DEPLOYMENT GUIDE

SnapLogic for Amazon Connect

SnapLogic for Amazon Connect

To deploy SnapLogic for Amazon Connect, the following items are installed in the customer environment by running the Cloud Formation template in the Marketplace listing. **Note**: for successful deployment the user must have existing accounts and permissions in these resources:

- Amazon Connect
- Lambda
- SnapLogic

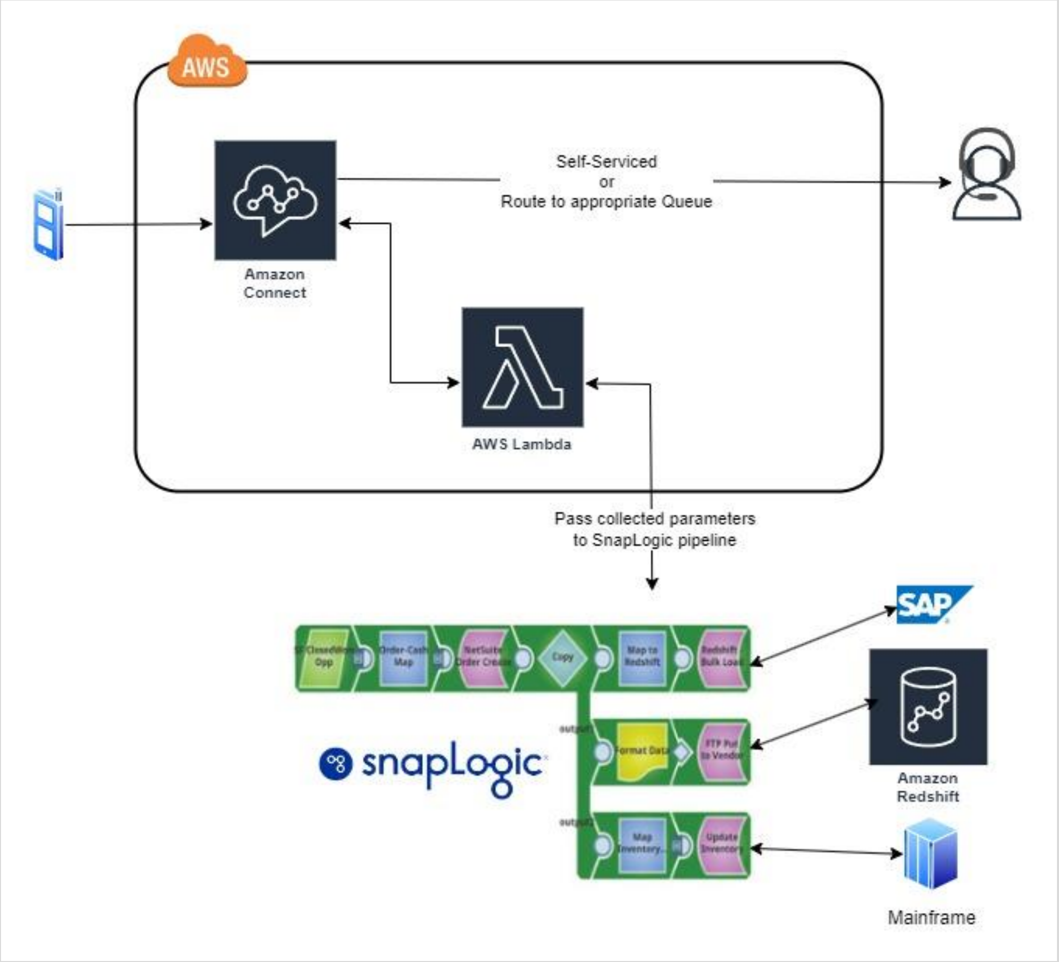


Figure 1: Sample Use case and workflow

Once the items have been installed, accounts need to be configured to run the Flow and call the pipeline.

Prerequisites

You should have the following prerequisites to get started with SnapLogic and Amazon Connect Contact Flows integration solution:

- SnapLogic Account
- Amazon Connect Account
- Salesforce Account & custom object
- ServiceNow Account
- Email Account

Sample Flow

The sample flow provided is one of the common and demanding use cases for call centers, scheduling a service call.

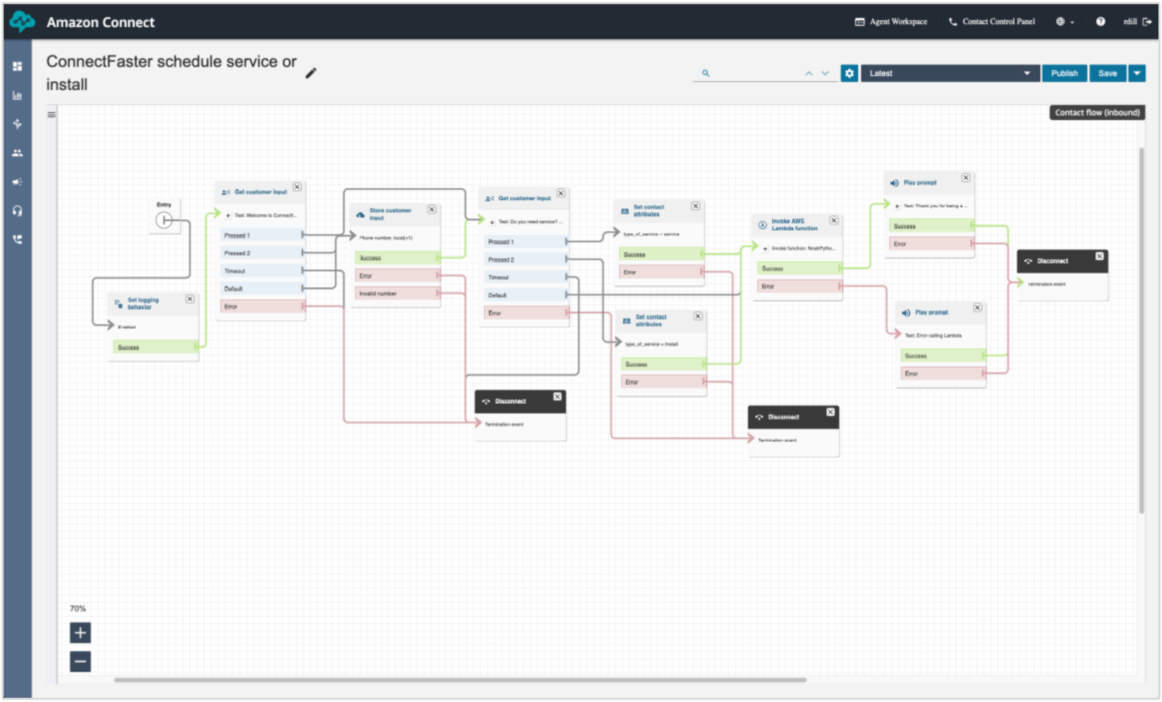
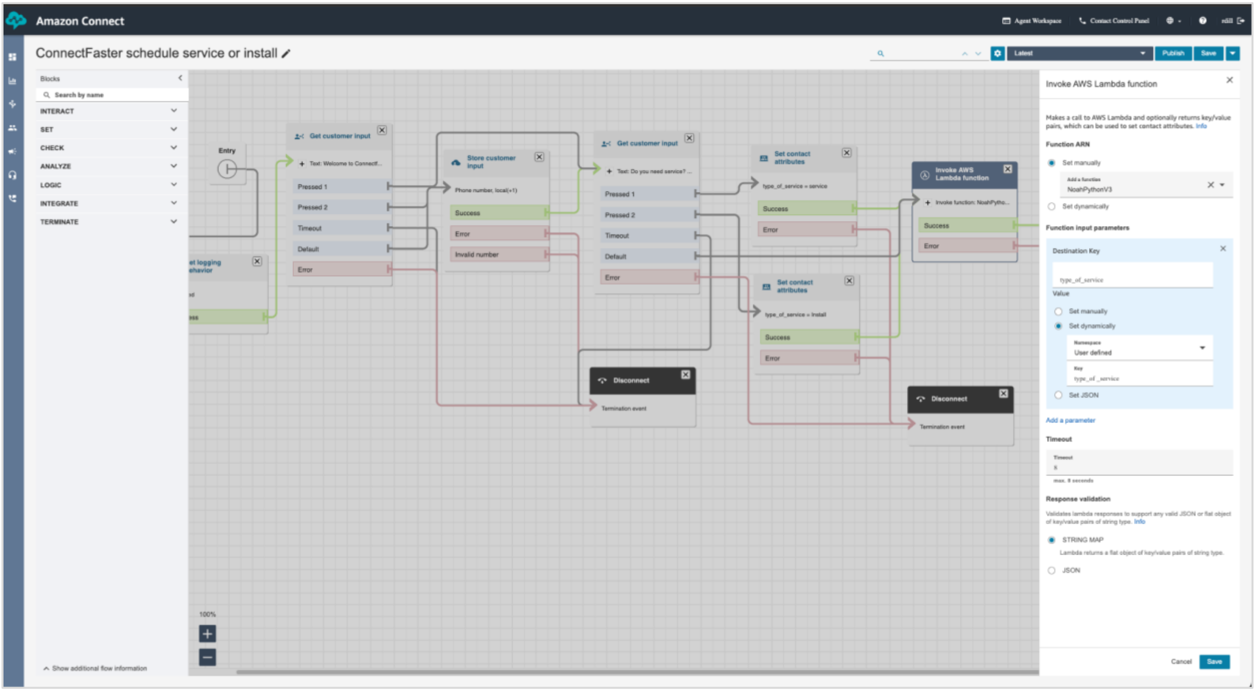


Figure 2: Amazon Connect Flow

A customer calls in to schedule an install or request a repair on site. First we have to determine the earliest availability of a technician to be able to go on site. Then, based on the request, we have to make sure they have the parts and tools necessary to perform the service or do the install.

The Amazon Connect Flow uses a Lambda function to call a SnapLogic pipeline.



Invoke AWS Lambda function

Makes a call to AWS Lambda and optionally returns key/value pairs, which can be used to set contact attributes. Info

Function ARN

● Set manually

Add a function: NoahPythonV3

○ Set dynamically

Function input parameters

Destination Key: type_of_service

Value

○ Set manually

● Set dynamically

Namespace: User defined

Key: type_of _service

○ Set JSON

Add a parameter

Timeout

Timeout: 8

max. 8 seconds

Response validation

Validates lambda responses to support any valid JSON or flat object of key/value pairs of string type. Info

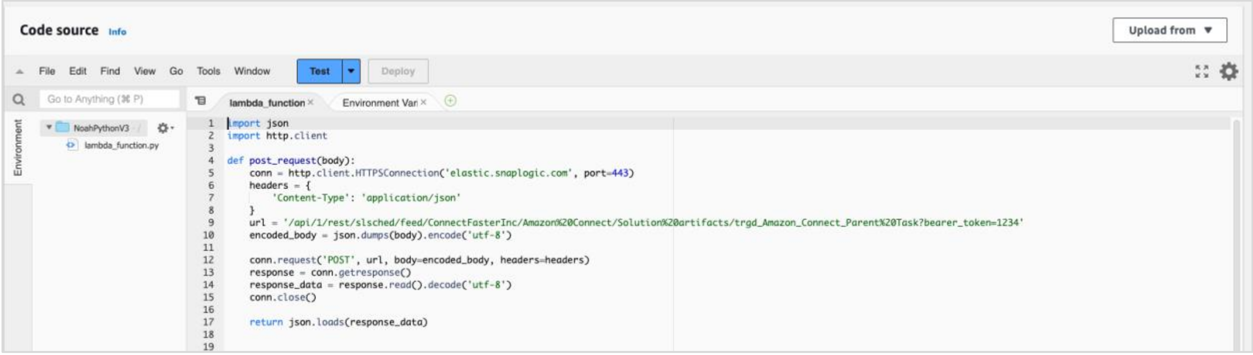
● STRING MAP

Lambda returns a flat object of key/value pairs of string type.

○ JSON

Cancel Save

Figure 3: How Amazon Connect calls the Lambda function



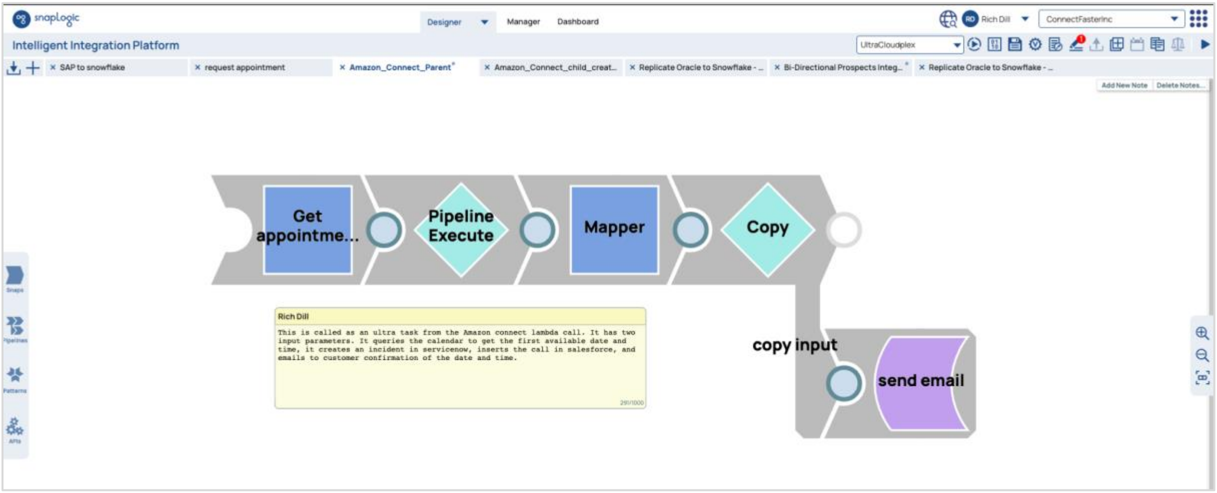
```python
import json
import http.client

def post_request(body):
    conn = http.client.HTTPSConnection('elastic.snaplogic.com', port=443)
    headers = {
        'Content-Type': 'application/json'
    }
    url = '/api/1/rest/slsched/feed/ConnectFasterInc/Amazon%20Connect/Solution%20artifacts/trgd_Amazon_Connect_Parent%20Task?bearer_token=1234'
    encoded_body = json.dumps(body).encode('utf-8')

    conn.request('POST', url, body=encoded_body, headers=headers)
    response = conn.getresponse()
    response_data = response.read().decode('utf-8')
    conn.close()

    return json.loads(response_data)
```

Figure 4: Lambda function

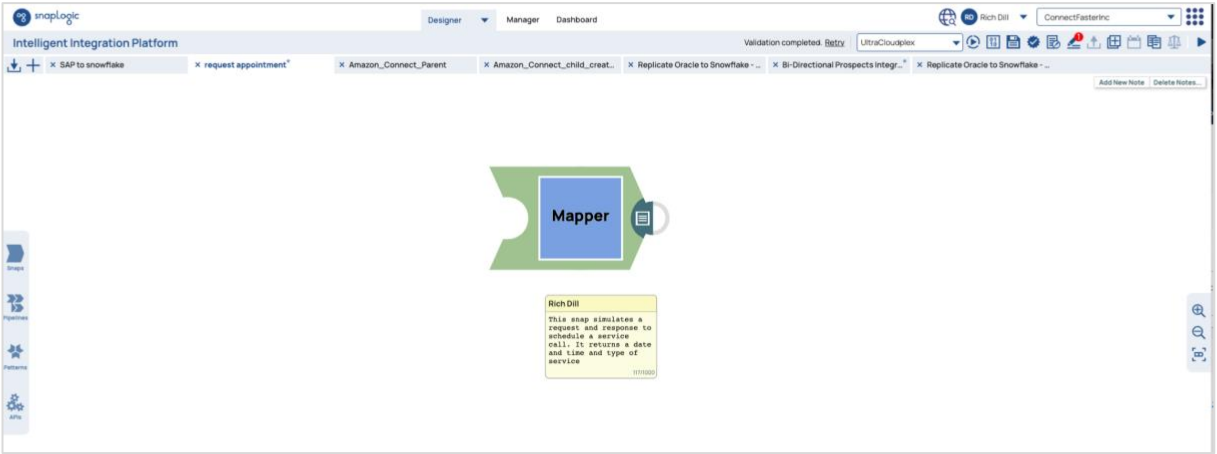


Rich Dill

This is called as an ultra task from the Amazon connect lambda call. It has two input parameters. It queries the calendar to get the first available date and time, it creates an incident in servicenow, inserts the call in salesforce, and emails to customer confirmation of the date and time.

291/1000

Figure 5: Parent pipeline

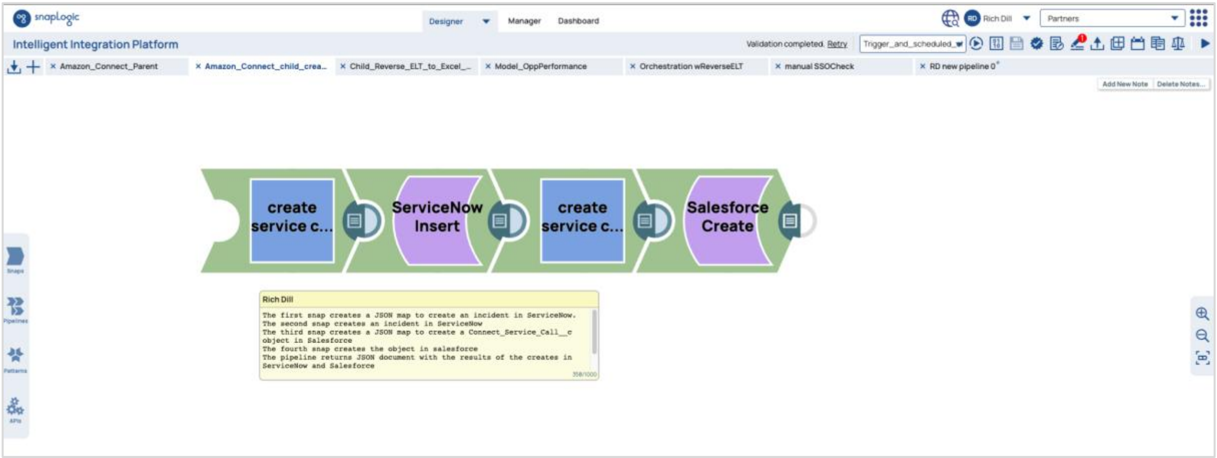

Intelligent Integration Platform
Designer
Manager
Dashboard
Rich Dill
ConnectFasterInc
Validation completed. Retry
UltraCloudplex
SAP to snowflake
request appointment*
Amazon_Connect_Parent
Amazon_Connect_child_creat...
Replicate Oracle to Snowflake - ...
Bi-Directional Prospects Integr...*
Replicate Oracle to Snowflake - ...
Add New Note
Delete Notes...
Mapper
Snaps
Pipelines
Patterns
APIs

Rich Dill

This snap simulates a request and response to schedule a service call. It returns a date and time and type of service

117/1000

Figure 6: Calendar pipeline



Rich Dill

The first snap creates a JSON map to create an incident in ServiceNow.
The second snap creates an incident in ServiceNow
The third snap creates a JSON map to create a Connect_Service_Call__c object in Salesforce
The fourth snap creates the object in salesforce
The pipeline returns JSON document with the results of the creates in ServiceNow and Salesforce

358/1000

Figure 7: Child pipeline

How to run the sample flow

- Create a phone number in Amazon Connect and associate it with the “ConnectFaster schedule service or install” Flow
- Create a salesforce account
- Create a custom salesforce object:

  Connect_Service_Call__c, fields

  "Date_and_time_of_service_call__c": dateTime

  "Customer_Account_Number__c": String

  "ServiceNow_ticket_number__c": String

  "OwnerId": "your ownerID"

  "Type_of_service__c": Install or service

- Create an SMTP email account
- Create a ServiceNow account
- Open the SnapLogic pipelines and add the accounts to the appropriate snaps
- Change the email snap to send the email to yourself

To run the demo, call the phone number you configured in Amazon Connect in the steps above.

Summary

SnapLogic for Amazon Connect shows how the user can create flows in Amazon Connect, using a standard Lambda function, call a SnapLogic pipeline to read, write and update data for a variety of sources and targets.

For more information: https://docs-snaplogic.atlassian.net/wiki/spaces/SD/overview

CloudFormation template for AWS Connect, Lambda, and SnapLogic integration

This CloudFormation template provides a robust and technically detailed solution for integrating AWS Connect, AWS Lambda, and SnapLogic. By following the steps outlined in this documentation, you can deploy the required resources on AWS, configure AWS Connect for call handling, and set up Lambda to trigger a SnapLogic pipeline execution.

Prerequisites

Before proceeding with the CloudFormation deployment, ensure you have the following prerequisites in place:

1. **AWS account:** You need an AWS account with the necessary permissions to create AWS Connect instances, Lambda functions, IAM roles, and CloudFormation stacks.
2. **SnapLogic account:** Obtain a valid SnapLogic account with appropriate credentials to access the SnapLogic API and invoke pipelines.
3. **Import SnapLogic SLP zip file:** Import zip file into your project space containing the required SnapLogic pipelines.
4. **Lambda function code:** Prepare the Lambda function code to invoke the SnapLogic pipeline. The code should include secure authentication mechanisms to interact with the SnapLogic API.

Deployment Steps

Follow these detailed technical steps to deploy the CloudFormation stack and set up the AWS Connect, Lambda, and SnapLogic integration.

1. **Access AWS Console:** Log in to the AWS Management Console.
2. **Open CloudFormation:** Navigate to the CloudFormation service.
3. **Create stack:** Click on "Create Stack" and select "With new resources (standard)" option.
4. **Specify template:** Upload the provided CloudFormation template YAML file.
5. **Stack details:**
    a. Stack name: Enter a unique stack name, acting as the identifier.
    b. Lambda function name: Specify a unique name for the Lambda function.
    c. SnapLogic pipeline import: Import the zip file containing the required pipelines for this configuration into your project space.
    d. SnapLogic API endpoint: Provide the endpoint URL of the SnapLogic API.
    e. SnapLogic pipeline details: Supply relevant details for the SnapLogic pipeline to trigger, such as pipeline name or ID.

   f. AWS Connect instance details: Specify the configuration for the AWS Connect instance, including the instance name, phone number, and routing profiles.

6. **Lambda function code:** Input the Lambda function code as a ZIP file or inline code. Ensure it includes:
   a. AWS SDK configuration to interact with AWS services.
   b. SnapLogic API authentication using appropriate credentials (e.g., API token or OAuth).
   c. API call to trigger the specified SnapLogic pipeline based on the received data from AWS Connect.
7. **IAM role permissions:** AWS CloudFormation will create an IAM role with necessary permissions for the Lambda function:
   a. IAM policy to execute the Lambda function.
   b. IAM policy to interact with the SnapLogic API using the provided credentials.
   c. IAM policy for accessing AWS Connect resources.
8. **AWS Connect configuration:** CloudFormation will set up the AWS Connect instance based on the specified configuration, including:
   a. Configuration of phone numbers and routing profiles.
   b. Setting up AWS Connect contact flows and Lambda trigger integrations.
9. **Review and create:** Carefully review the stack details, ensuring all parameters are correctly set. Acknowledge the creation of IAM resources, and click "Create stack" to initiate the deployment.
10. **Stack creation progress:** Monitor the CloudFormation stack creation progress in the AWS Console. Verify the successful provisioning of AWS Connect, Lambda function, IAM roles, and SnapLogic pipeline.
11. **Test the integration:** After stack creation, navigate to the AWS Lambda function's configuration in the AWS Console. Manually trigger the Lambda function to verify it correctly invokes the designated SnapLogic pipeline. Additionally, conduct test calls to your AWS Connect instance to validate proper call routing and functionality.

Cleanup

To remove the integration and delete resources, follow these cleanup steps:

1. **Select stack:** In the AWS CloudFormation service, select the stack created for this integration.
2. **Delete stack:** Right-click on the stack and choose "Delete Stack"
3. **Confirm deletion:** Confirm the deletion when prompted. AWS CloudFormation will automatically handle cleanup and remove all associated resources, including AWS Connect instance, Lambda function, IAM roles, and other created resources.

Conclusion

With this technically detailed integration of AWS Connect, AWS Lambda, and SnapLogic, you have enhanced customer interactions through AWS Connect and automated complex business processes using SnapLogic. Your Lambda function now triggers the designated SnapLogic pipeline seamlessly, providing a robust and efficient solution for your business needs.

Please refer to the documentation for your specific Lambda function code, AWS Connect configuration, and SnapLogic pipeline for further details on their functionalities and interactions.

For any technical issues during deployment or questions, don't hesitate to reach out to the AWS support team or the SnapLogic support team for assistance.

Happy integrating!