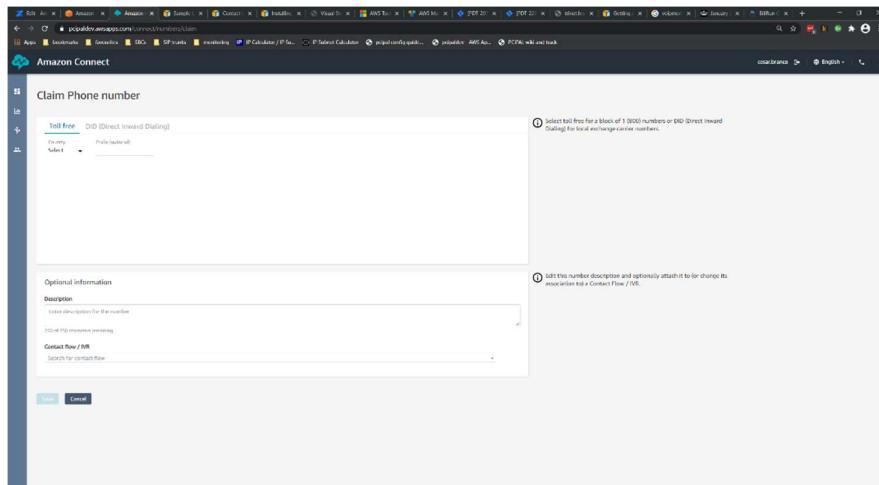


PCI Pal - accept call  
from PCI Pal and deli

#### vi. DIDs

PCI Pal requires a DID to be used by the returning leg of the loop (i.e., the outgoing call to the agent) to answer the call using the “**Answers Call from PCI Pal**” Contact flow. This is not a public DID in the sense that it is only used by PCI Pal, not the cardholder.

First claim (buy) a DID or select one already created:



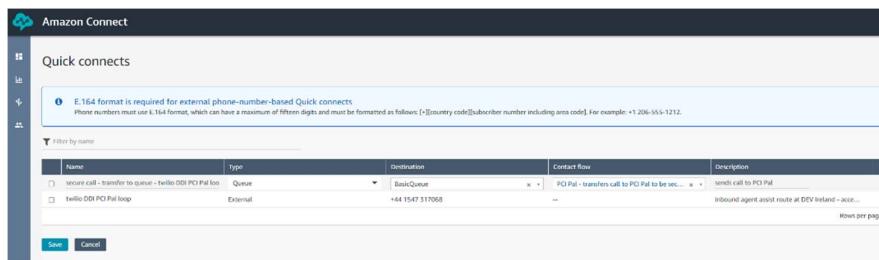
Note that AWS only allows customers to buy DIDs for the region(s) where the Amazon Connect instance is running. Then configure the DID by giving it a description and selecting the flow that answers the call:



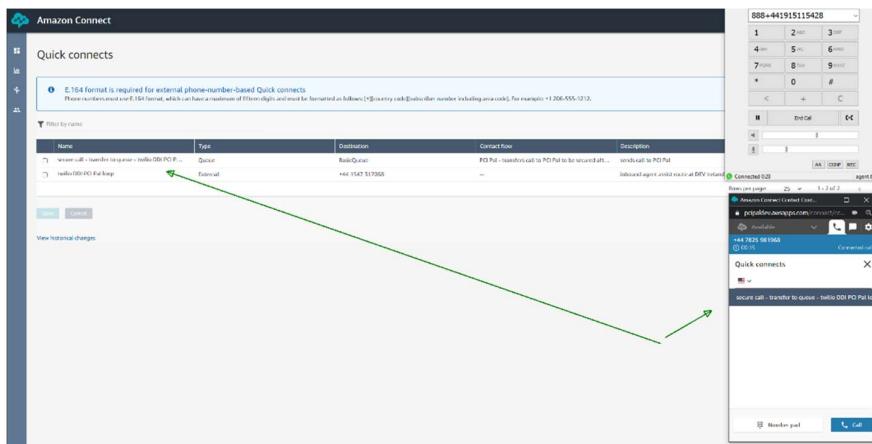
## vii. Quick Connects

A Quick Connect is a contact that will be available to the agent to click and transfer the call to when she/he needs to secure a payment. This Quick Connect is associated with a flow (transfer to queue flow) and one or more queues (which determines which agents see which Quick Connects). The queue defined on the actual Quick Connect has no impact on the integration because the call is eventually transferred not to a queue, but to a phone number - so pick one.

Depending on the customer there may be multiple “Quick Connects” that secure calls for different queues for different regions for example.



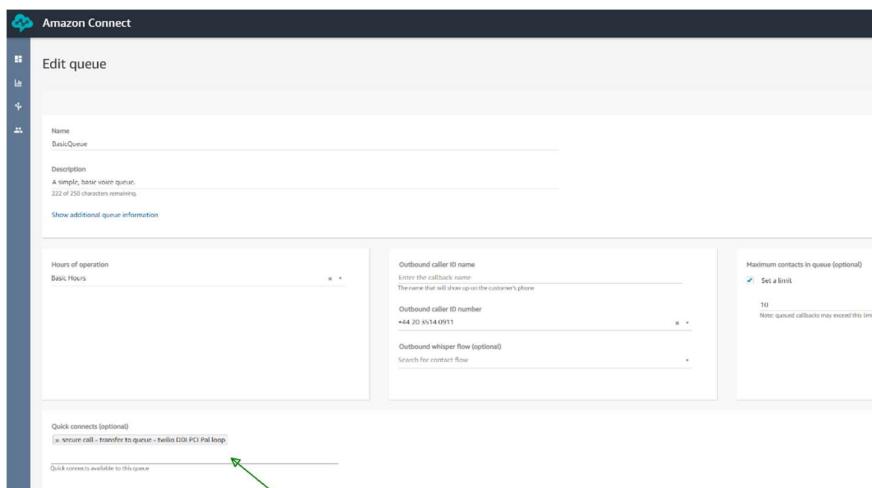
Create a “Quick Connect” suitably named, “Secure Call” or something similar, and associate with the “Transfer to PCI Pal” transfer to queue flow created before.



The “Quick connects” only show **during** call established.

### viii. Queues

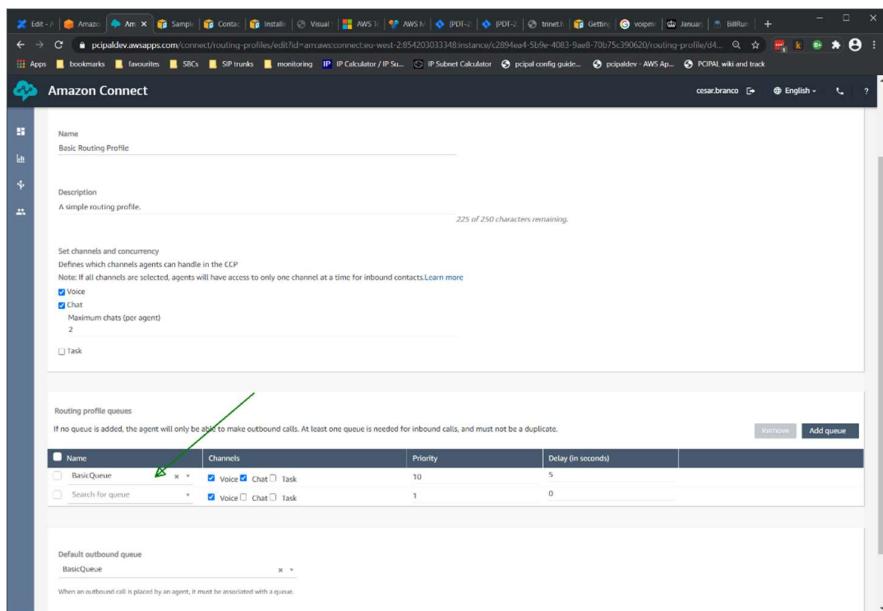
For all the queue(s) the agents may take payments on, associate the relevant secure call “Quick Connects”.



As noted before, depending on the customer, there may be multiple “Quick Connects” that secure calls for different queues for different regions for example.

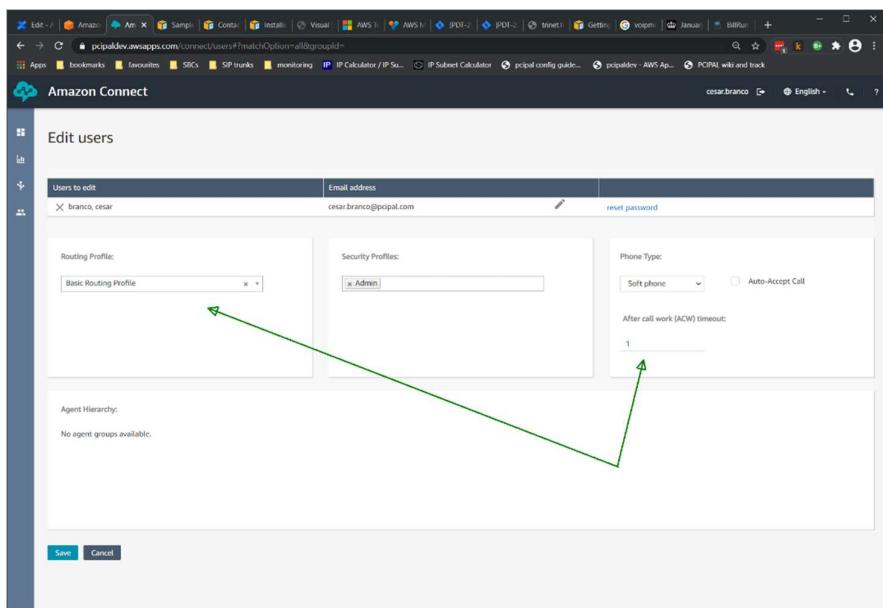
### ix. Routing Profiles

Check the agent routing profile to make sure the relevant queues are there. In our case just “Basic Queue”:



## x. Agents (User Management)

Finally, for the agents confirm they are associated with the correct routing profile (the one that has the correct queues, which are the ones that have the correct secure call quick connects...).



### NOTE:

The “After call work (ACW) timeout” value is the time the system gives the agent to finish her/his work after a customer disconnects before another call is delivered, for this example we configured **1 second**; if **0** is configured then the agent will not receive another call (aka contact) until she/he “Closes contact” on the soft phone.

## 5 Amazon Connect Agent Operation during PCI Pal Payments

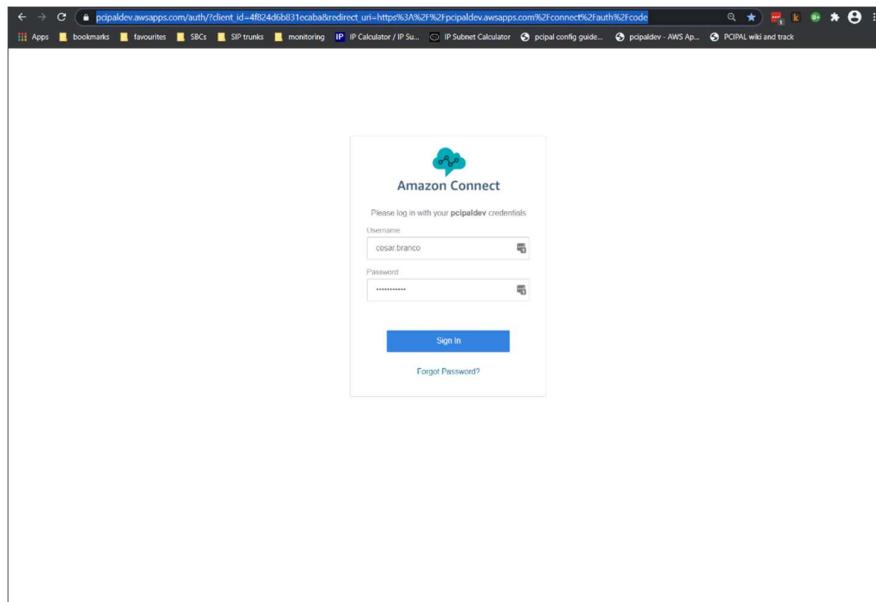
Read the [Agent training guide on AWS >>](#)

for generic guidance on agent operation (it is not complex).

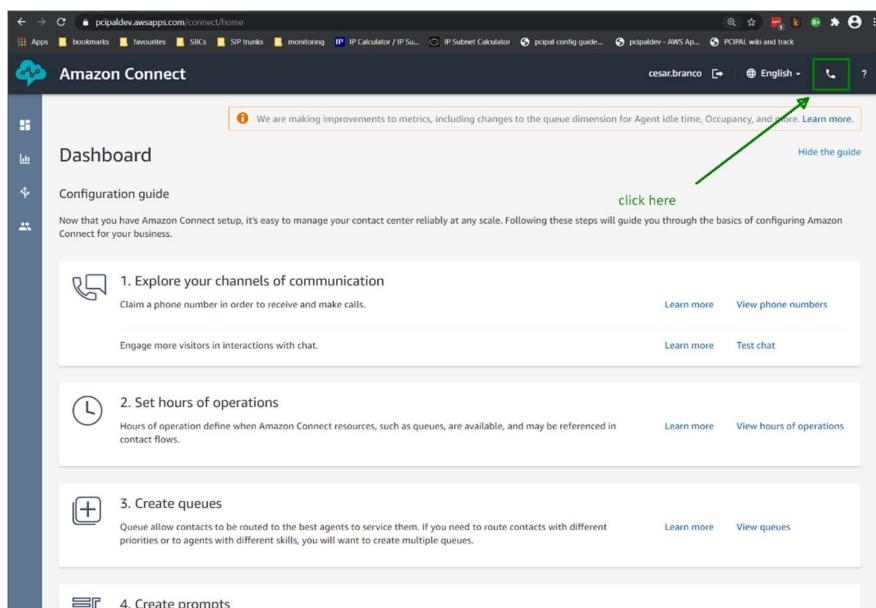
## 5.1 How to take a secure payment

This section describes what the agent must do to secure a call and take a PCI compliant payment.

Open the Amazon Connect console: <https://<your connect tenant>.awsapps.com/connect/login>:



Open the agents CCP (contact control panel):



The CCP pops up (this is a WebRTC soft client whose URL is <https://<your connect tenant>.awsapps.com/connect/ccp-v2/softphone>):

We are making improvements to metrics, including changes to the queue dimension for Agent idle time, Occupancy, and more. [Learn more](#)

[Hide the guide](#)

## Dashboard

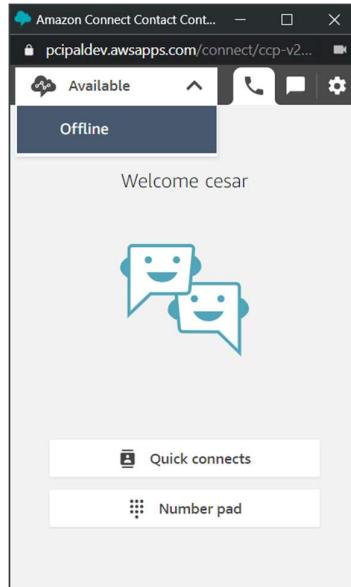
### Configuration guide

Now that you have Amazon Connect setup, it's easy to manage your contact center reliably at any scale. Following these steps will guide you through business.

- 1. Explore your channels of communication**  
Claim a phone number in order to receive and make calls.  
  
Engage more visitors in interactions with chat.
- 2. Set hours of operations**  
Hours of operation define when Amazon Connect resources, such as queues, are available, and may be referenced in contact flows.
- 3. Create queues**  
Queue allows contacts to be routed to the best agents to service them. If you need to route contacts with different priorities or to agents with different skills, you will want to create multiple queues.
- 4. Create prompts**  
Prompts are media that can be used to play back audio to customers or agents contact flows. You can upload a pre-recorded .wav file, or quickly record one through our web interface using your computer's microphone. Updates to prompts take immediate effect in all contact flows they are referenced in.

[Learn more](#) [View prompts](#)

Set the agent available:



Open the flow the agent will use to take the payment (it will be on a separate tab as usual). Production deployments will have the CRM integrated with the flows and sometimes even the CCP together with the CRM:

The screenshot shows the PCI Pal web interface with a test flow group named "Test Flow Group". It contains two steps: "JamesTest1" and "WorldPay Corporate 3DS2". Below the interface is a browser window showing the payment details for a 3DS transaction. The browser window has a title bar "euwest1.pcipaldev.cloud session/1/launched" and a search bar "Search for Tenants, Numbers, Sessions etc...". The payment form includes fields for Card Holder Name (3DS\_V2\_CHALLENGE\_IDENTIFIED), Order Code (ABC123ABC), Description, Currency (EUR), Amount (€ 29.32), and Shopper Email Address.

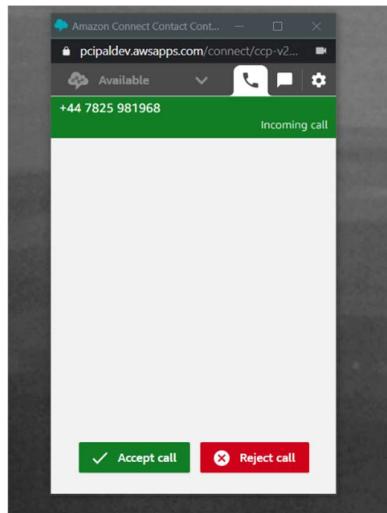
Then make a call to the PSTN number visible to the cardholder:

The screenshot shows the Amazon Connect "Manage Phone numbers" interface. It lists three phone numbers with their descriptions: "+44 1224 980821" (PCI Pal integration test - return call from PCI Pal), "+44 191 511 5428" (PCI Pal integration test - PSTN incoming), and "+44 20 3514 0911" (First Phone Number).

Note the PSTN call, the payment flow and the agent's soft phone:

The screenshot shows the PCI Pal web interface with the payment form. To the right, a soft phone interface from Amazon Connect is shown, displaying the PSTN number "+44 191 511 5428" and the message "Calling 888+441915115428... cesar.branco".

and accept the call:



The call is answered. When the agent needs to take a payment click “**Quick connects**”:

**Order Summary**

- Card Holder Name: 3DS\_V2\_CHALLENGE\_IDENTIFIED
- Order Code: ABC123ABC
- Description:
- Currency: EUR
- Amount: € 29.92
- Shopper Email Address:

**Billing Address**

- Address Line 1: 1 Road Name
- Address Line 2:
- Address Line 3:
- City: TestCity
- Post Code: TES 7ER
- Country: United Kingdom
- Telephone:

**Connected call** (+44 7825 981968)

Call controls: Hold, Mute, Number pad, Quick connects, End call.

Then double click on the “**secure call - PCI Pal etc**” quick connect created before:



Order Summary

Card Holder Name	3DS_V2_CHALLENGE_IDENTIFIED
Order Code *	ABC123ABC
Description *	
Currency *	EUR
Amount *	€ 29.32
Shopper Email Address	

Billing Address

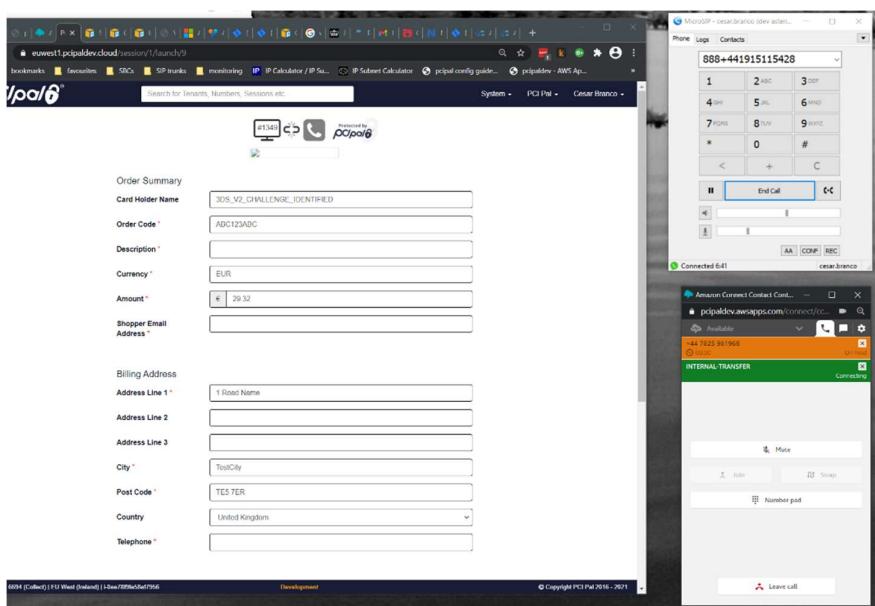
Address Line 1 *	1 Road Name
Address Line 2	
Address Line 3	
City *	TestCity
Post Code *	TE5 7ER
Country	United Kingdom
Telephone *	

Amazon Connect Contact Center... pcipaldev.awsapps.com/connect/cc... Available +44 7825 981968 03:40 Connected call Quick connects X secure call - transfer to queue - twilio DDI PCI Pal loop

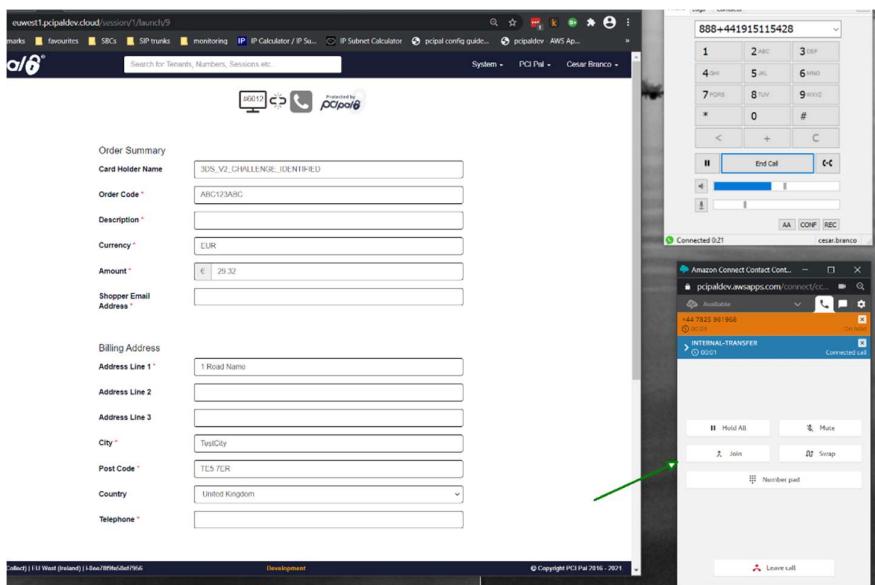
Amazon Connect initiates the 2-step transfer:

The screenshot shows two windows side-by-side. On the left is the PCI Pal Order Summary page, identical to the one above. On the right is the MicroSIP application interface, which includes a number pad and a call control panel. A green arrow points from the PCI Pal window towards the MicroSIP window, indicating the transfer process.

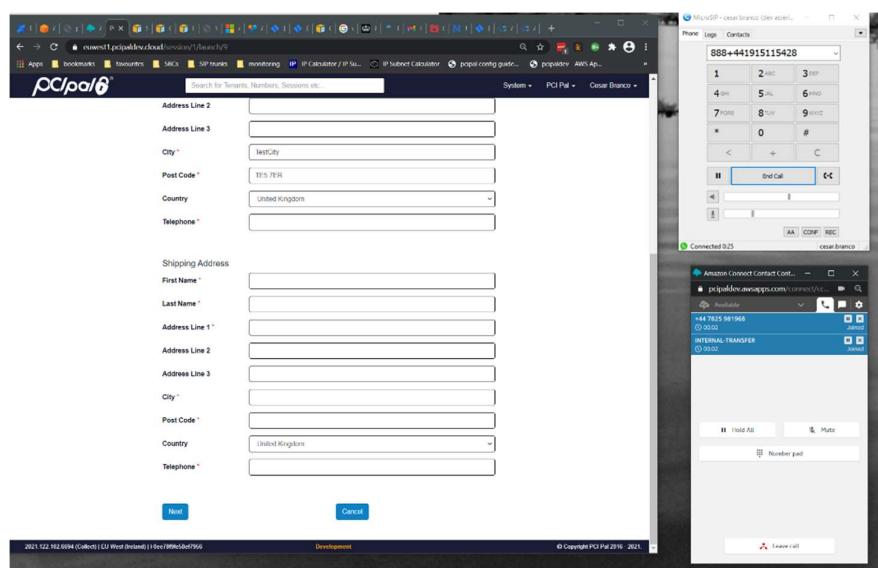
The call reaches PCI Pal and then loops back to Amazon Connect:



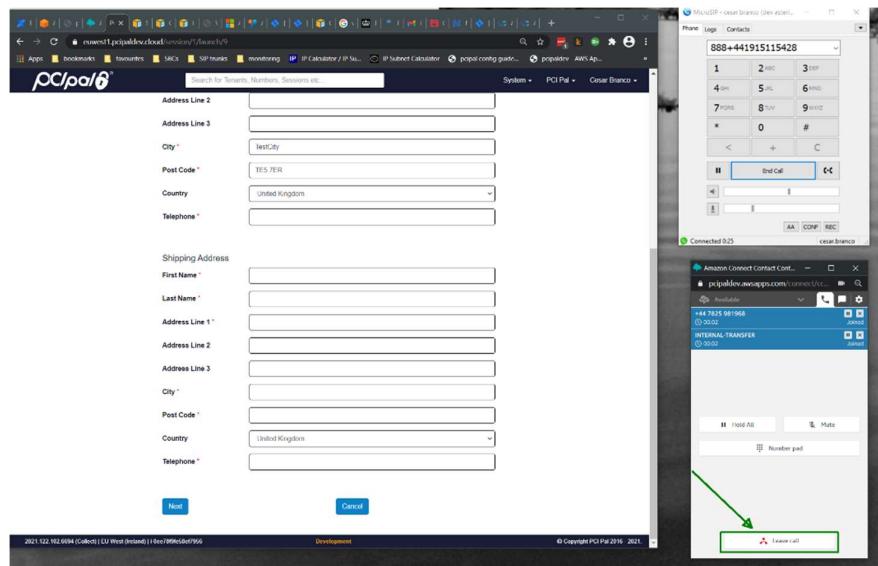
Click “Join” so the cardholder is no more than 1 or 2 seconds on music:



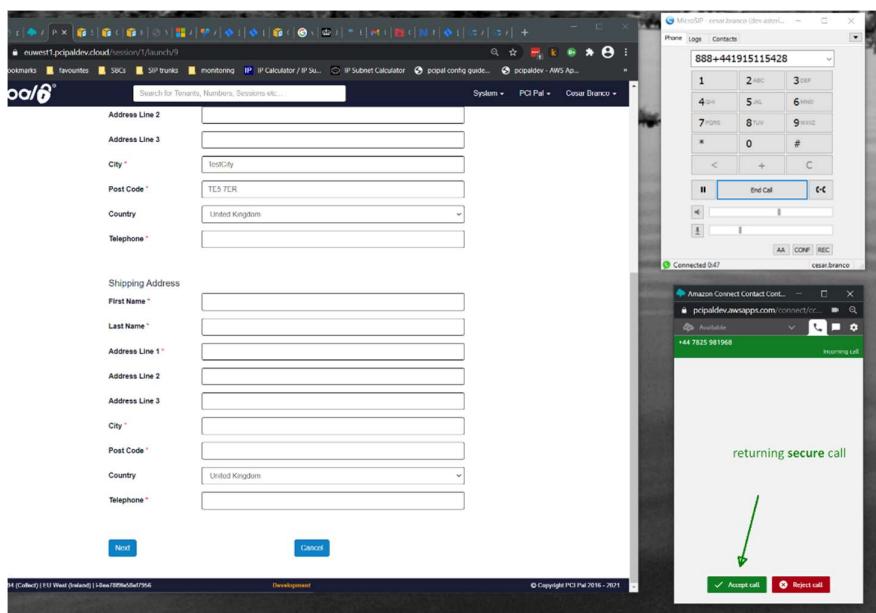
The agent and the cardholder are joined and talking again:



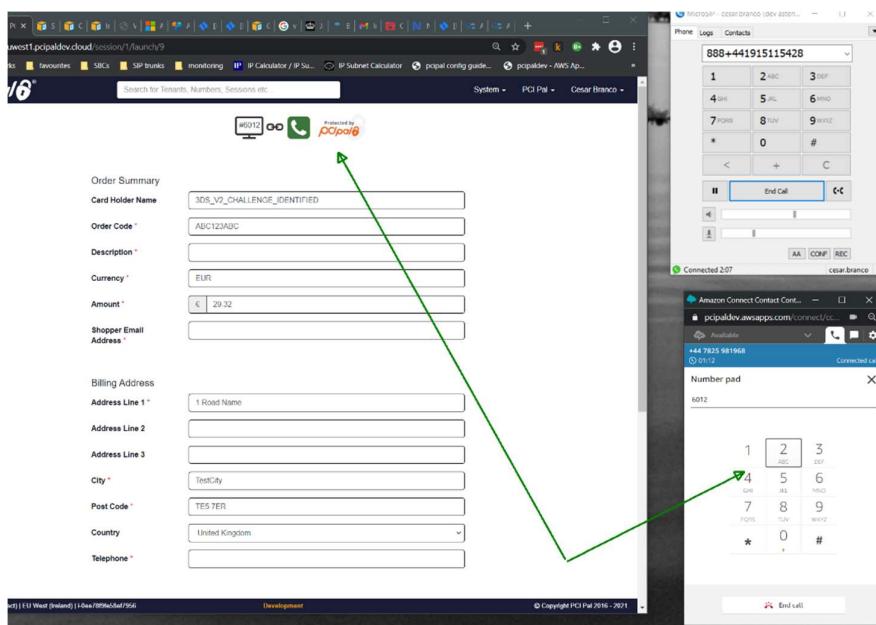
The “**accept call from PCI Pal**” flow will say “**call secure**” (wait for the prompt “call secure”) and the agent can “**leave call**” (after hearing “call secure”):



Which will immediately ring again on the agent personal queue, the agent clicks “**accept call**” and is reunited with the cardholder:



Click "Number pad", and enter the PIN to link the session:



And take the payment:

Card Number	<input type="text" value="••••"/> <span>C</span>
Expiry Date (MMYY)	<input type="text"/> <span>C</span>
Security Code (Cv2)	<input type="text"/> <span>C</span>

Back Cancel

The call will remain looped through PCI Pal until either the cardholder or the agent disconnects.

And this is the end of the functional integration.

The following diagram represents the logical functionality that is achieved via this integration.

