Amazon Connect Project Documentation

Designing and Implementing a Call Center for a Small to Mid-Sized Restaurant Using Amazon Connect

Index

1. **Project Overview**
2. **Business Use Case**
3. **Architecture Overview**
4. **Prerequisites**
5. **Infrastructure Provisioning with Terraform**

5.1 Terraform Structure

5.2 Amazon Connect Instance Provisioning

5.3 AWS Lambda Function Provisioning

5.4 IAM Roles and Permissions

6. **Post-Provisioning Configuration in Amazon Connect Console**

6.1 Contact Flow Design

6.2 Routing Profiles and Queues

6.3 Phone Number Claim and Association

7. **Call Handling and Agent Workflow**
8. **Post-Call Lambda Trigger and Payment Link Delivery**
9. **Testing and Validation**
10. **Security Considerations**
11. **Cost Considerations**
12. **Limitations and Known Gaps**
13. **Future Enhancements**
14. **Proof of Concept**

1. Project Overview

This project focuses on the design and implementation of a cloud-based call center for a small to mid-sized restaurant using Amazon Connect. The solution enables restaurant staff (agents) to receive customer calls for reservations, takeaway orders, or general inquiries, and automatically sends a payment link to customers after a call is completed.

Core infrastructure components such as the Amazon Connect instance and AWS Lambda functions are provisioned using Terraform. Operational configurations, including contact flows, routing profiles, and phone number associations, are completed manually through the Amazon Connect console.

2. Business Use Case

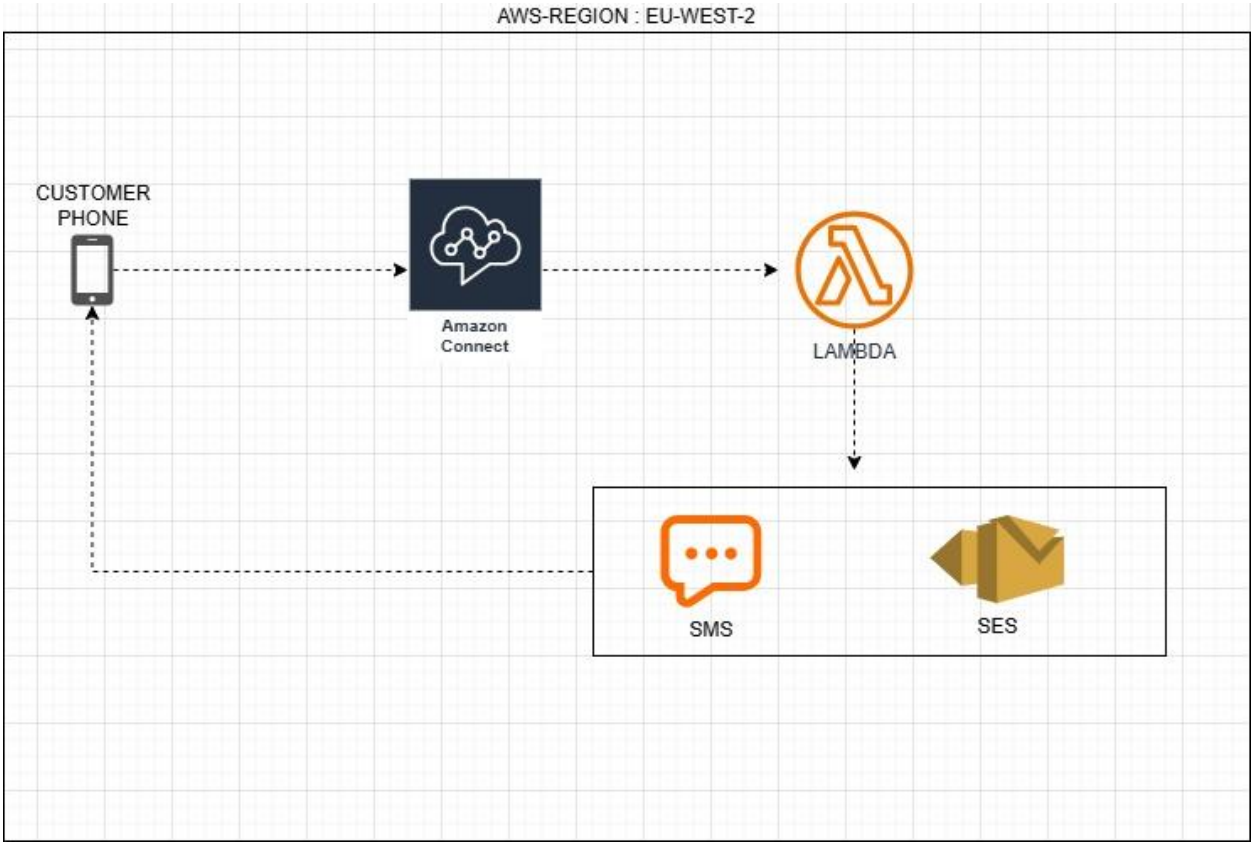
Restaurants often rely on phone calls for:

- Table reservations
- Takeaway or delivery orders
- Event bookings and large group inquiries

Manual payment handling during calls can slow down operations and introduce errors. This solution streamlines the process by allowing agents to handle the call normally and automatically triggering a Lambda function after the call to send a secure payment link to the customer via SMS or email.

This approach improves customer experience, reduces call handling time, and enables faster payment collection.

3. Architecture Overview



Key Components:

Connect Instance: UK-based call center with DID phone number

Lambda Function: Dual-channel payment link sender (SMS + Email)

SNS: SMS delivery

SES: Email delivery with HTML formatting

Manual Contact Flows: Created in Connect console manually for call routing.

Call flow summary:

1. Customer calls the restaurant phone number
2. Amazon Connect executes the contact flow
3. Call is routed to an available agent
4. Agent completes the interaction (order or reservation)
5. After the call ends, a Lambda function is triggered (can be automatic or agent-initiated)

6. Lambda sends a payment link to the customer

4. Prerequisites

- AWS account with permissions for Amazon Connect, IAM, and Lambda
- Terraform installed locally
- AWS CLI configured
- Basic understanding of Amazon Connect contact flows and queues

5. Infrastructure Provisioning with Terraform

Terraform is used to provision foundational infrastructure components required for the call center. Full configuration files can be found in this GitHub repo.

5.1 Terraform Structure

The Terraform configuration is organized into logical files or modules:

- Provider configuration
- Amazon Connect instance resource
- AWS Lambda function resource
- IAM roles and policies

This structure supports maintainability and future expansion.

5.2 Amazon Connect Instance Provisioning

Using Terraform, the Amazon Connect instance is provisioned with:

- A unique instance alias
- Identity management configuration
- Basic access settings

The instance acts as the central platform for call routing and agent management.

5.3 AWS Lambda Function Provisioning

The Lambda function responsible for sending payment links is provisioned using Terraform. It includes:

- Runtime and handler configuration
- Source code or deployment package

This function is invoked after the call completes. (can be automatic or agent-initiated)

5.4 IAM Roles and Permissions

IAM roles and permissions are configured to:

- Allow Amazon Connect to invoke the Lambda function
- Allow Lambda to log execution details to CloudWatch

Least-privilege access is applied to reduce security risks.

6. Post-Provisioning Configuration in Amazon Connect Console

Certain Amazon Connect features are configured manually through the AWS Management Console.

6.1 Contact Flow Design

The contact flow is designed to:

- Greet the customer
- Route the call to the appropriate queue
- Capture customer attributes such as phone number or email
- Trigger post-call processing

The contact flow focuses on simplicity to reduce call handling time.

6.2 Routing Profiles and Queues

Routing profiles are created to:

- Assign agents to queues
- Control agent availability
- Define call priority

Queues represent different call types, such as reservations or orders.

6.3 Phone Number Claim and Association

A phone number is claimed within Amazon Connect and associated with:

- The Amazon Connect instance
- The restaurant's main inbound contact flow

This enables customers to reach the restaurant through a single public number.

7. Call Handling and Agent Workflow

Agents receive calls through the Amazon Connect contact control panel. During the call, agents:

- Confirm customer details
- Take orders or reservations
- Inform customers that a payment link will be sent after the call

Agents do not process payments directly, reducing call duration and complexity.

8. Post-Call Lambda Trigger and Payment Link Delivery

After the call ends, Amazon Connect triggers a Lambda function that:

- Retrieves call attributes (customer phone number or email)
- Generates a payment link using a third-party payment provider
- Sends the link to the customer via SMS or email

This asynchronous processing ensures the agent can move on to the next call immediately.

9. Testing and Validation

The solution is validated by:

- Placing test calls to the restaurant phone number
- Verifying agent call handling

- Confirming Lambda invocation after call completion
- Ensuring payment links are successfully delivered

CloudWatch logs are used for troubleshooting.

10. Security Considerations

Security best practices include:

- Least-privilege IAM roles
- Secure handling of customer contact details
- Restricted access to payment provider credentials

Sensitive data is never hardcoded in Lambda functions.

11. Cost Considerations

Costs are driven by:

- Amazon Connect call minutes
- Phone number rental fees
- Lambda execution costs
- Messaging costs for SMS or email

This serverless architecture scales cost-effectively with call volume.

12. Limitations and Known Gaps

- Contact flows and queues are manually configured
- Payment provider integration depends on external APIs
- Advanced reporting requires additional services
- SNS and SES services needs request for production access for flexible message sending.

13. Future Enhancements

Possible enhancements include:

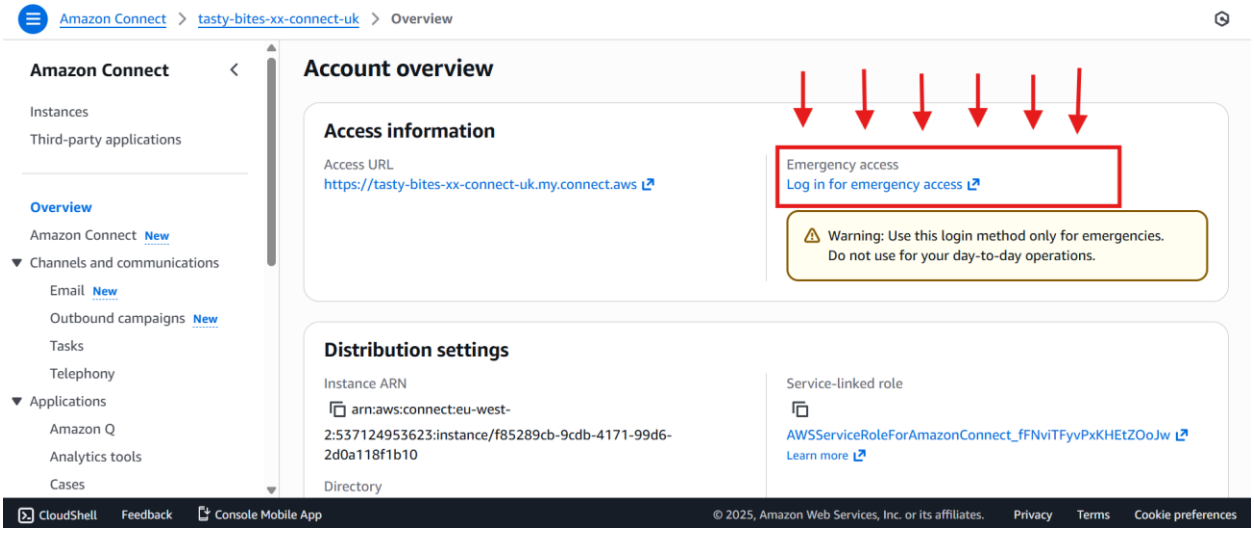
- Amazon Lex integration for automated ordering
- CRM or POS system integration
- Automated receipts and order confirmations
- DynamoDB to store customer history and preferences

14. Proof of Concept

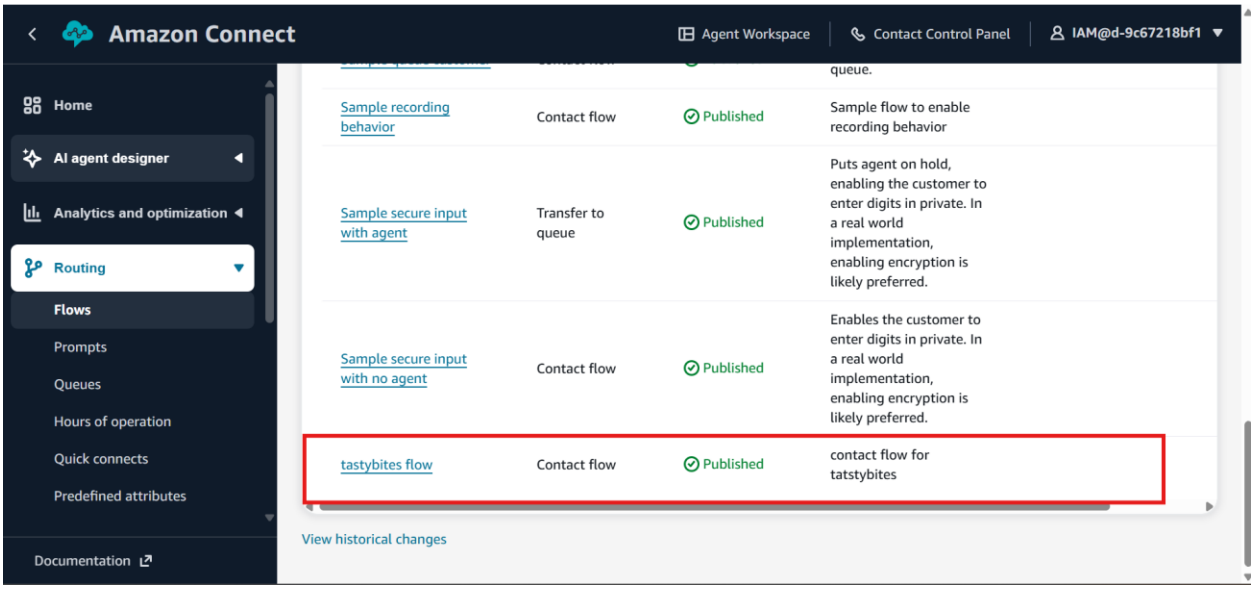
After Terraform apply the following will be provisioned:

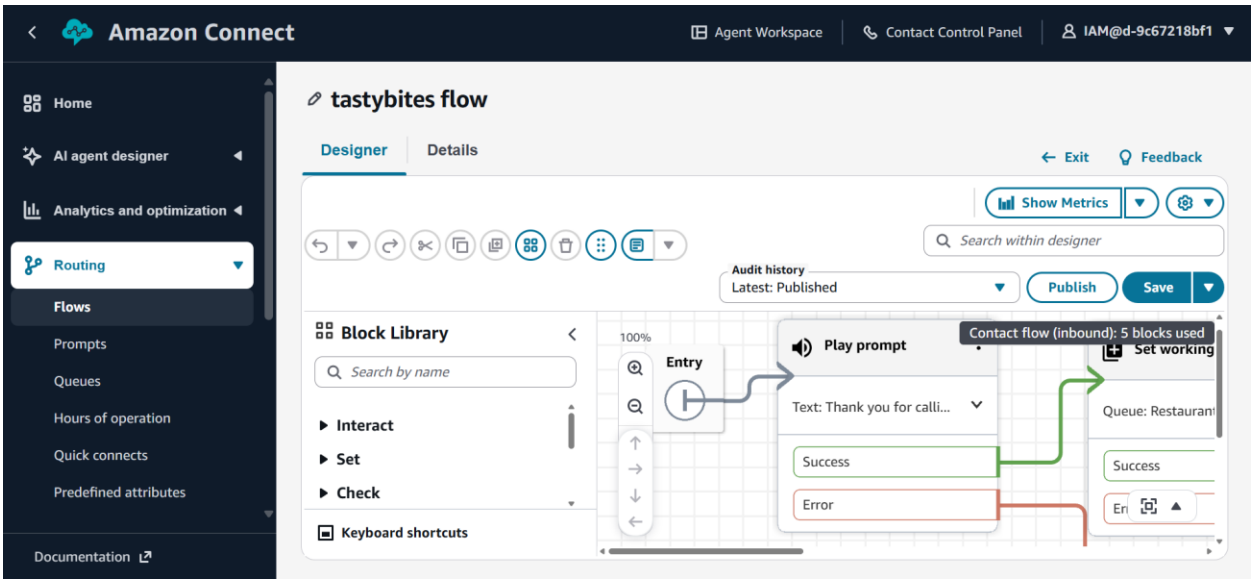
- **Connect instance -** Core call center platform
- **Phone number -** Customers can call
- **Queue -** Which calls will be routed to
- **Hours of operation -** Business hours defined
- **Lambda function -** SMS payment functionality

You log into the admin console by accessing the emergency access url which allows you to manage users and assign them to various queues, create contact flows, associate the phone number with the contact flow and configure the routing profiles. Since this was a demo, I used the admin console for testing but it is advisable to create users for daily activities.

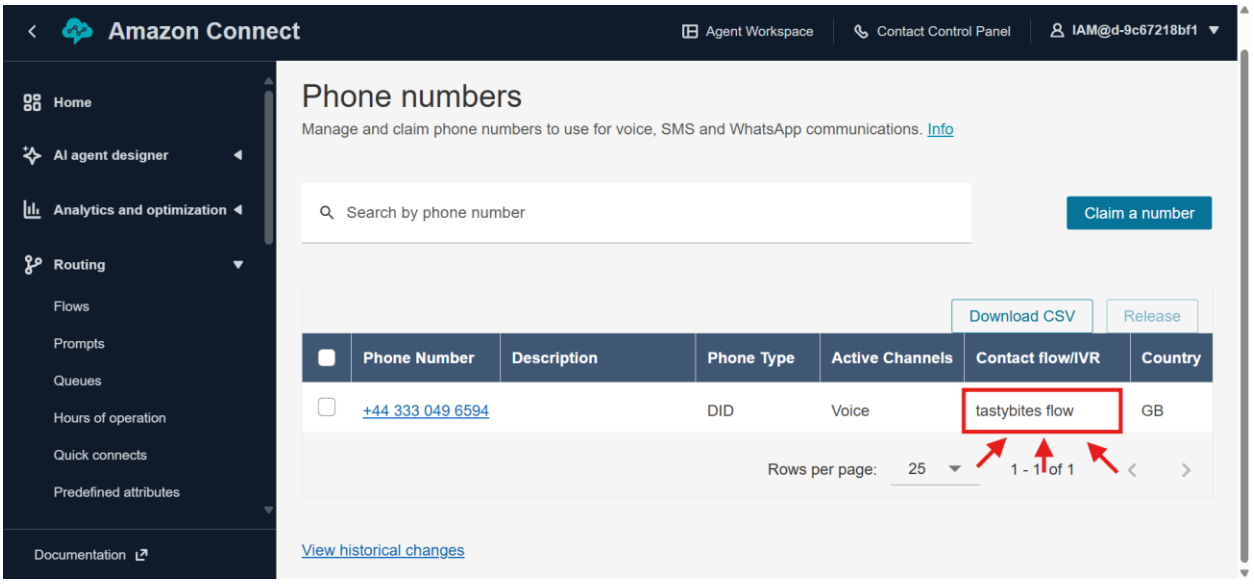


Create contact flow

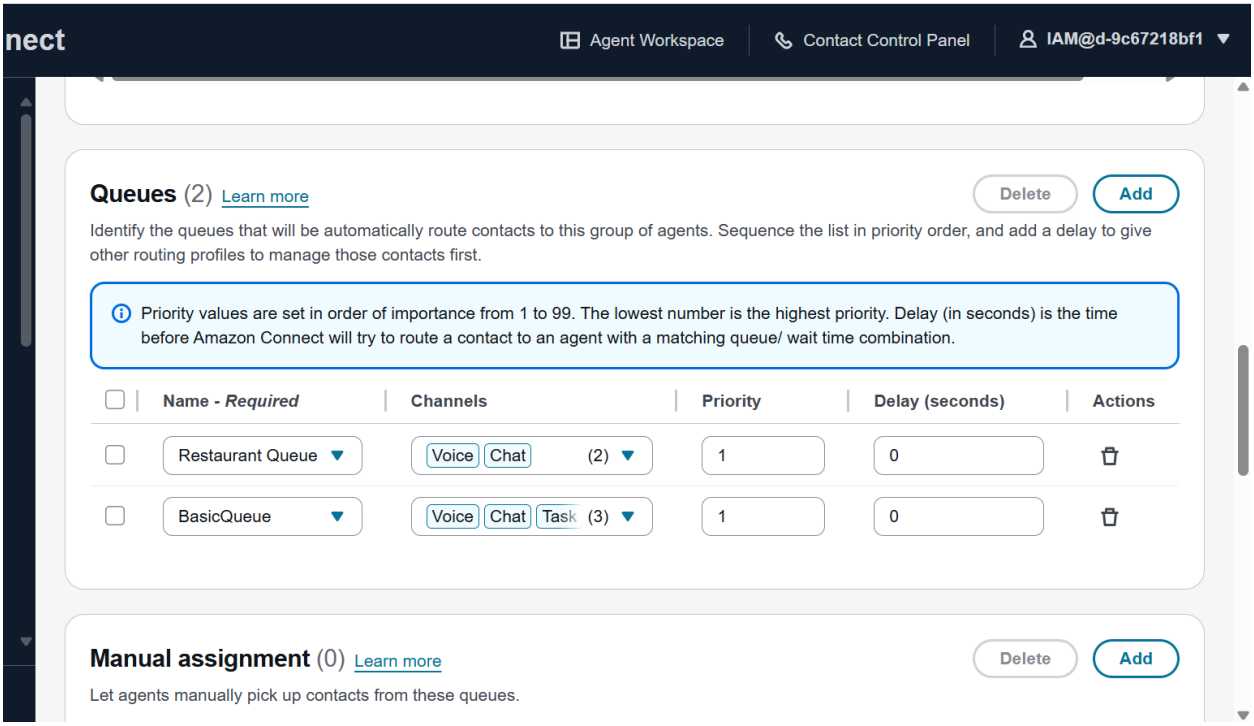


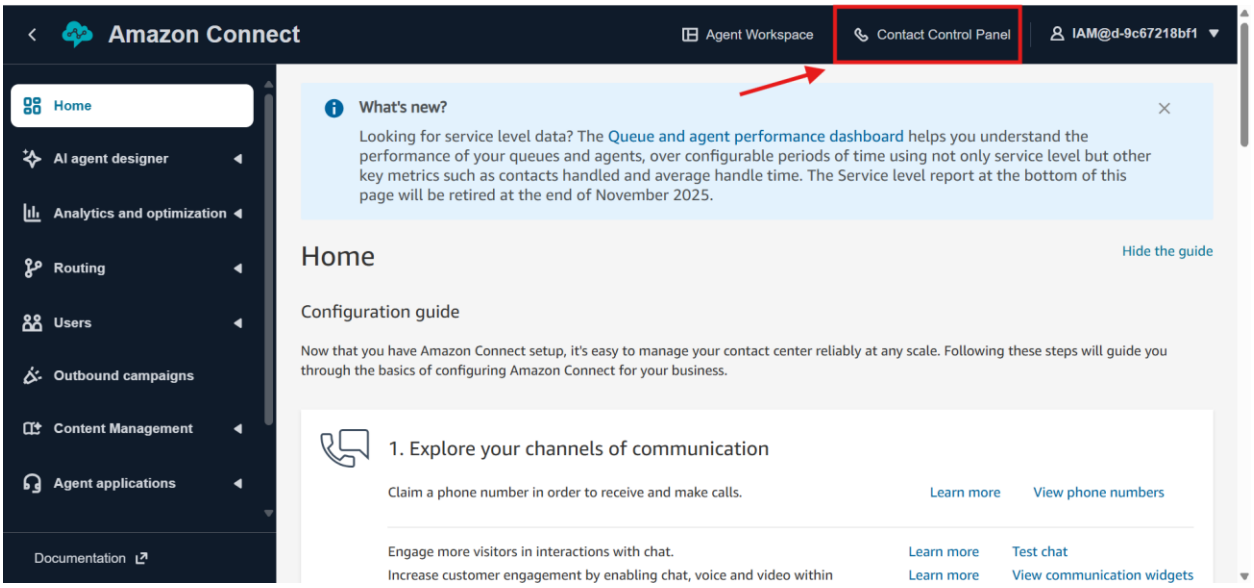


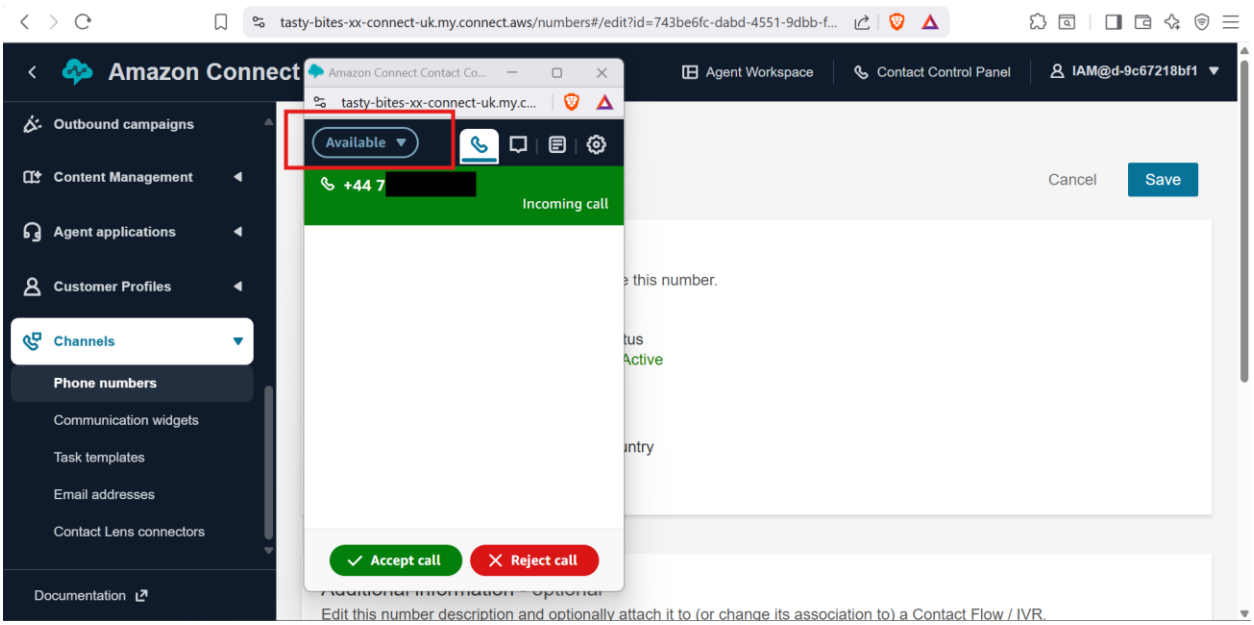
Associate phone number with contact flow. Phone number will be useless without a contact flow.



For agents to receive calls you need to assign a queue to the agent under the routing profiles in user management, Access the contact control panel, Set agent status to Available.






tasty-bites-xx-connect-uk.my.connect.aws/numbers#/edit?id=743be6fc-dabd-4551-9dbb-f...
Amazon Connect
Agent Workspace
Contact Control Panel
IAM@d-9c67218bf1
Outbound campaigns
Content Management
Agent applications
Customer Profiles
Channels
Phone numbers
Communication widgets
Task templates
Email addresses
Contact Lens connectors
Documentation
Amazon Connect Contact Co...
tasty-bites-xx-connect-uk.my.c...
Available
+44 7
Incoming call
Accept call
Reject call
Cancel
Save
this number.
tus
Active
untry
Edit this number description and optionally attach it to (or change its association to) a Contact Flow / IVR.

After call disconnects a payment link is sent to the customer, it can be agent initiated or automatic due to how you configure your contact flow. I used an email in this demo since I couldn't verify a phone number in AWS SNS, due to some limitations but could verify an email in SES. You can also opt for third party tools like Twilio in your Lambda function for SMS delivery.