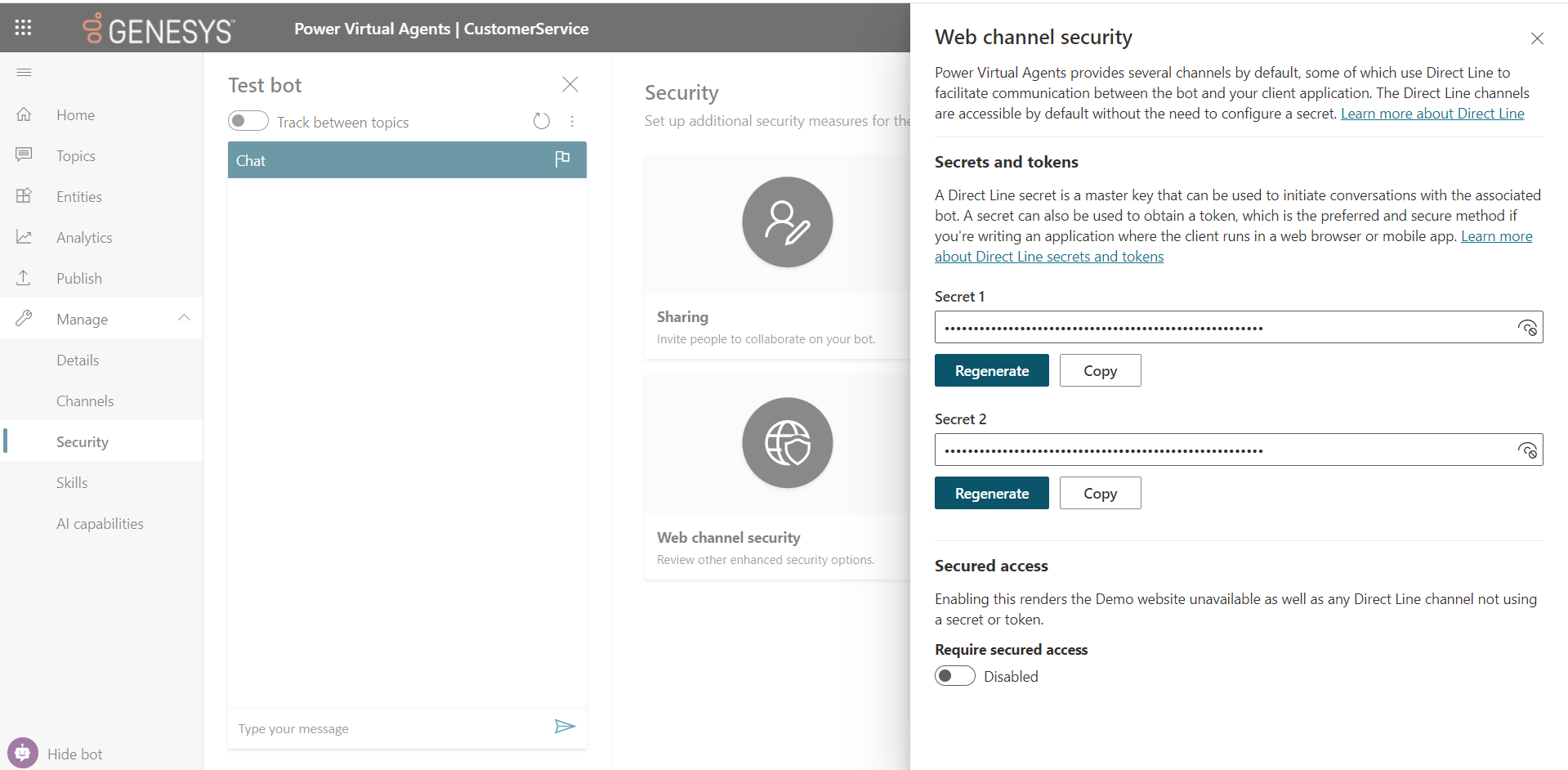
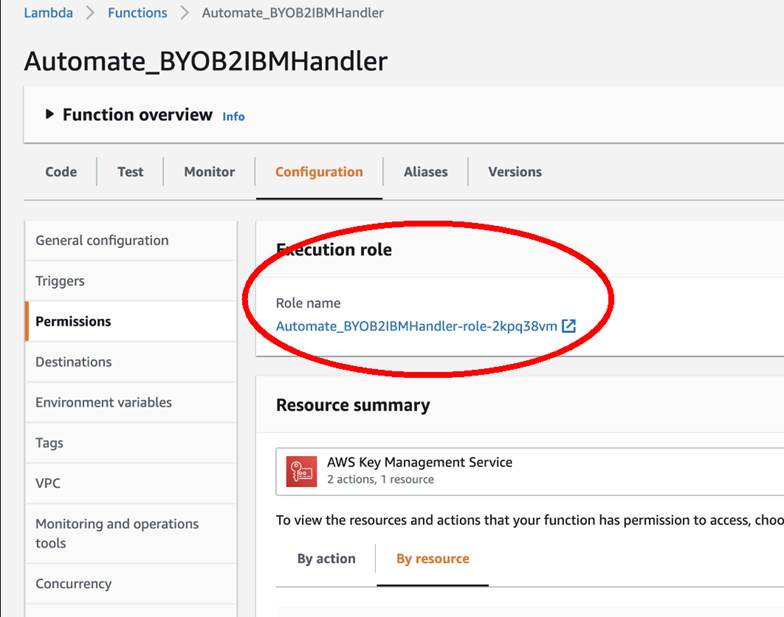
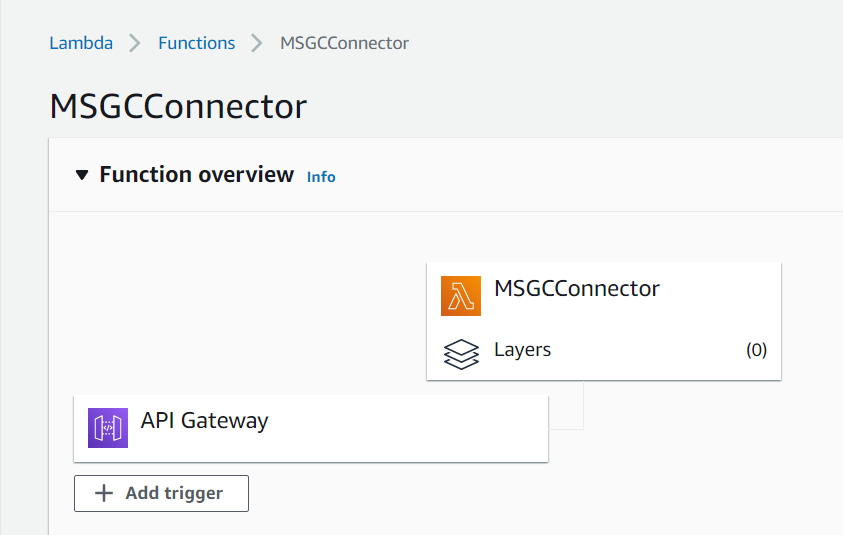
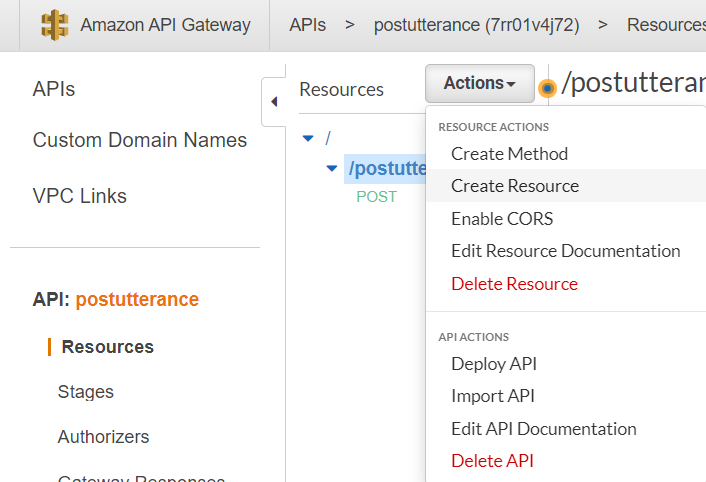
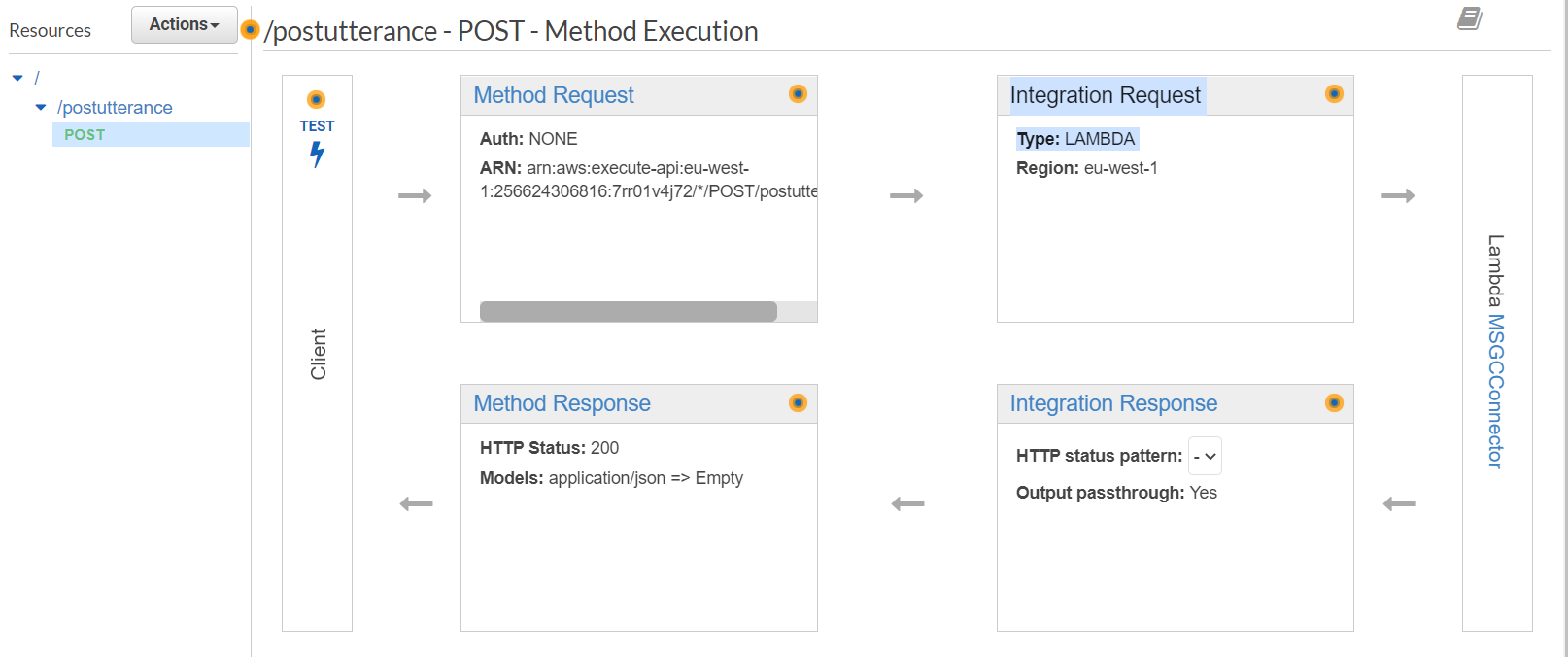
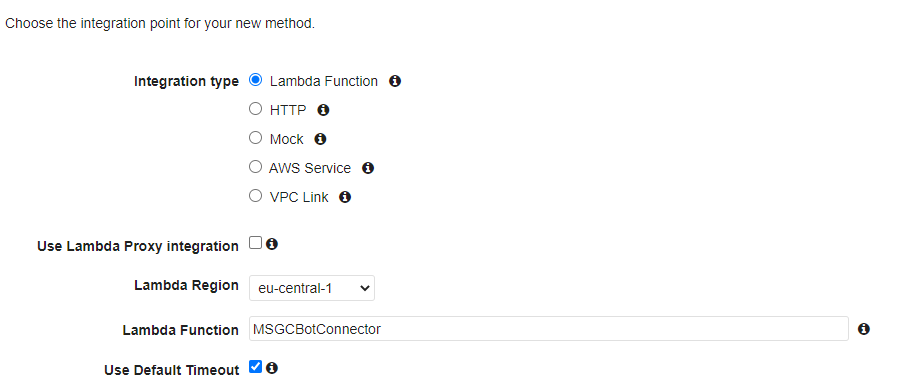
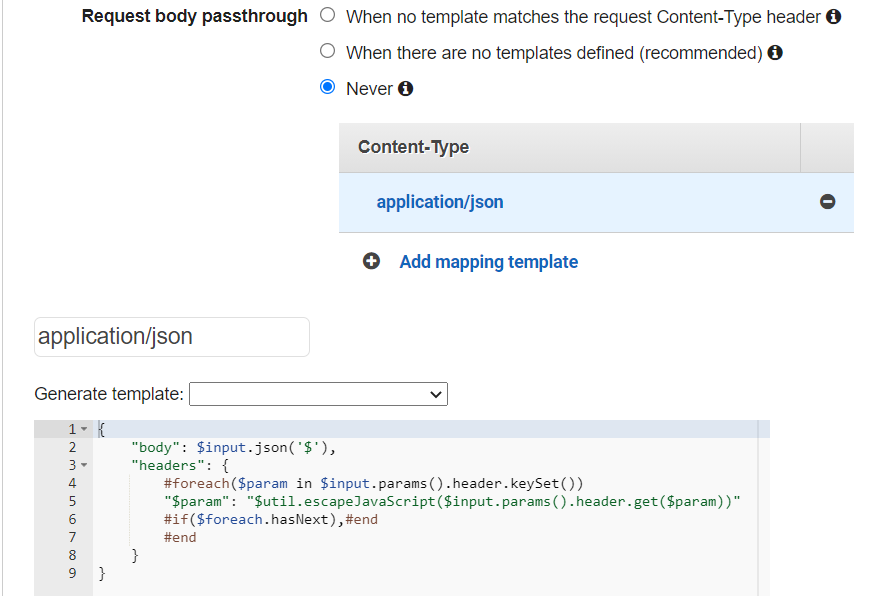
**Requirements:**

* A PowerVA bot: <https://web.powerva.microsoft.com/>
* An AWS account with access to API Gateway, Lambda, CloudWatch and DynamoDB
* This feature is currently in beta, so you will need to contact your Genesys Sales team to access this.

**Getting Started:**

* Create a PowerVA bot
  + Go to <https://web.powerva.microsoft.com/>
  + Create at least one Topic (Intent) that you want to use later
  + Publish your bot at least once
  + Get Token (Secret 1) from PowerVA web site (Click Manage/Security/Web Channel Security and copy "Secret 1"  
    
  + Open the Automate\_BYOB2MS.py file and replace the MS\_BOT\_AUTHORIZATION\_SECRET value with “Bearer {YOUR SECRET 1}” e.g. MS\_BOT\_AUTHORIZATION\_SECRET = "Bearer aNieNrIk.YcKFpSi-nPShwuL9Jhji00-218c2f2P8xDBSa"
  + In the Power VA URL, take the part after bots and keep this for later. For example <https://web.powerva.microsoft.com/environments/Default-785ce69c-90cf-4dc7-a882-eaf334d1d25g/bots/b80cde13-489d-4eab-acad-26893fd9rft1/> url, the id you need is b80cde13-489d-4eab-acad-26893fd9rft1. We will need this for the botList
* AWS configuration
  + Create a new DynamoDB table (name is "automate-byob2ms-sessions" by default unless you change it in bot\_sessions.py line 10 DYNAMODB\_SESSIONS\_TABLE\_NAME)
    - Use botSessionId as the Primary Key
  + Create a new Lambda function (give it a meaningful name like “MSGCConnector”)
    - Select Runtime as Python
    - Add both .py files from the “src” directory. Click on Deploy to ensure the code is copied to AWS.
    - Add the AmazonDynamoDBFullAccess policy to the role used by the lambda function (you can also restrict permissions to the “automate-byob2ms-sessions” table only)



* + Add an API gateway to your lambda
    - Click the +Add Trigger button  
      
    - Select API Gateway, then Create and API from the list. Choose REST API and choose a security method (Open is fine for testing). Click Add.
    - Once created, select the API Gateway from the list to edit it.
    - Select the root of the Resources and click on the Actions button. Select Create Resource and call it postutterance. Click Create Resource.  
      
    - Now, from Actions, Select Create Method. From the drop down that appears, select POST. Click on the new POST option  
      
    - Click on Integration Request
    - Select Lambda Function  
      
    - Scroll down to Mapping Templates. Click Add mapping template. Enter application/json. Click the tick and scroll down to see a new box  
      
    - Paste this code into the code box as shown above

{

"body": $input.json('$'),

"headers": {

#foreach($param in $input.params().header.keySet())

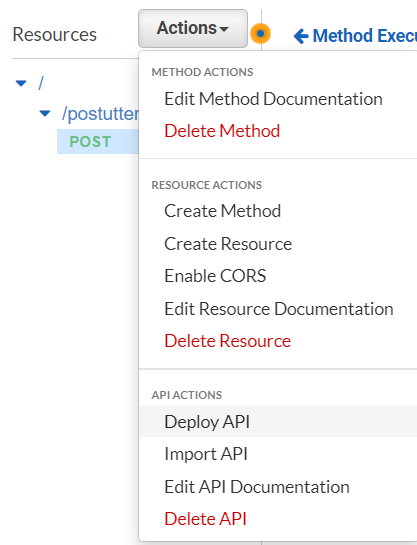
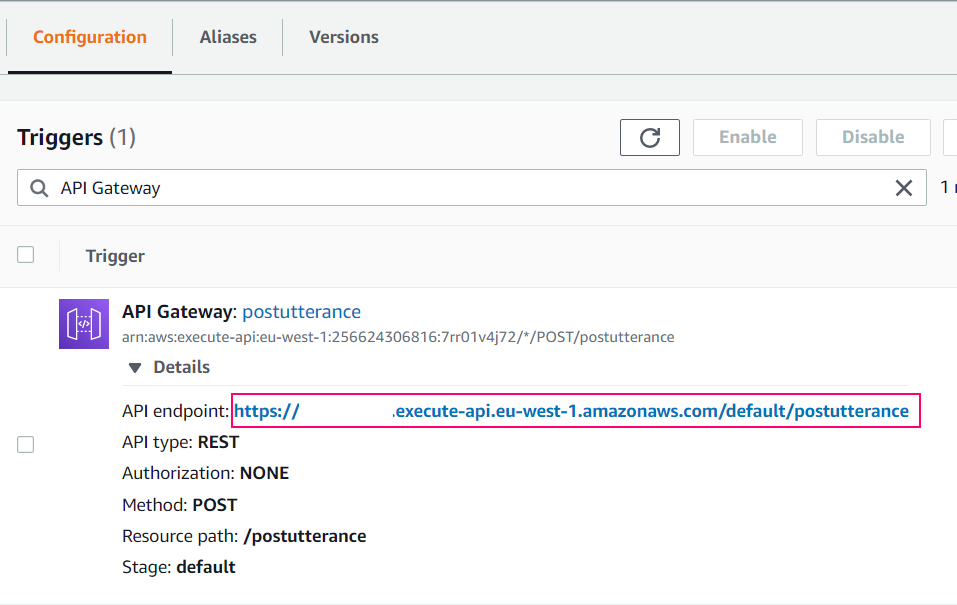
"$param": "$util.escapeJavaScript($input.params().header.get($param))"

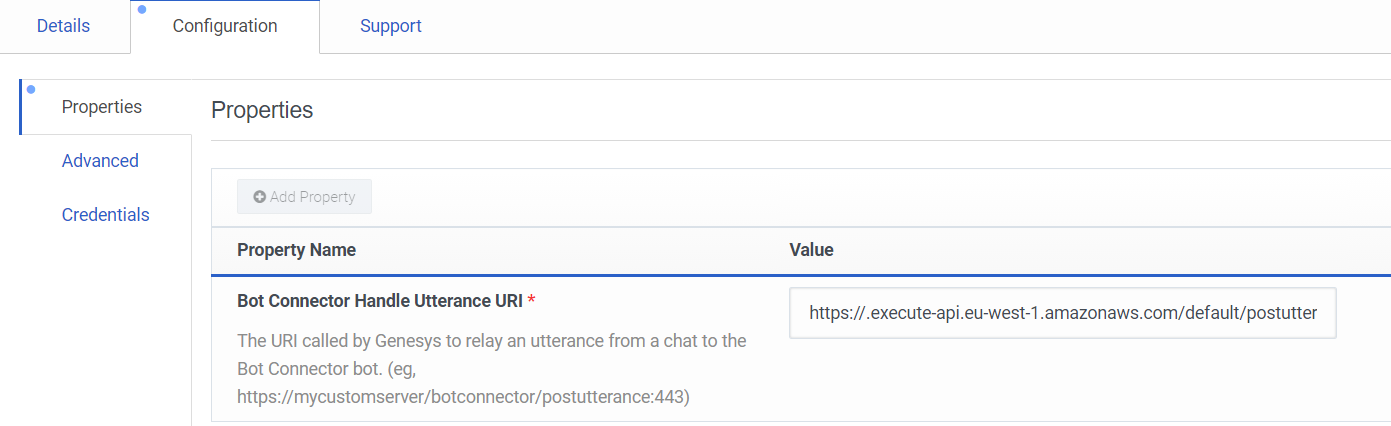
#if($foreach.hasNext),#end

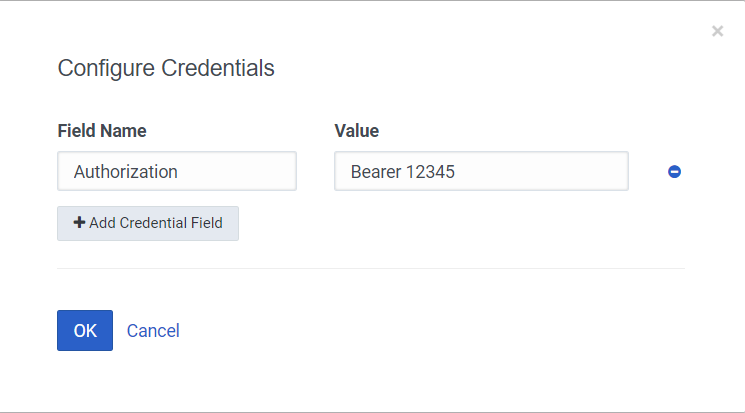
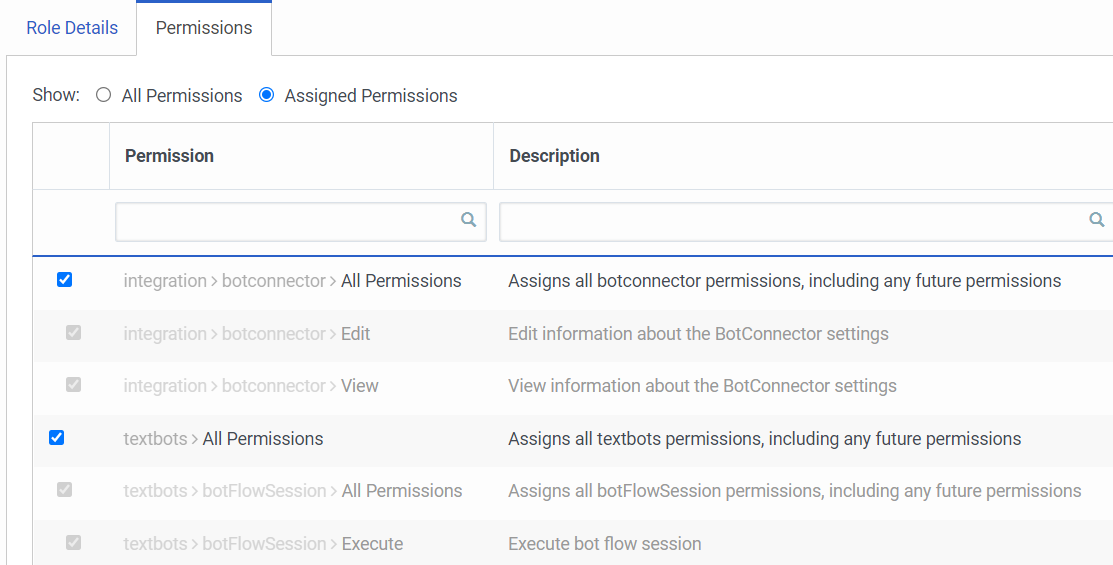
#end

}

}

* + - Request Body Passthrough should be set to "Never"
    - Click Save.
    - Click the Actions button again and select Deploy API  
      
    - You can now close the tab and go back to the Lambda function.
    - Under Configuration, open the Triggers and the API Gateway. Click on Details and record the API Endpoint url for later  
      
* Genesys Cloud
  + Create a new “Genesys Bot Connector” integration
    - Go to Admin and select Integrations.
    - Click on the +Integrations button to add a new one. Search for Genesys Bot Connector and click Install
    - Name the integration and click on Configuration
    - In the field Bot Connector Handle Utterance URI paste in the API endpoint url saved from the AWS API Gateway config e.g. 



* + - Under the Credentials tab, we must add something. You can use the bearer we created earlier, or any authorization headers you want to send.
    - Click Configure. Click Add Credential Field. You can enter Authorization for the Field Name and “Bearer {YOUR SECRET 1}” for the Value. The sample does not use this, but for other integrations it can help  
      
    - Save and activate the integration
    - Get the ID for the integration from the URL as we will need this later.
  + Create a role (or edit an existing one) and add the permissions to it:
    - Integration -> botconnector -> All Permissions
    - Textbots -> All Permissions
    - Assign this role to your account in Genesys Cloud.
  + Bot configuration
    - Get the PVA bot id from the Microsoft PowerVA (after “Default”) URL: [https://web.powerva.microsoft.com/environments/Default-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/bots/{id}/](https://web.powerva.microsoft.com/environments/Default-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/bots/%7bid%7d/) as mentioned earlier.
    - Get your access token from Genesys Cloud to authenticate against the API.
    - Call the following Genesys Cloud API (check if topics and entities are correct first):

*PUT* [*https://api.mypurecloud.ie/api/v2/integrations/botconnector/{YOUR*](https://api.mypurecloud.ie/api/v2/integrations/botconnector/%7bYOUR) *GENESYS BOT CONNECTOR INTEGRATION ID}/bots*

*Authorization: bearer {YOUR GC ACCESS TOKEN}*

*Content-Type: application/json*

*{*

*"chatBots": [*

*{*

*"id": "{PVA BOT ID}",*

*"name": "{DISPLAY NAME}",*

*"versions": [*

*{*

*"version": "Delta",*

*"supportedLanguages": [*

*"en-us",*

*"es"*

*],*

*"intents": [*

*{*

*"name": "Check Stock",*

*"slots": {*

*"Slot1 ": {*

*"name": "Slot1",*

*"type": "String"*

*}*

*}*

*},*

*{*

*"name": "Escalate",*

*"slots": {*

*"Slot2": {*

*"name": "Slot2",*

*"type": "String"*

*}*

*}*

*}*

*]*

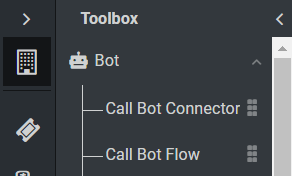
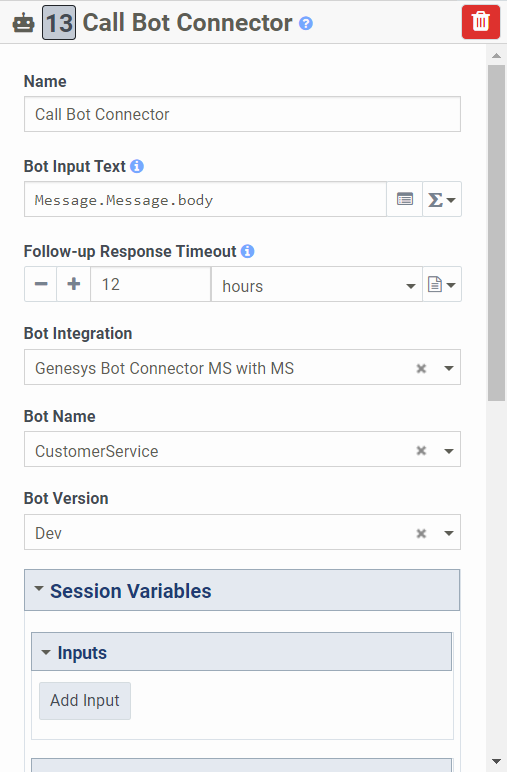
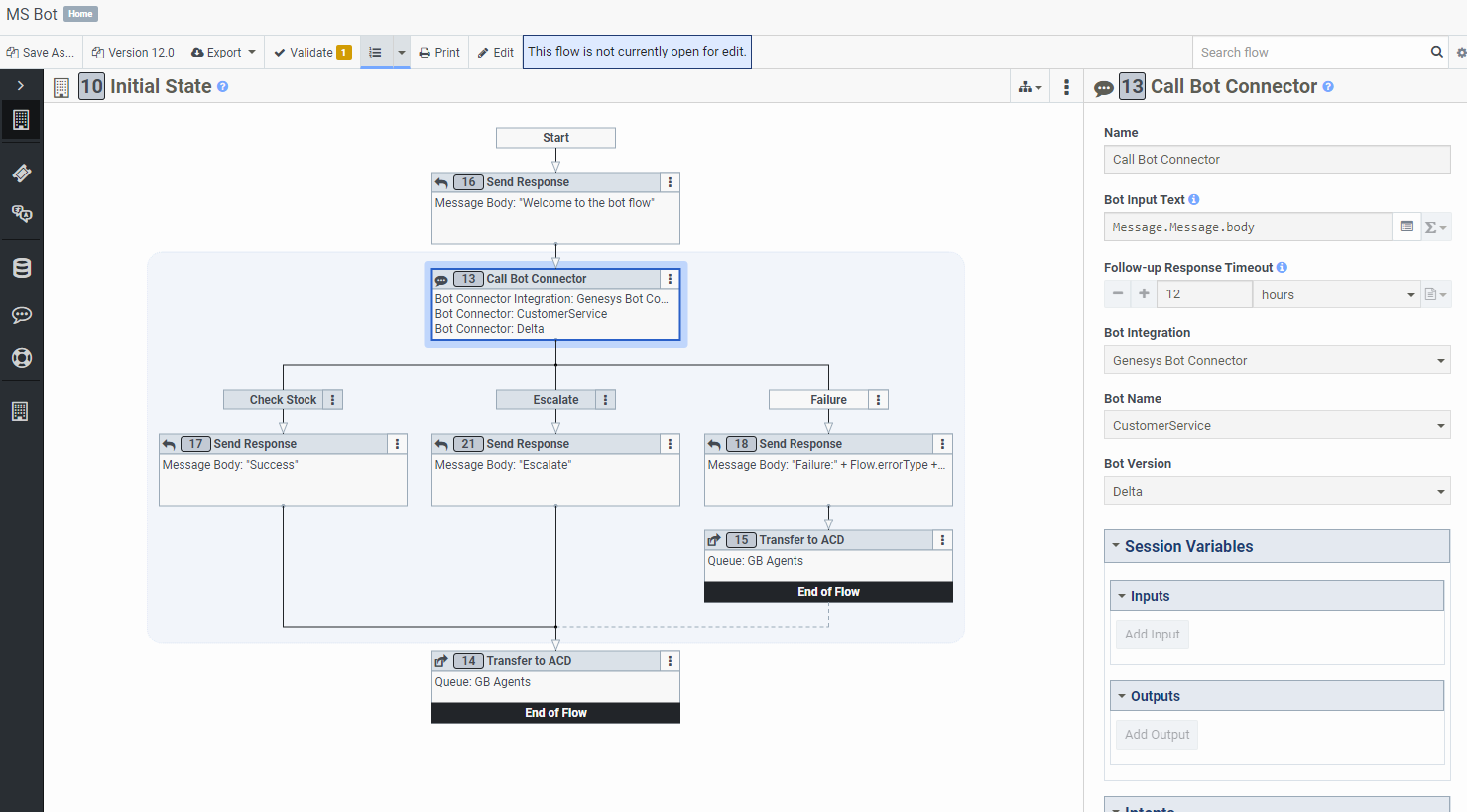
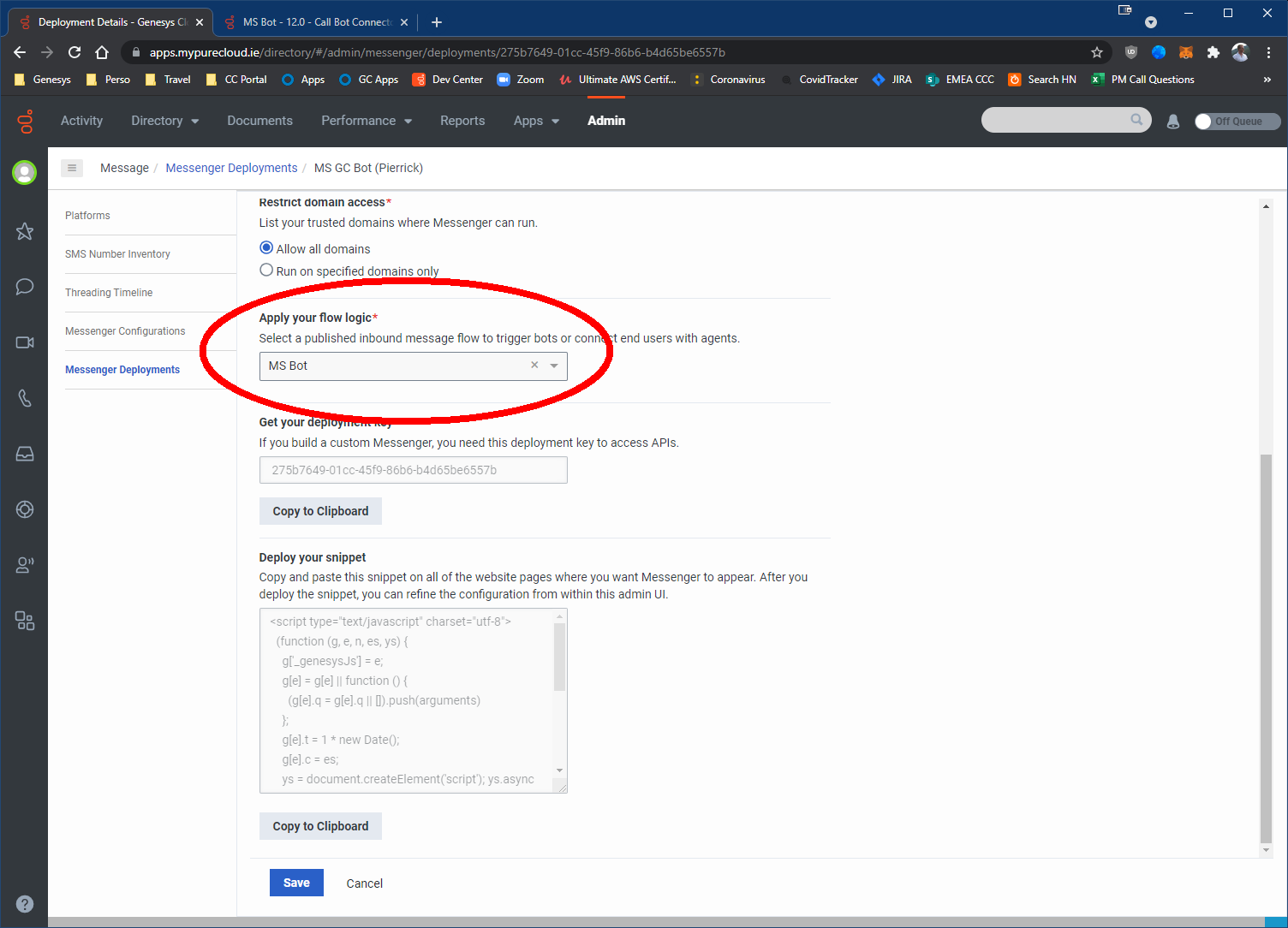
*}*

*]*

*}*

*]*

*}*

* + - You should receive a 204 No Content if all works well
    - This declares your bot config to Genesys Cloud so it can be found in Architect.
  + Create an Architect flow
    - In Architect, create a new Inbound Message flow (no other type is supported).
    - From the Tools, drag on a Call Bot Connector block from the Bot menu  
      
    - In the options of the Call Bot Connector block, you should now be able to select your Bot Integration, Bot Name and Bot Version as specified in the API call we made earlier  
      
    - Create the rest of the flow as you would for other bots  
      
* Web Messaging
  + Create a configuration in any way you like
  + Create a deployment
    - Send messages to your message flow  
      
    - Copy the snippet from the Deployment and deploy to a web page to test  
      