

Building Intelligent Chatbots with Amazon Lex & Amazon Polly

Pradyumna Dash
Solutions Architect
Amazon Web Services





Serverless Computing

**80% of IT budgets go towards just
keeping the lights on.**

Gartner



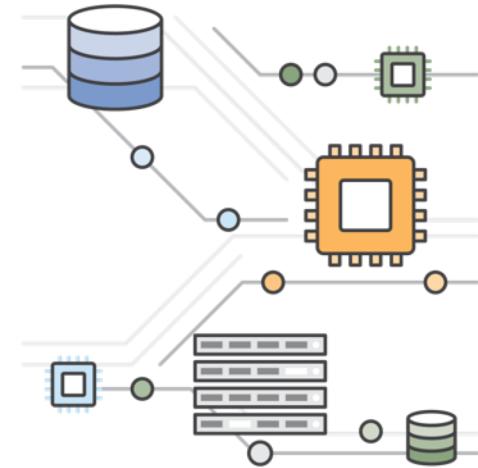
Physical Hardware

- Procurement
- Power
- Cooling
- Asset Tracking
- Remote hands
- IP transit
- Colocation
- Capacity planning
- Hardware refreshes
- Storage
- Deprecation
- Physical security
- Networking equipment
- Cabling

Serverless Computing

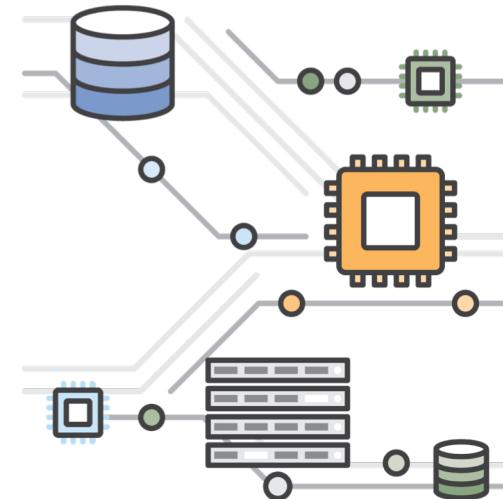
Build and run applications and services without thinking of servers

- Fully managed
- Developer productivity
- Continuous scaling

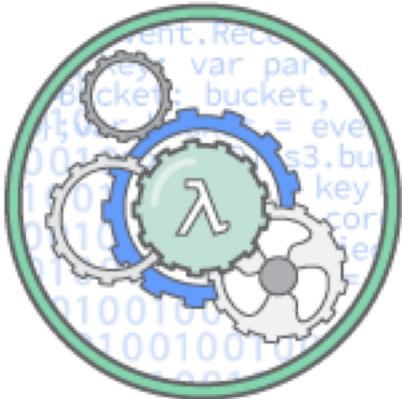


AWS Lambda: Serverless computing

- Run code without servers. Pay only for the compute time that you consume.
- Triggered by events or called from APIs:
 - PUT to an Amazon S3 bucket
 - Updates to Amazon DynamoDB table
 - Call to an Amazon API Gateway endpoint
 - Mobile app backend call
 - And many more...
- Makes it easy to:
 - Perform real-time data processing
 - Build scalable backend services
 - Glue and choreograph systems

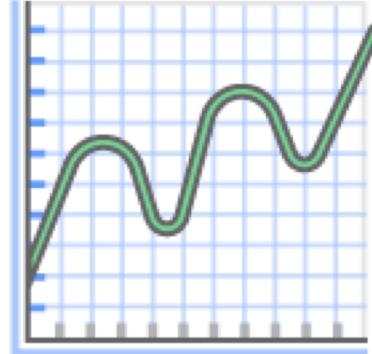


Benefits of AWS Lambda



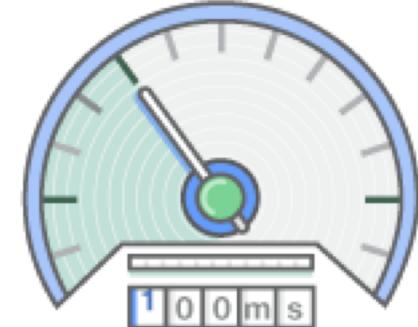
No servers to manage

Lambda allows you to run application logic without provisioning servers or worrying about the health or security of underlying resources



Continuous scaling

Lambda scales infrastructure beneath your application logic; just send requests and events and Lambda automatically scales to accommodate it



Don't pay for idle resources

With Lambda, you're billed in 100ms increments of execution time and number of requests and you're never charged for anything when your code isn't running



Amazon DynamoDB

- Fast and flexible NoSQL database service for any scale
- GetItem(primaryKey)
- PutItem(item)



```
const doc = require('dynamodb-doc');
const dynamo = new doc.DynamoDB();

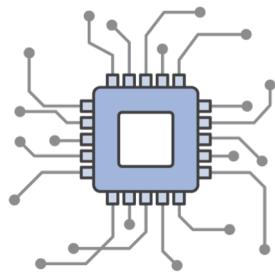
exports.handler = (event, context, callback) => {

  const id = event.payload.id;
  dynamo.getItem(id, callback);

};
```

Amazon API Gateway

- Create, publish, maintain, monitor, and secure APIs at any scale



Create a unified API frontend for multiple backend microservices



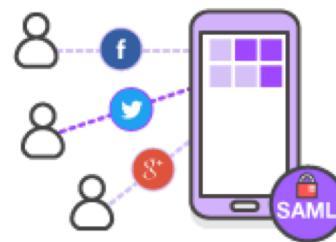
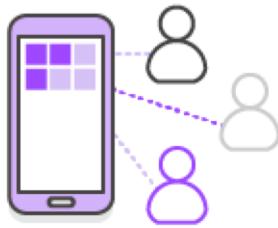
DDoS and throttling to protect your backend microservices



Authenticate and authorize requests to your backend microservices

Amazon Cognito

- Add user sign-up, sign-in, and data synchronization to your apps



Amazon Simple Storage Service

- Durable, massively scalable object storage
- Designed for 99.999999999% durability and 99.99% availability
- Stores trillions of objects and regularly handles millions of requests per second
- Virtually infinite storage without provisioning capacity





ML at Amazon

The Amazon ML Stack

AI Services



AMAZON RECOGNITION

AMAZON
TREXTRACT



AMAZON
POLLY

AMAZON
TRANSCRIBE



AMAZON
COMPREHEND

AMAZON
TRANSLATE



AMAZON
LEX



AMAZON
PERSONALIZE

AMAZON
FORECAST

VISION

SPEECH

LANGUAGES

CHATBOTS

VERTICAL

ML Services



AMAZON
SAGEMAKER



AMAZON
SAGEMAKER RL



AMAZON
SAGEMAKER
GROUND TRUTH



AWS
DEEPRACER



AWS
DEEPLENS



AWS
MARKETPLACE
FOR ML



AMAZON
SAGEMAKER NEO

ML
Frameworks +
Infrastructure

mxnet PYTORCH TensorFlow

FRAMEWORKS

K KERAS GLUON

INTERFACES

P3 P3dn

C5 C5n

AMAZON ELASTIC INFERENCE

AWS INFERENTIA

AWS GREENGRASS

INFRASTRUCTURE





Conversational Offerings

Conversational offerings

Amazon Alexa

A free cloud-based service that provides:

- Natural voice control and built-in intelligence for news, weather, music, smart home, web search and more
- Access to more than 30K+ custom developed skills

Use Alexa to:

- Differentiate your product or reach millions of Alexa-enabled homes
- Add Alexa to your device using the Alexa Voice Service (AVS)
- Publish a skill using the Alexa Skills Kit (ASK)

Alexa for Business

A pay-as-you-go service that provides:

- Use of Alexa as an intelligent assistant at work
- Tools and controls that administrators need to deploy and manage shared Alexa devices, skills and users at scale

Use Alexa for Business to:

- Make skills available to shared devices and enrolled users without having to publish them to the Alexa Skills store
- Build context aware skills for use on shared devices through a public API

Amazon Lex

A pay-as-you-go service that provides:

- A completely managed service for building conversational interactions into any application using voice and text
- Interoperability and integration with AWS services
- Uses the same conversational engine (NLU + ASR) that powers Alexa

Use Lex to:

- Build a text or speech chatbot that works on multiple mobile devices and multiple chat services (e.g., Facebook Messenger, Slack, Kik, Twilio)
- Create highly customizable conversational interfaces and chatbots

Additional: Amazon Polly, Amazon Comprehend, Amazon Translate, Amazon Transcribe



AWS Language Services

Amazon Lex

Service for building conversational interfaces into any application using voice and text

GA April 2017

Amazon Comprehend

Natural language processing (NLP) service that uses machine learning to find insights and relationships in text

GA Nov 2017

Amazon Translate

Neural machine translation service that delivers fast, high-quality, and affordable language translation

GA Apr 2018

Amazon Transcribe

An automatic speech recognition (ASR) service that makes it easy for developers to add speech to text capability to their applications

GA Apr 2018

Amazon Polly

A text to Speech service that converts text to lifelike speech

GA Nov 2016



Amazon Polly

Amazon Polly

Use Cases

Content Creation

Mobile & Desktop Applications

Internet of Things (IoT)

Education & E-Learning

Telephony

Game Development

Key Features

54 Voices across 30 languages

Lip-Syncing & Text Highlighting

Fine-grained Voice Control

Custom Vocabularies

Speech Quality

- **Natural sounding speech**

A subjective measure of how close TTS output is to human speech.



- **Accurate text processing**

Ability of the system to interpret common text formats such as abbreviations, numerical sequences, homographs etc.



- *Today in Las Vegas, NV it's 90°F.*



- *"We live for the music", live from the Madison Square Garden.*

- **Highly intelligible**

A measure of how comprehensible speech is.



"Peter Piper picked a peck of pickled peppers."

Time to play with Polly

- Try some plain text
 - Hi Kartik – my name is Salli.
- Try some SSML
 - <speak>Hi <phoneme alphabet="ipa" ph="kərθɪk">Kartik</phoneme>! My name is Salli.</speak>
- Try at least 2 voices



Amazon Lex

Developer challenges

Security

Speech
Recognition

Disparate Systems

Authentication

Language
Understanding

Messaging
platforms

Availability

Scale

Business Logic

Testing

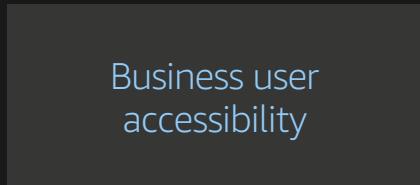
Mobile

Conversational interfaces need to combine a large number of sophisticated algorithms and technologies



Amazon Lex: Complete Solution





Native dialog management



Pre-defined resources

Test

Test UI



Alias and versioning

Deploy

One click deployment to Facebook,
Slack, Twilio, Kik

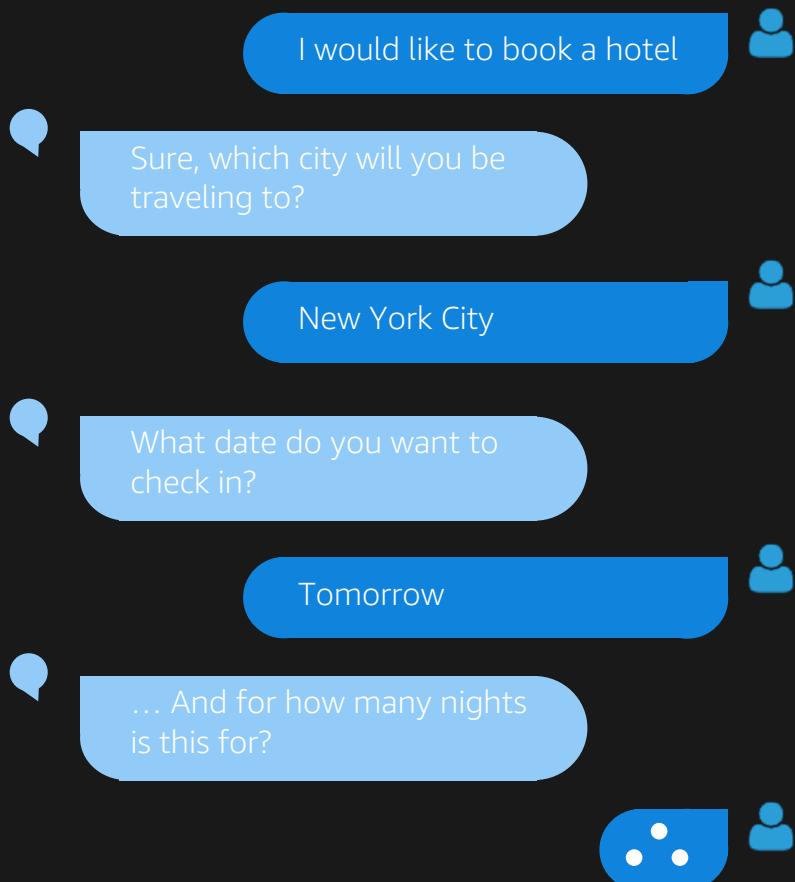


Missed utterances

Monitor



Native dialog management



Simple Declarative Model

Slots

City
Check-In Date
Check-Out Date

Prompts

Which city will you be travelling to?
What date do you want to check in?
How many nights is this for?

Build Multi-turn Conversations

Easy Setup in Console

Pre-defined resources

Built-in Slot Types

- Ready to use slot types that are already trained with sample values
- Do not need to spend the time enumerating sample values for these slot types

AMAZON.DATE

AMAZON.TIME

AMAZON.NUMBER

.....

Benefit from continuously expanding pre-defined resources

Test UI

Comprehensive
testing from
the Console

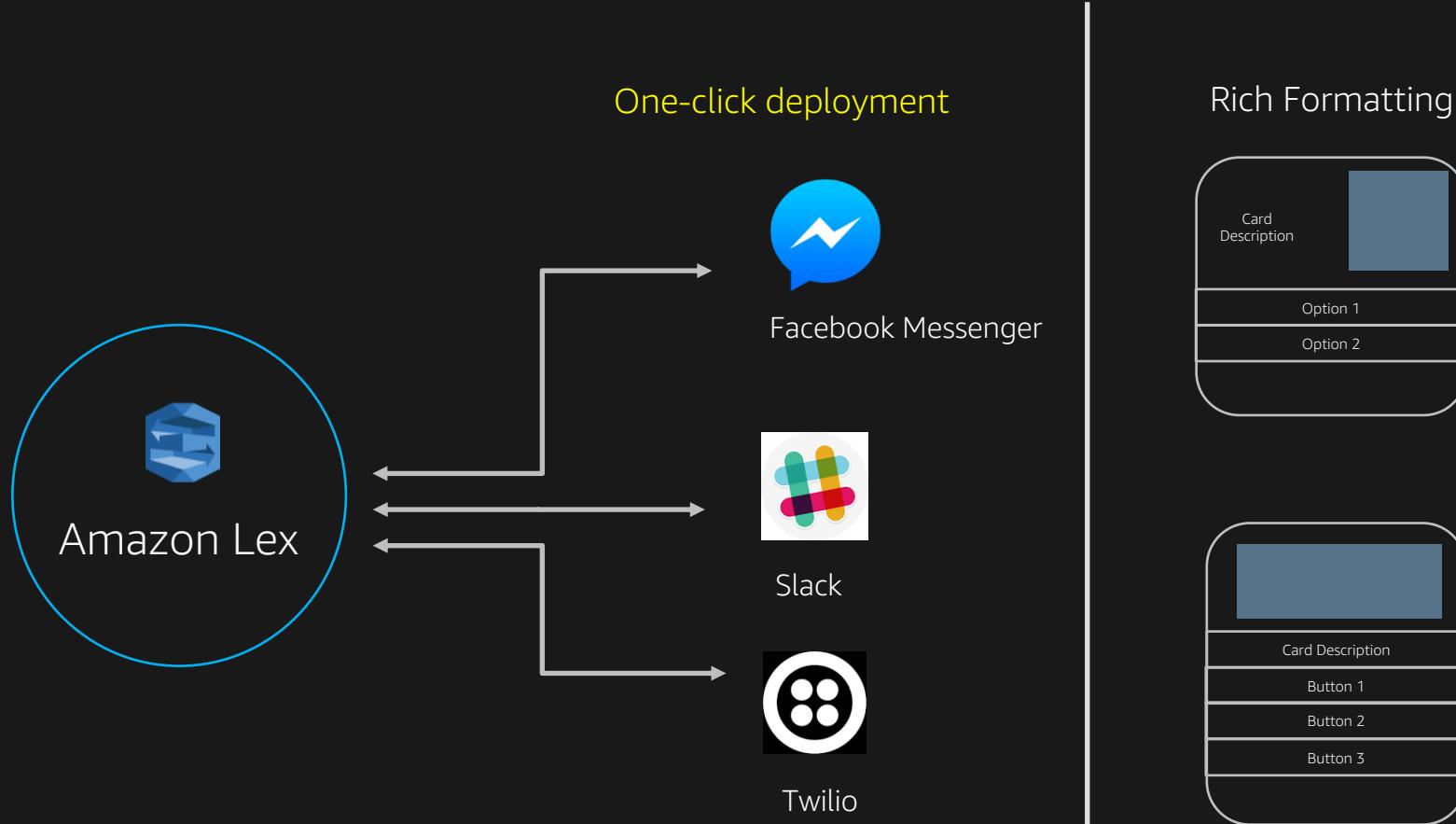
Speech and Text input

Detailed slot information

Dialog state

Rich message format display

Deployment to Chat Services



Monitoring



Granular monitoring



Integrated CloudWatch
Metrics



Missed utterances

Simplified monitoring for a business user



Multi-dev environment
CICD

Comprehensive
SDKs

Integration with Lambda

Application Development Environment



Easy to use Console



UX Support



Sample bots
& blueprints

Easy to use Console

The screenshot shows the AWS Lambda console interface for a bot named "PoochParlor". The top navigation bar includes "Services", "Resource Groups", "Build", "Publish", and "Support". The left sidebar lists intents: ppAgent, ppCSATSurvey, ppFAQBigDogs, ppFAQServices, ppGreeting, ppScheduleAppt (which is selected), and ppSendSMS. It also shows "Slot types: No slots created" and "Error Handling". The main area has tabs for "Editor", "Settings", "Channels", and "Monitoring". The "Monitoring" tab is active, showing a preview of the bot's conversation:

I can come in on [ApptDate]
Can we come in [ApptDate]

Lambda initialization and validation

- Initialization and validation code hook: PoochParlor_KitchenSink

Slots

Slot Name	Type	Value	Slot Type	Description
ApptDate	Text	ApptDate	Built-in	When would you like
ApptTime	Text	ApptTime	Built-in	What time would yo
ApptName	Text	ApptName	Built-in	May I have a name,
PhoneNum	Text	PhoneNum	Built-in	What's a good num

Confirmation prompt

- Confirmation prompt

Fulfillment

- AWS Lambda function
- Return parameters to client

On the right, a preview window shows a conversation with the bot:

Would you like to schedule an appointment for a particular day?
What about Friday?
We might need a couple of hours and we have an 11 AM slot and a 2 PM slot.

Chat to your bot...

Inspect Response

Dialog State: ElicitSlot

RequestID: b4ec8a3d-173b-11e8-9294-1bce0d648fd1

```
{  
  "dialogState": "ElicitSlot",  
  "intentName": "ppScheduleAppt",  
  "message": "We might need a couple of hours and we have an 11 AM slot and a 2 PM slot.",  
  "messageFormat": "PlainText",  
  "responseCard": null,  
  "sessionAttributes": {},  
  "slotToElicit": "ApptTime",  
}
```

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Easy to use Console

AWS Services Resource Groups Admin/hira-Isengard @ 5098-3... N. Virginia Support

PoochParlor Latest Build Publish ?

Editor Settings Channels Monitoring Refresh response Exit preview

Intents ppAgent ppCSATSurvey ppFAQBigDogs ppFAQServices ppGreeting ppScheduleAppt ppSendSMS

Slot types No slots created

Error Handling

Greetings from the Pooch Parlor; how can we help you? You can ask about our services or schedule an appointment for your loved one.



Next

What would you like to do next?

Services

Schedule an Appointment

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

Easy to use Console

The screenshot shows the AWS Lambda Chatbot configuration interface for a project named "PoochParlor". The "Channels" tab is selected, showing options for Facebook, Kik, Slack, and Twilio SMS. The Facebook configuration is displayed on the right, with fields for Channel Name*, Channel Description, IAM Role (set to AWSServiceRoleForLexChannels), KMS Key (aws/lex), Alias*, Verify Token*, Page Access Token*, and two "Ann Contact Keys". A "Test Chatbot" button is visible on the right side of the interface.

PoochParlor Latest

Build Publish ?

Editor Settings Channels Monitoring

Channels

Facebook Kik Slack Twilio SMS

Facebook

Fill in the form below and click activate to get a callback URL to use with Facebook. You can generate multiple callback URLs. Learn more on steps to integrate with Facebook.

Channel Name*

Channel Description

IAM Role **AWSServiceRoleForLexChannels** ⓘ
Automatically created on your behalf

KMS Key **aws/lex** ⓘ

Alias* ⓘ

Verify Token* **Verify Token** ⓘ

Page Access Token* **Page Access Token** ⓘ

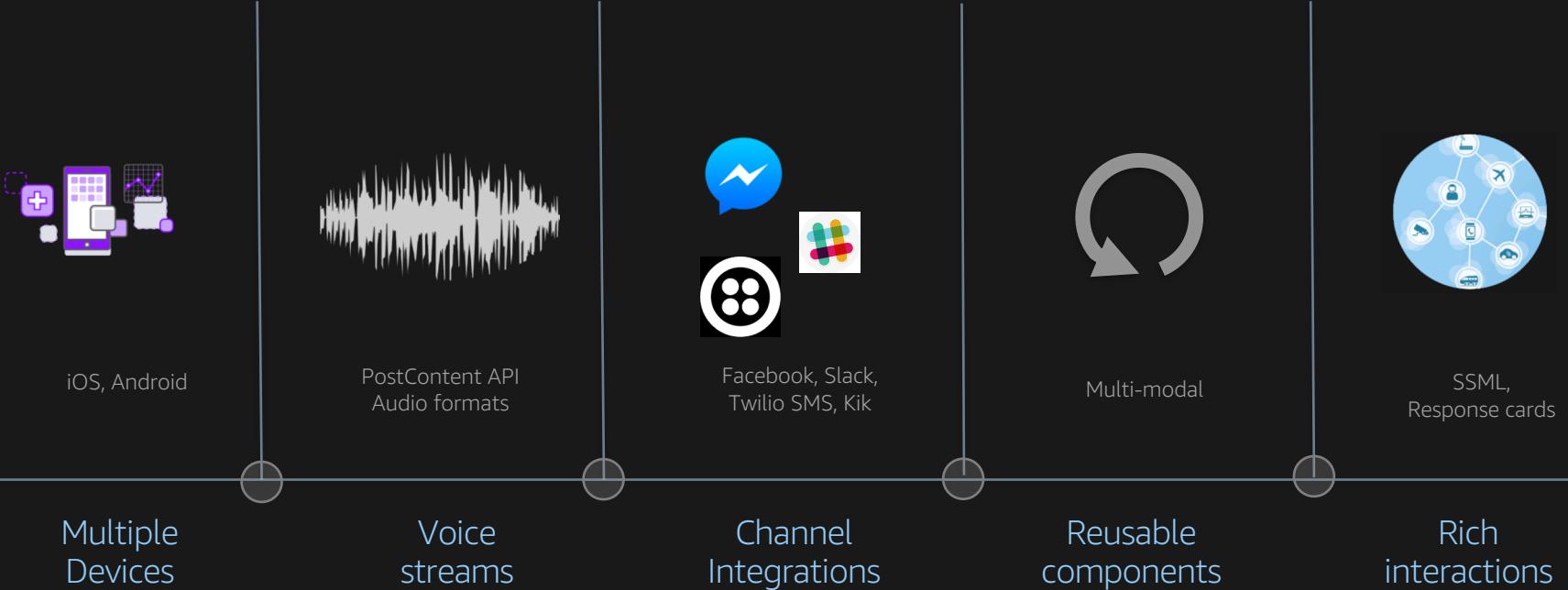
Ann Contact Keys ⓘ

Test Chatbot

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

End user experience



Sample bots & blueprints

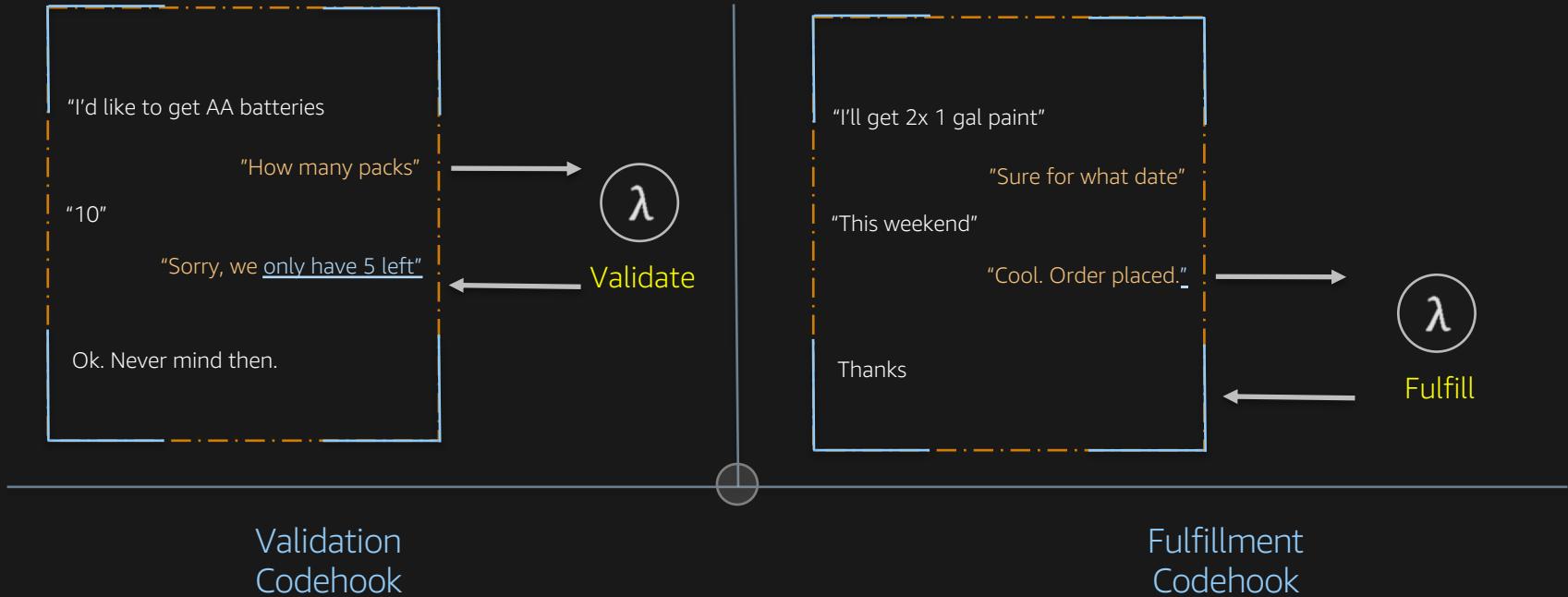
The screenshot shows the 'Create your Lex bot' interface. At the top, there are buttons for 'Custom bot', 'BookTrip', 'OrderFlowers', and 'ScheduleAppointment'. Below these, a 'TRY A SAMPLE' button is visible. A 'Bot name' input field contains 'BookTrip'. Two mobile phone icons illustrate the interaction: one shows a user message 'I'd like to book a hotel.' and a bot response 'Sure, which city?'; the other shows a user message 'Are you sure you want to book the hotel in New York City?' and a bot response 'Yes.' Labels with arrows point from the text to their respective components: 'Intents' points to the first phone's user message; 'Utterances' points to the second phone's user message; and 'Slots' points to the first phone's bot response.

Sample bots

The screenshot shows the 'Create function' interface. It features three main sections: 'Author from scratch' (radio button), 'Blueprints' (radio button, selected), and 'Serverless Application Repository' (radio button). The 'Blueprints' section includes a search bar with 'keyword : book-trip' and a list of two items: 'lex-book-trip-python' and 'lex-book-trip'. Both items are described as 'Book details of a visit, using Amazon Lex to perform natural language understanding' and have 'python2.7 - lex' listed under them. Navigation buttons for 'Feedback', 'English (US)', 'Cancel', and 'Configure' are at the bottom.

Lambda blue prints

Integration with Lambda



Fulfillment



Intents and slots passed
to AWS Lambda function
for business logic
implementation

AWS Lambda Integration



User input parsed to derive
intents and slot values.
Output returned to client
for further processing

Return to Client

Comprehensive SDKs

```
updateFulfillmentActivity.setCodeHook(buildCodeHook(specification.getFulfillmentActivity()));
if(updateFulfillmentActivity.getCodeHook() != null) {
    addPermission(specification.getName(), updateFulfillmentActivity.getCodeHook());
}
updateIntentRequest.setFulfillmentActivity(updateFulfillmentActivity);
updateIntentRequest.setRejectionStatement(specification.getRejectionStatement());
updateIntentRequest.setSampleUtterances(specification.getSampleUtterances());
updateIntentRequest.setIntentToOverrideId(specification.getIntentToOverrideId());
UpdateIntentResponse updateIntentResponse = addCredentials(modelBuildingServiceClient.newUpdateIntentRequest(updateIntentRequest).execute());
```

Add Utterances

```
UpdateSlotTypeRequest updateSlotTypeRequest = new UpdateSlotTypeRequest();
updateSlotTypeRequest.setChecksum(getResponse.getChecksum());
updateSlotTypeRequest.setName(postFixAsNecessary(specification.getName()));
updateSlotTypeRequest.setDescription(specification.getDescription());
updateSlotTypeRequest.setEnumerationValues(specification.getEnumerationValues().entrySet()
    .map(entry -> EnumerationValue.builder().WithValue(entry.getKey()).build())
    .collect(Collectors.toList()));
UpdateSlotTypeResponse updateResponse = addCredentials(modelBuildingServiceClient.newUpdateSlotTypeRequest(updateSlotTypeRequest).execute());
```

Add Slot Values

... build, test and deploy bots using SDKs

SDKs: Java, JavaScript, Python, CLI, .NET, Ruby
on Rails, PHP, Go, and CPP

Dialog Directives

Close (Finish conversation)— Lex doesn't expect a response from the user after this. For eg., "*Your pizza order has been placed*" does not require a response.

ConfirmIntent (Asking for confirmation) — To ask user for a yes or no answer to confirm the current intent. Eg. "Are you sure you want a large pizza?" This confirmation question is set through the message field in the output JSON. The result of the confirmation comes as confirmationStatus field in the input event created after user has responded.

ElicitSlot (Asking for information)— To ask user for a value. Eg. "What size pizza would you like?" The above value will go into a slot. This slot is specified by the slotToElicit field in the output JSON. Lex will set the user provided input to the slot value in the next input event.

ElicitIntent (Start a dialog) — To start a new intent. After this directive, Lex expects user to say some utterance that will trigger an intent.

Delegate — This is an interesting one. Delegate tells Lex that here's the event, go decide your next steps of action. Delegate is particularly useful when you want to pre-fill some default slot values before lex starts processing an event. Eg. we want to set userName or deliveryAddress slot and then let Lex proceed with handling the event.



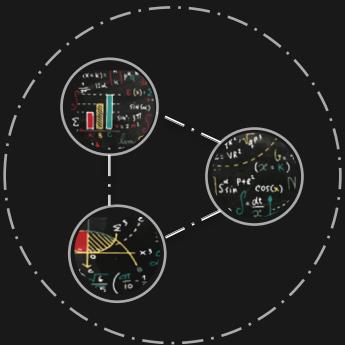
Handling state in conversation

When we're having dialogs back and forth, sometimes we want to maintain state information. The start to end of a dialog is called a session. To maintain any data between dialogs, add them to the sessionAttributes key in the response JSON.

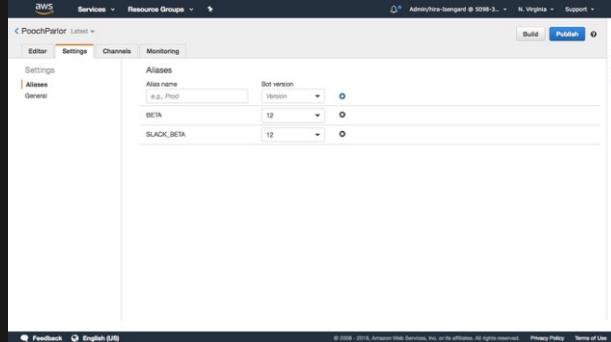
```
{  
  "sessionAttributes": {  
    "key1": "value1",  
    "key2": "value2"  
    ...  
  },  
  "dialogAction": {  
    "type": "ElicitIntent, ElicitSlot, ConfirmIntent, Delegate, or Close"  
  }  
}
```



Multi-dev environment & CICD



Granular control to resources

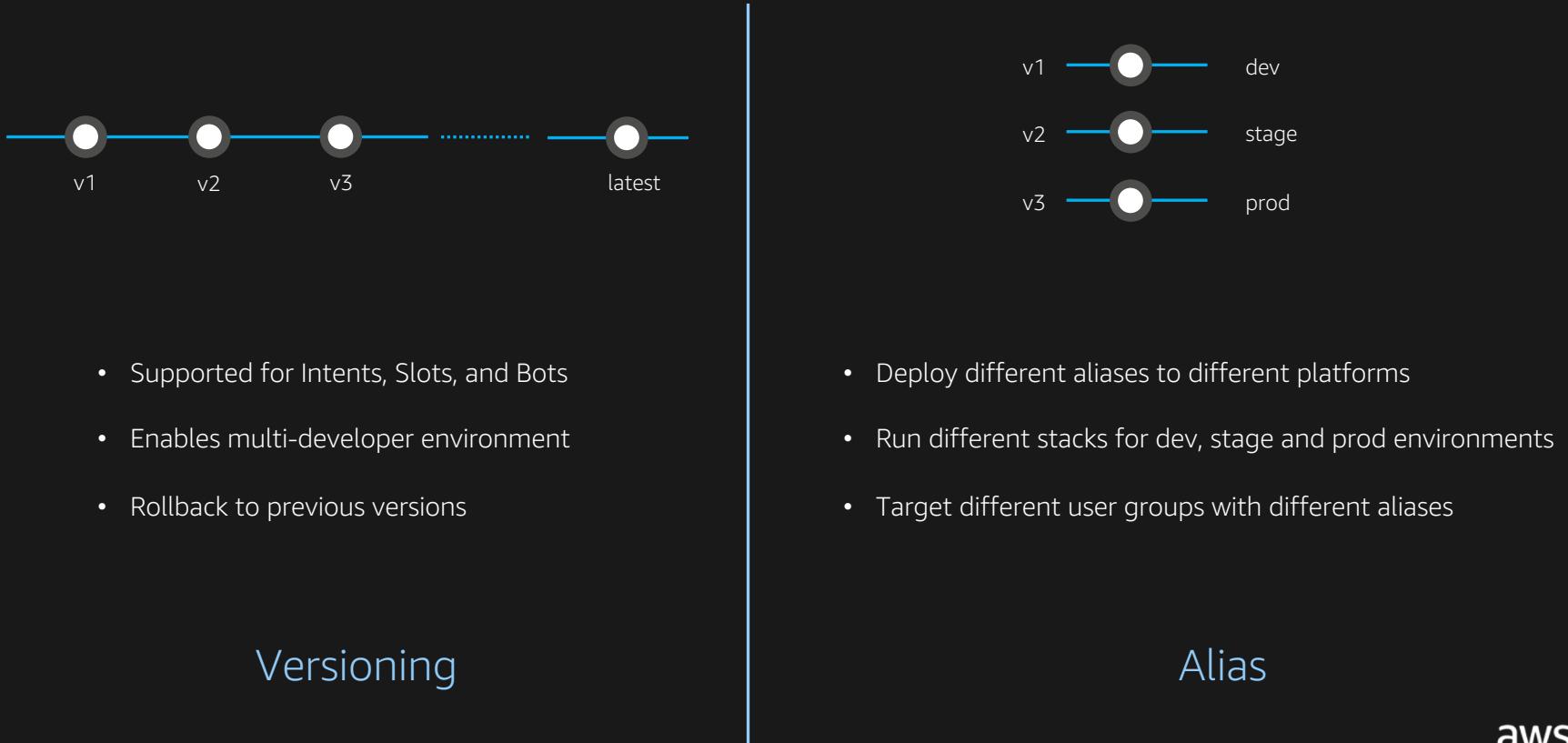


Versioning and Alias



Import/Export

Versioning and Alias support



Security

Authentication &
Authorization

Sigv4 signing

IAM roles

Amazon Cognito integration

OAuth tokens

Encryption

Session attributes

User utterance data

Channel configuration credentials with customer managed keys
(FB token, Slack team token)

HTTPS endpoints

Conversational Best Practices

Social Intelligence

- Personalize the conversation based on user's social profile
- Maintain context to build intelligent conversations

Personality

- Use multiple prompts to give your bot a personality
- Leverage different voices depending on target audience

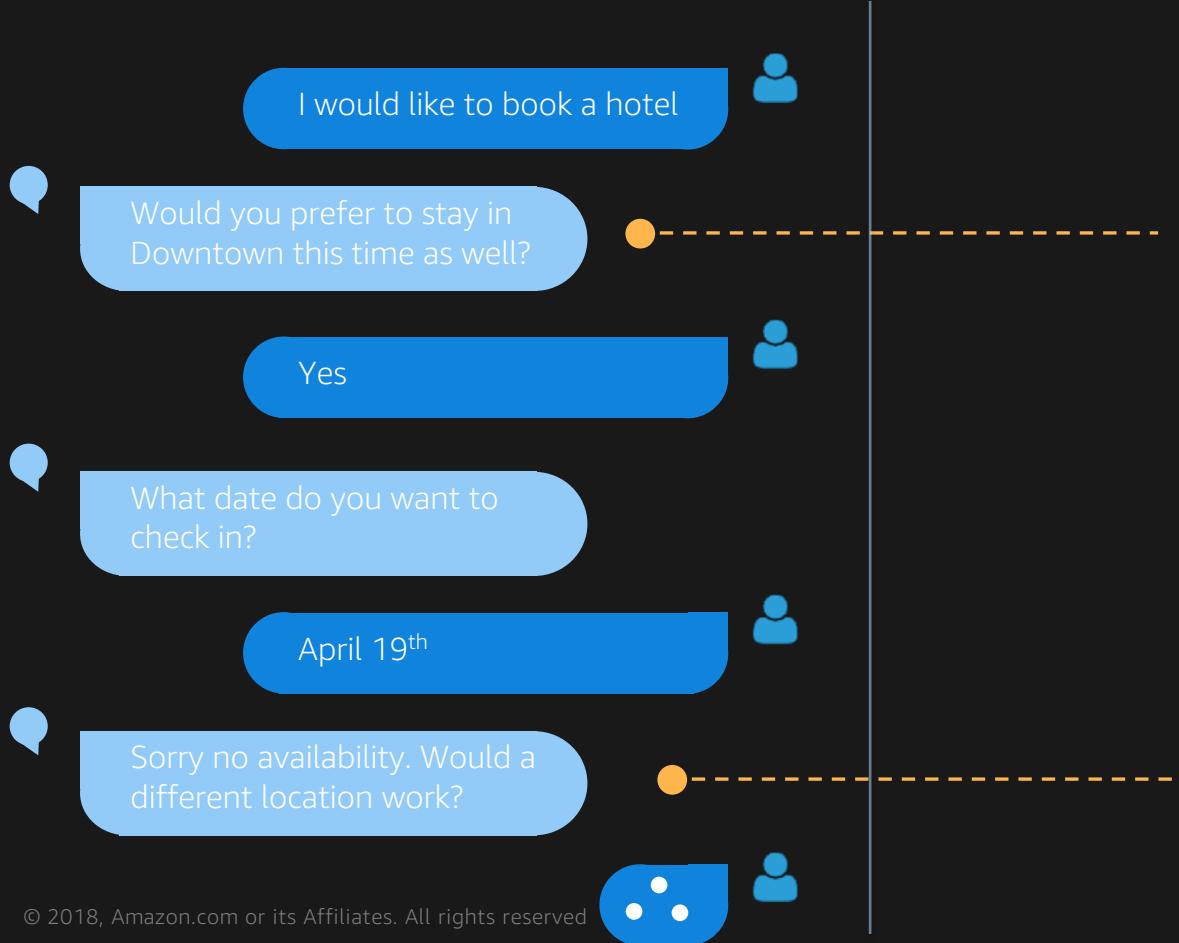
Modality

- Leverage rich formatting capabilities for a more engaging experience
- Create conversation combining speech and text in the same session

Dynamic Conversation

- Built in error handling allows you to manage unexpected scenarios
- Use intent chaining to build a conversation flow based on user input

Customize conversations



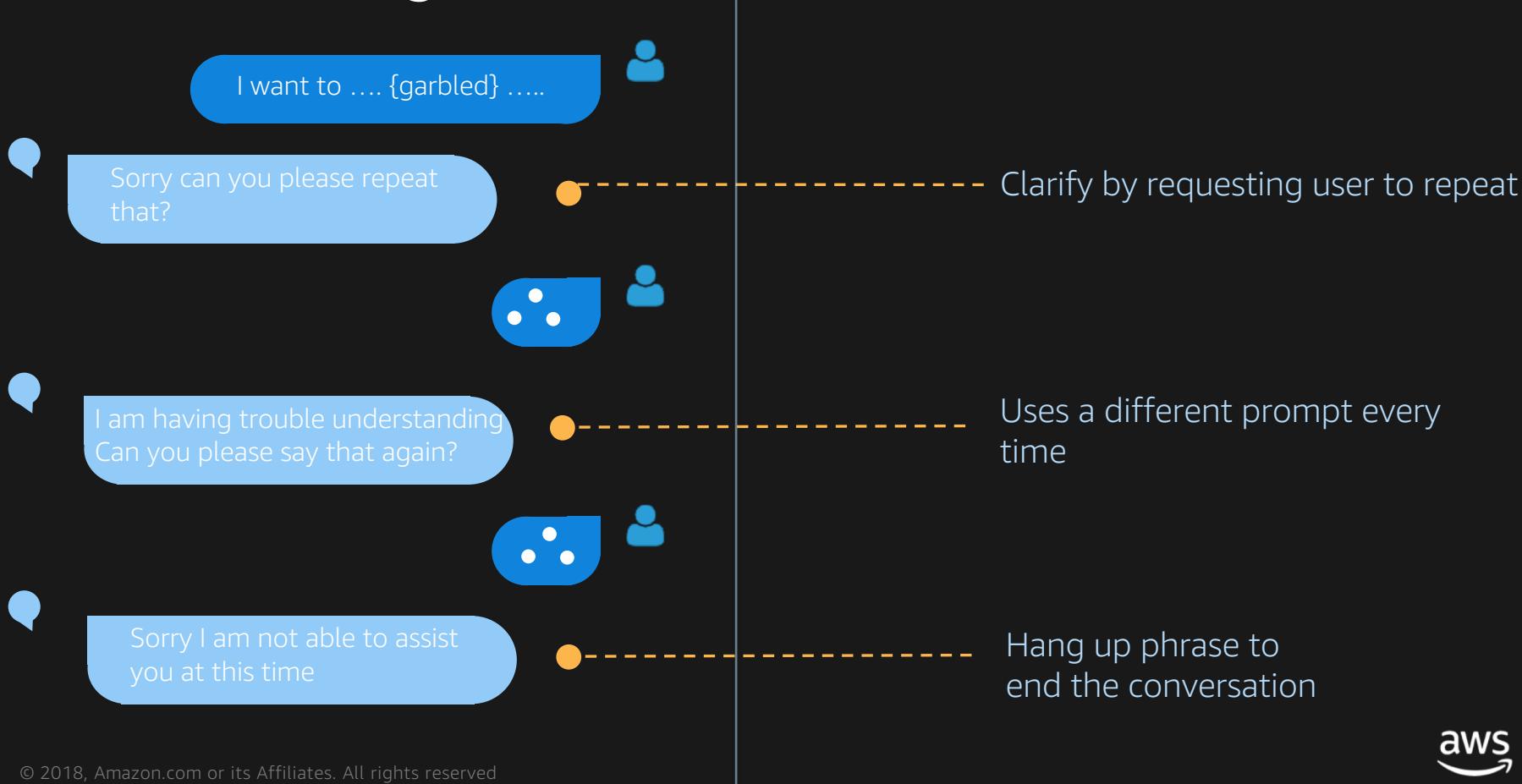
Personalize conversation based on user preferences



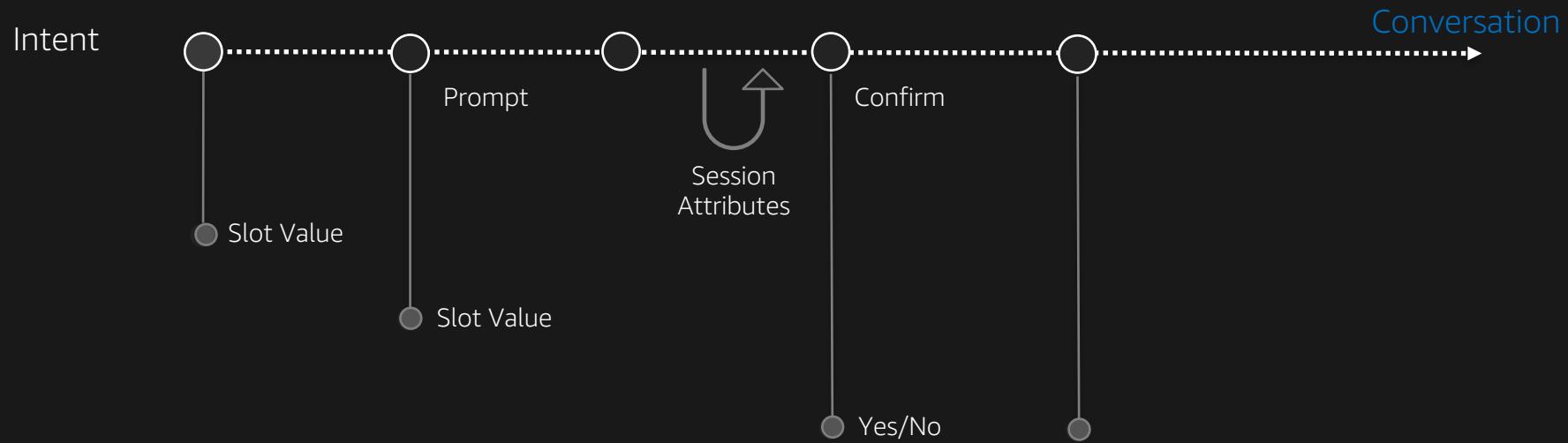
Validate user input and re-prompt as necessary



Error handling



Conversation context



Lex maintains context by storing data throughout the conversation

Slot Values

Intents

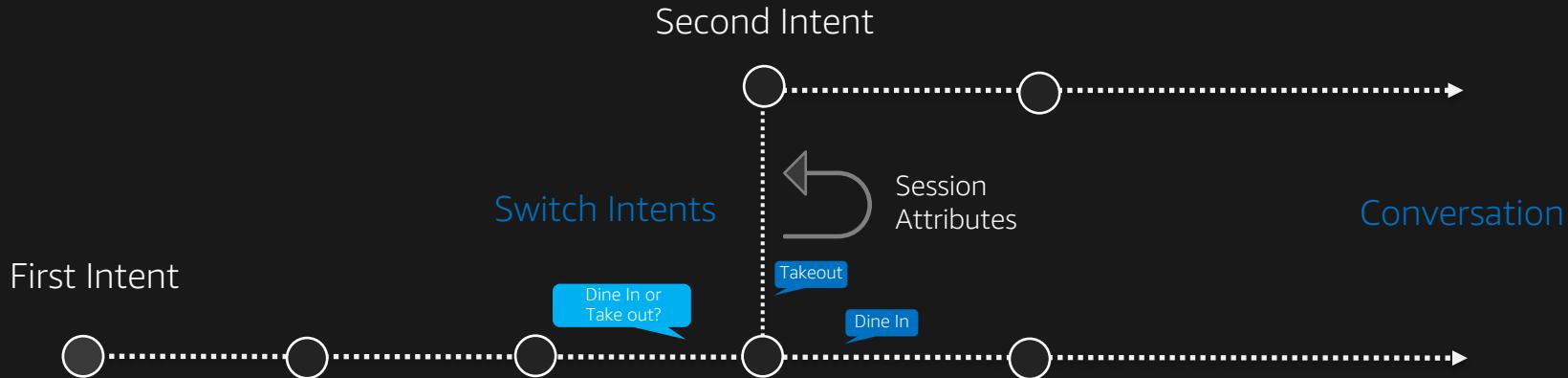
Prompts

Session Attributes

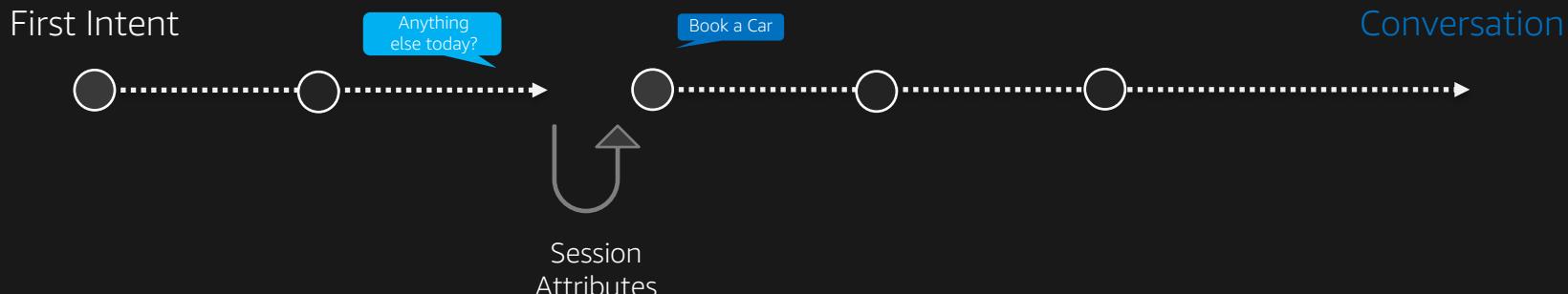
Confirmations



Dynamic conversation flow



Chain Intents



AWS Mobile integration

Authenticate users



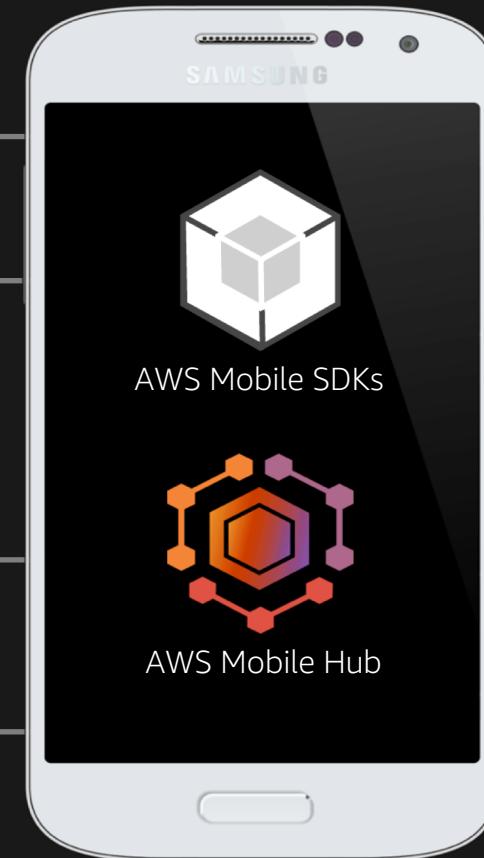
Synchronize data



Analyze user behavior



Track retention



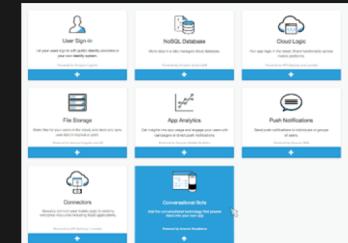
Store and share media



Conversational bots



Amazon Lex



More



Continuous learning

Improve

Add to intent



Analyze

Missed utterances



Monitor

Amazon CloudWatch metrics





Services ▾

Resource Groups ▾



Admin/hira-Isengard @ 5098-3...

N. Virginia

Support ▾

PoochParlor 12 ▾

Build

Publish



Test Chatbot >

Editor

Settings

Channels

Monitoring

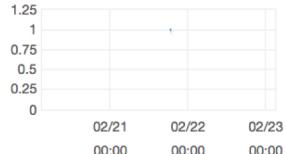
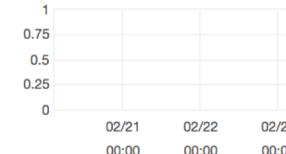
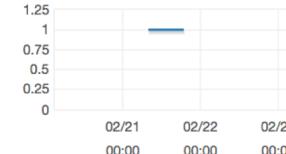
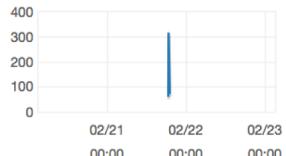
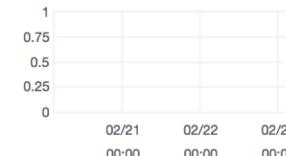
Charts

Monitoring Graphs

Tables

Utterances

Below are your CloudWatch metrics for the selected resources. Click on a graph to see an expanded view. [View all CloudWatch metrics](#)

Text Requests (PostText) Count**Text Requests (PostContent) Count****Speech Requests Count****Text Requests Latency (PostText) MS****Text Requests Latency (PostContent) MS****Speech Requests Latency MS****Text Missed Utterances (PostText) Count****Text Missed Utterances (PostContent) Count****Speech Missed Utterances Count**[Feedback](#)[English \(US\)](#)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)

Integration with CloudWatch

Editor

Settings

Channels

Monitoring

Charts

Monitoring Graphs

Tables

Utterances

Utterances

Add utterance to Intent ▾

Filter: Filter by keyword

Detected

Missed

	Utterances	Count	Status	Last said date
<input type="checkbox"/>	do you do big dogs	5	Missed	February 21, 2018 at 12:48:48 PM UTC-6
<input type="checkbox"/>	sure tell me about your servi	4	Missed	February 21, 2018 at 12:49:05 PM UTC-6
<input type="checkbox"/>		3	Missed	February 21, 2018 at 11:24:46 AM UTC-6
<input type="checkbox"/>	tell me about your services	2	Missed	February 21, 2018 at 11:23:50 AM UTC-6
<input type="checkbox"/>	any	2	Missed	February 21, 2018 at 2:04:05 AM UTC-6
<input type="checkbox"/>	sure	2	Missed	February 21, 2018 at 11:23:43 AM UTC-6
<input type="checkbox"/>	Hi	1	Missed	February 21, 2018 at 12:34:34 PM UTC-6
<input type="checkbox"/>	Sure, tell me about your serv	1	Missed	February 21, 2018 at 12:35:09 PM UTC-6
<input type="checkbox"/>	Do you handle big dogs?	1	Missed	February 21, 2018 at 12:34:56 PM UTC-6
<input type="checkbox"/>	i want a	1	Missed	February 21, 2018 at 11:19:15 AM UTC-6
<input type="checkbox"/>	change	1	Missed	February 21, 2018 at 11:19:50 AM UTC-6
<input type="checkbox"/>	can i have a	1	Missed	February 21, 2018 at 11:24:09 AM UTC-6

Feedback

English (US)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Monitor bots



Amazon Lex Lab

Lab

<https://master.d78l3q0u6mko7.amplifyapp.com/03-hands-on/06-amazon-lex/>

Thank You