PUBLIC CLOUD AI/ML

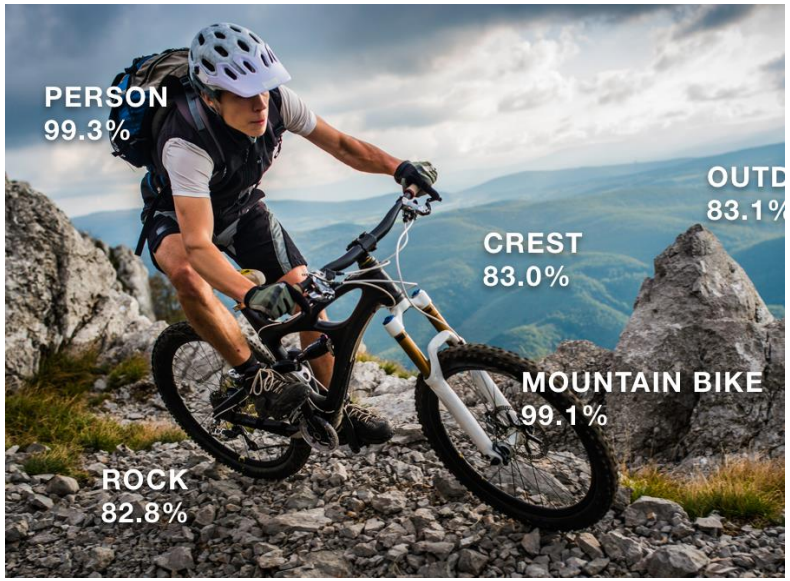
BACKGROUND

- All of the public cloud service providers are investing heavily in AI/ML services
- These services broadly take two forms
  1. Services to ease the effort required to select, train and deploy custom models
  2. Pre-packaged OR pre-trained ML models that can be accessed through REST APIs

AWS EXAMPLE: SAGEMAKER

- **Amazon Sagemaker** helps data scientists to prepare, build, train and deploy high-quality machine learning models quickly
- An ***Amazon SageMaker notebook instance*** is a machine learning (ML) compute instance running the Jupyter Notebook App. SageMaker manages creating the instance and related resources.
- **Amazon SageMaker Studio** provides a web-based visual interface (IDE) to easily build your models in the Python language.

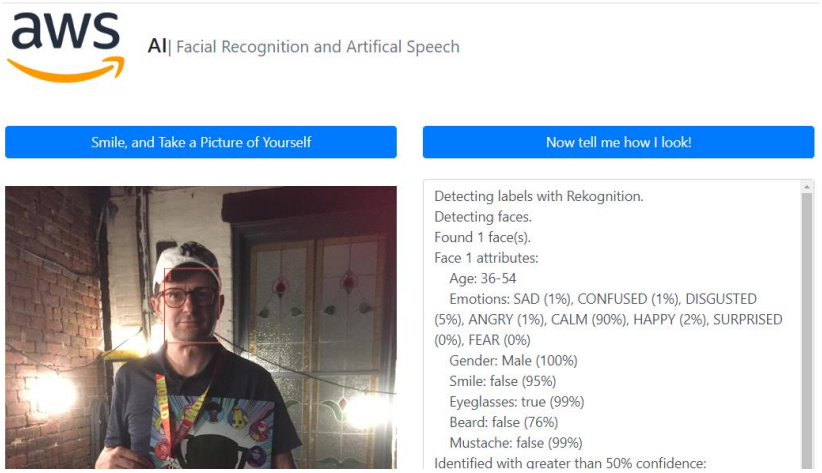
AWS Rekognition: ML Image & Video Analysis



- AWS have already done all the machine learning, so you can focus on the artificial intelligence
- The Amazon service is called Rekognition
  - https://aws.amazon.com/rekognition/
- Not only can it do faces, but it can do
  - Scene detection
  - Image appropriateness detection
  - Path detection
  - Text detection

AWS Rekognition: ML Image & Video Analysis

- Now you can try some image recognition
- Using your phone, visit
  - http://neueda.conygre.com/recognition/index.html



AWS Comprehend: ML Natural Language Processing

- Store datasets in S3 or upload through API
- Extract key phrases, extract entities, sentiment analysis
- Can be customized for specific datasets

Key phrases	Confidence
Zhang Wei	0.95
John	0.99+
Your AnyCompany Financial Services	0.98
LLC credit card account 1111-0000-1111-0008	0.87
a minimum payment	0.99+
$24.53	0.99+
July 31st	0.99+

AUTOMATED CHATBOTS

WHAT IS LEX?

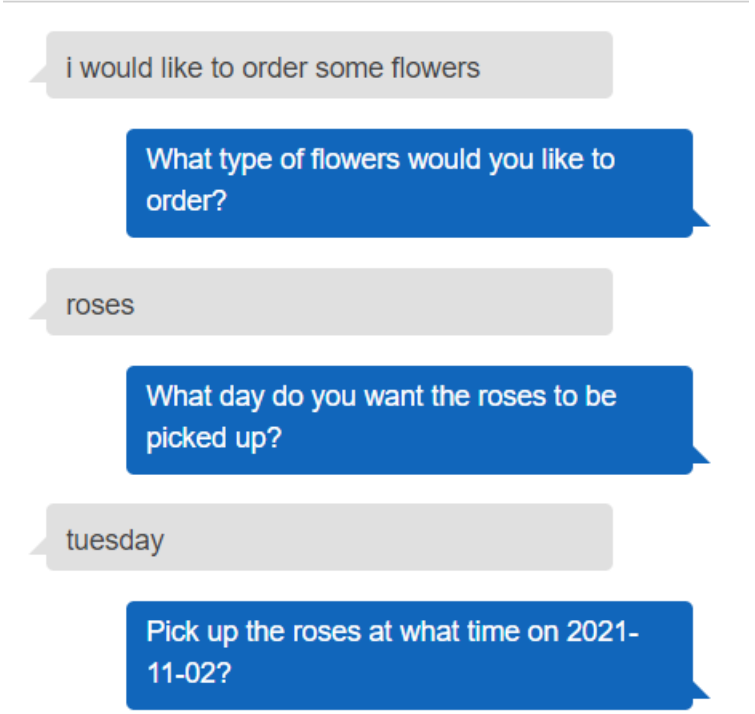
- Amazon describes Lex as a 'service for building conversational interfaces into applications using voice and text'.
- Essentially Lex takes all the **Natural Language Understanding** and **Automatic Speech Recognition** from the Amazon Echo and makes it available to developers in order to build a more engaging **UX**.

WHY CHOOSE LEX

- Full AWS integration
  - Lambda, Cloudwatch, Logging
- All the Machine Learning has been done for you
- Seamless integration into Facebook, Slack, Kik and Twilio SMS

EXAMPLE

- To demonstrate how Lex works you will see how to create a simple chatbot to order flowers



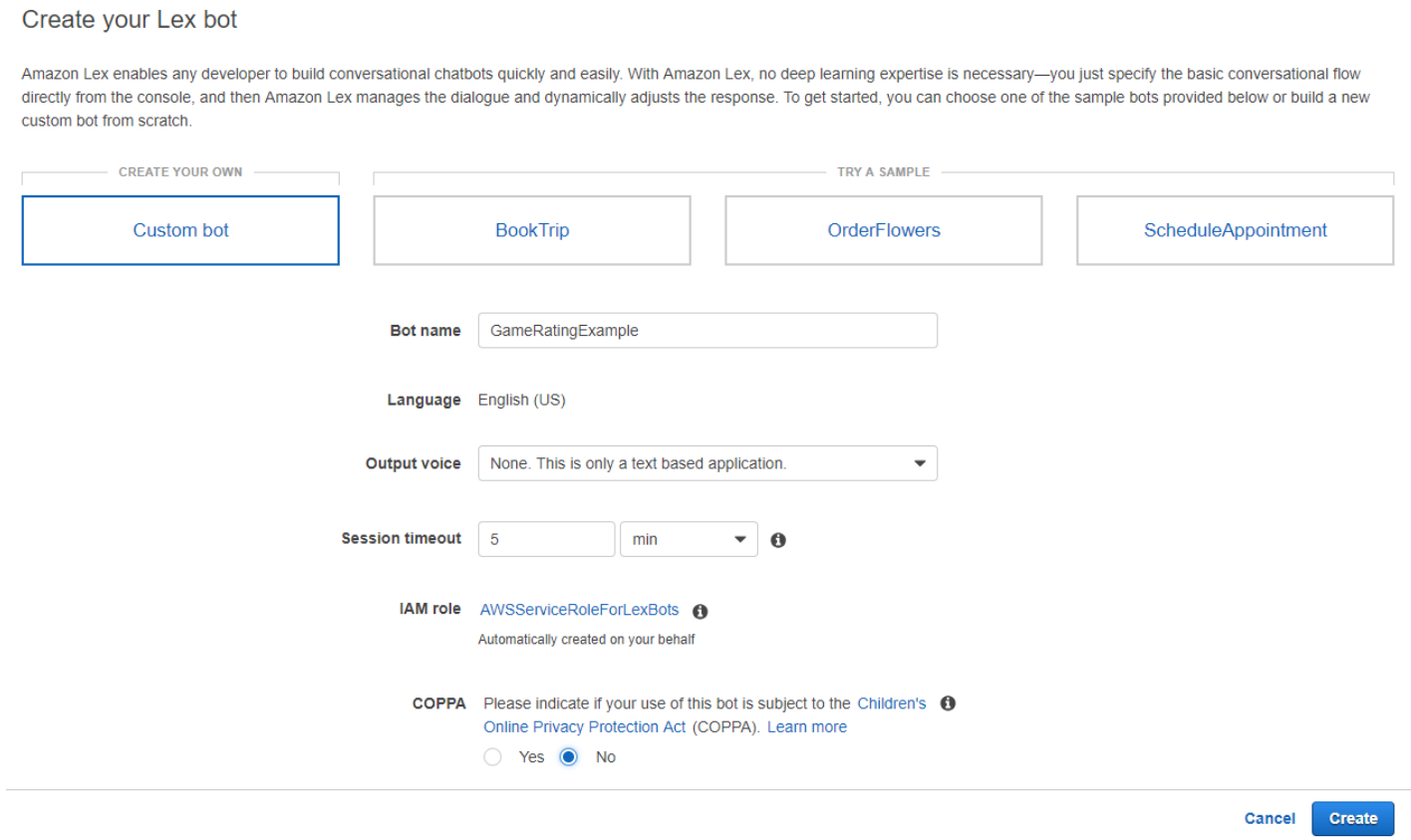
BUILDING A CHAT BOT TO ORDER FLOWERS

- Need to establish how the conversation will flow

Customer	Bot Response
Hello, I would like to order some flowers	Okay, what type of flowers would you like?
Roses	What day would you like the roses to be picked up?
Tuesday	Pick up the roses at what time on 2021-11-02?
10am	Okay, your roses will be ready for pickup by 14:00 on 2021-11-02. Does this sound okay?
Yes	Thanks for ordering roses with us.

CREATING THE BOT

- Bots can be created through the AWS Web Console



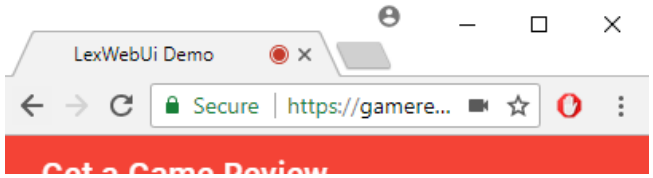
SAMPLE BOTS

- There are some sample bots you can use to get started quickly
  - Order Flowers
  - Book a trip
  - Schedule an appointment
- Or you can create your own from scratch

VOICE CONTROL

- Since Lex is the engine behind the Amazon Echo with Alexa you can optionally select that your chatbot can also work with voice as well as text

- If you do enable this, the red dot appears in the browser bar if the user enables it



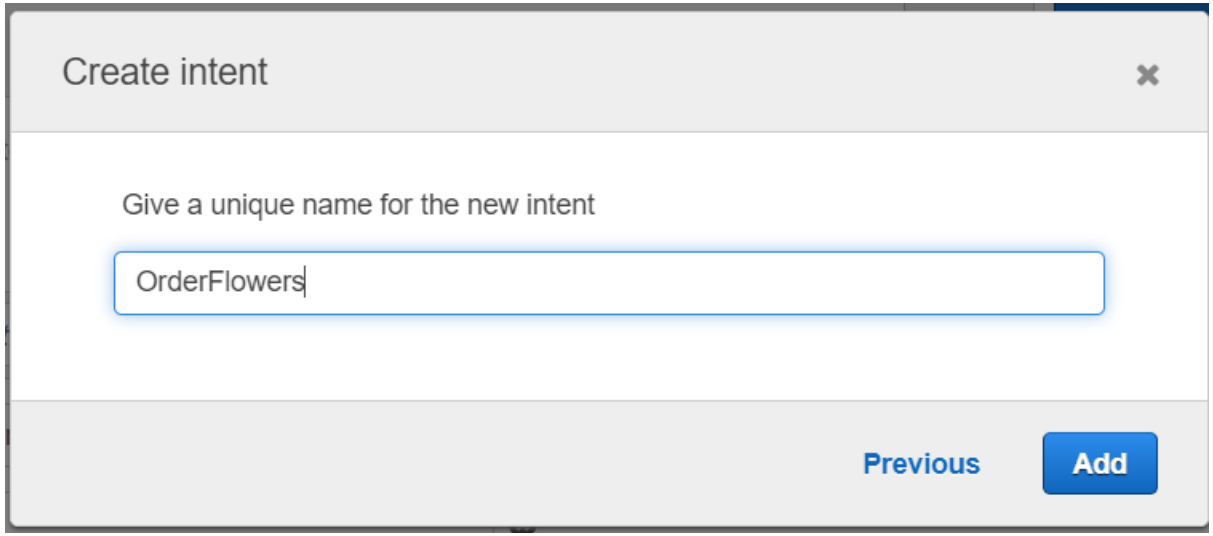
INTENTS AND SLOTS

- You now need to set up **Intents** and **Slots**
  - Intent – what the user wants to do
  - Slot – specific pieces of information provided by the user

Customer	Bot Response	Intents and Slots
Hello, I would like to order some flowers	Okay, what type of flowers would you like to order?	Intent: OrderFlowers established
Roses	What day do you want the roses to be picked up?	Slot: FlowerType = Roses
Tuesday	Pick up the roses at what time on 2021-11-02?	Slot: PickupDate = 2021-11-02
2pm	Okay, your roses will be ready for pickup by 14:00 on 2021-11-02. Does this sound okay?	Slot: PickupTime = 2pm
	Thanks for ordering some roses	Intent is now fulfilled

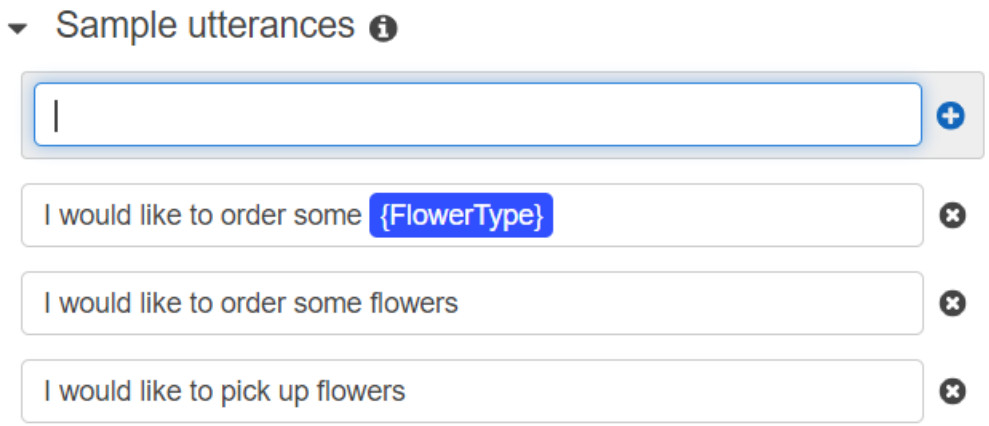
CREATING AN INTENT

- To create an Intent, click **Create Intent**
- You can have multiple intents
- In the example we require one intent and our **Intent** is to **Order Flowers.**



INVOKING THE INTENT

- We need to tell Lex what kinds of things the customer might enter in order to invoke our Intent
- Lex calls these phrases **utterances**



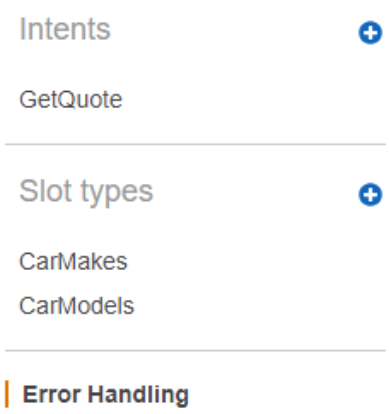
- The words in curly brackets are slots – discussed next

GETTING INFORMATION FROM THE USER

- Almost always, in order to fulfil our intent, we will need to information from the user
  - In our case, we want to know the **FlowerType** that they want to order, and also the **PickupDate** and **PickupTime**
- Each piece of information is referred to as a **Slot** by Lex

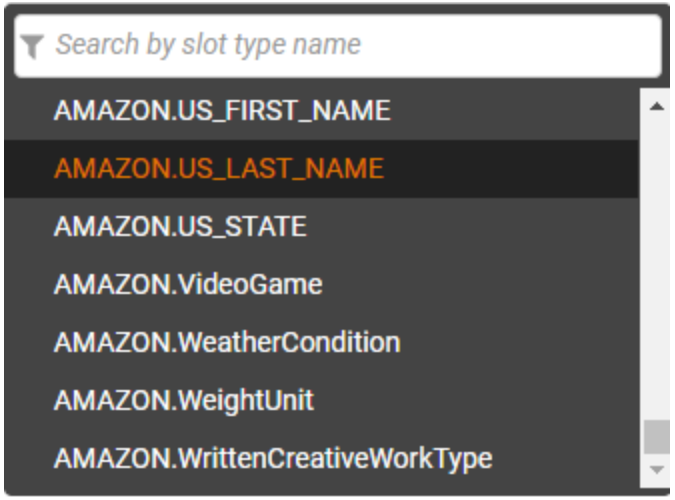
ERROR HANDLING

- Sometimes a user will enter something that Lex doesn't understand.
- In this scenario, Lex comes with features to prompt a user to re-enter some information (ideally in a way Lex **will** understand)



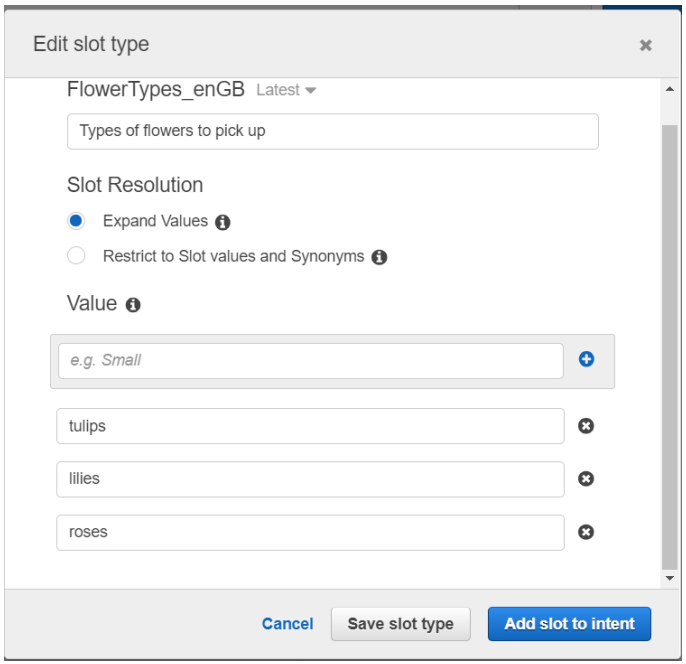
SLOTS AND SLOT TYPES

- User data is placed into **Slots**
- Slots always have a Slot Type
- Lex defines some commonly used types
  - There are lots of them!



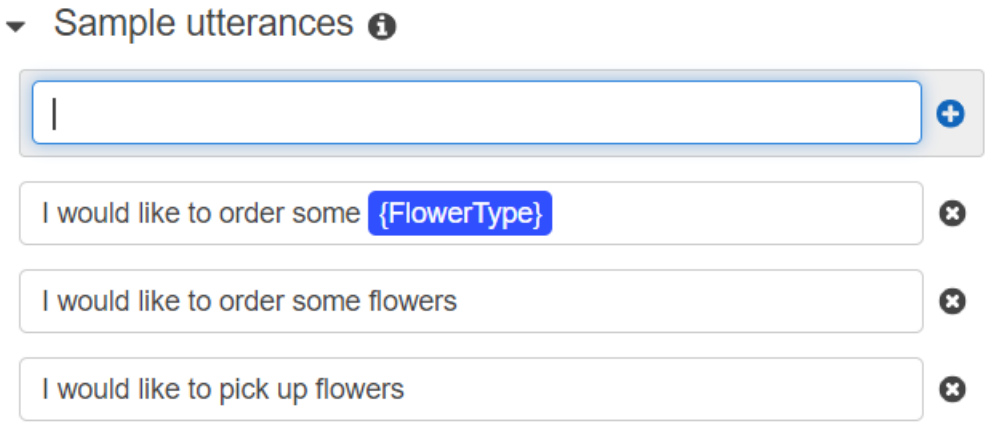
CUSTOM SLOT TYPES

- For the **FlowerType** slot we need to define a custom slot type which lists all the possible platforms that our bot is able to get game ratings for
- In this case we want to make sure that the slot type is **not** set to expandable, as we only want Lex to populate the Platform slot with valid flower types
- Note that slot names are case sensitive



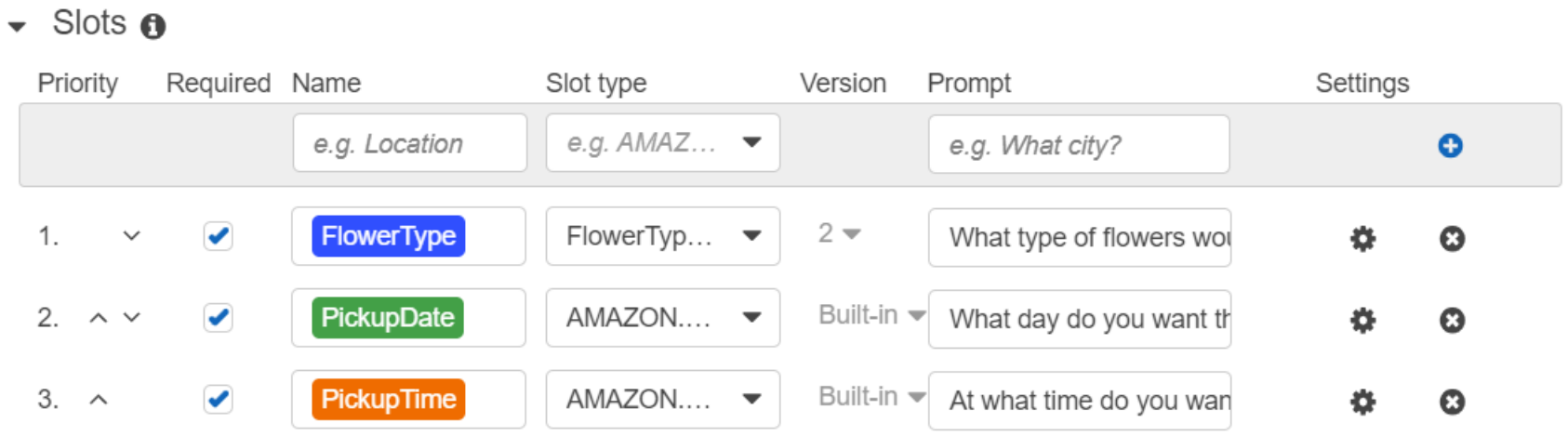
CUSTOM SLOT TYPES

- Slots also have a Prompt which is the phrase the bot will use to elicit the information from the customer



- Always place your slot names in curly brackets when adding the prompts

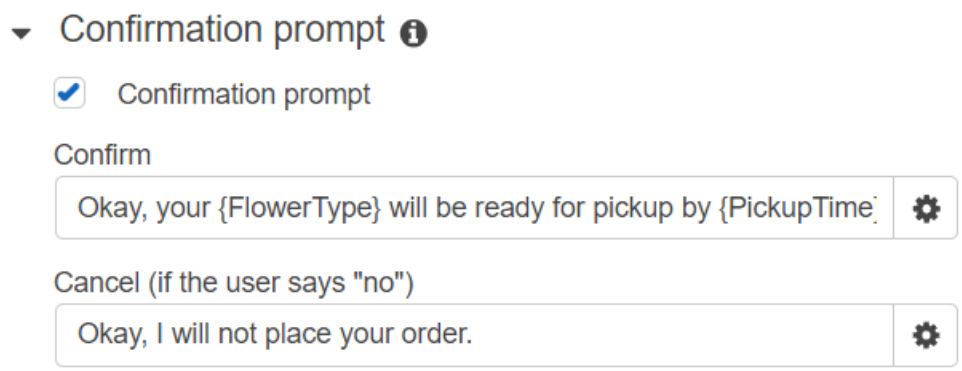
CREATING THE SLOTS



- Notice that now the slots have been defined the References to them in the **Utterances** section have been {**color-coded**}

CONFIRMATION

- An important 'good practice' of Bot Design is to make sure that the user knows what is going on
- Repetition, Confirmation and Repetition are some excellent ways to incorporate this.
- Lex provides a useful **Confirmation Prompt** section to easily ensure the user is happy with the information they have given



FULFILLING THE INTENT

- In Lex an intent can either be **Fulfilled** or **Failed**
- In order to determine this, we can either
  - Use the built in 'return parameters to client' function. In which case the intent will be **fulfilled** once Lex has populated all the required **Slots**

    Or

  - We can utilise Amazon **Lambda** is order to do some logic on the filled **Slots** and determine for ourselves whether or not the intent has *actually* been fulfilled

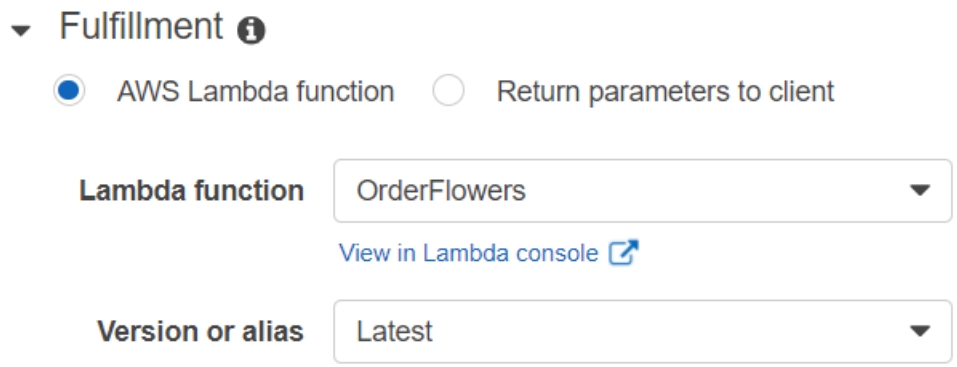
    [Similarly to above, the **lambda** will be called once all the required **slots** are filled.]

INTRODUCING LAMBDA

- Lambda Functions allow you to create **serverless architectures**
- In the serverless paradigm, you would create a function that could be invoked when someone uses your chatbot
- The function is all that you deploy!
  - No servers
  - No maintenance
  - No ongoing server costs
  - No hassle!

ADDING A LAMBDA TO LEX

- Incorporating the lambda into our lex bot is straightforward
  - In the fulfilment tab we just tell Lex to call our lambda function



SUMMARY

- Public cloud providers such as AWS provide many AI/ML services.

- Lex is an AWS Environment for **Natural Language Understanding**
  - **Intents** describe what your bot will do
  - **Slots** describe the different pieces of information your bot will need to perform the **intent**
  - Slots can be explicitly **confirmed** to improve **UX**
  - **Lambda** is implemented to create advanced intents that your bot can **Fulfil**