

# Build Your Own Chatbot

Arun Gupta @arungupta

Amazon Web Services

Docker Captain  
Java Champion  
JavaOne Rock Star (4 years)  
NetBeans Dream Team  
Silicon Valley JUG Leader  
Author  
Runner  
Lifelong learner



# STAR WARS CHATBOT

Tell me a Yoda quote

# Developer challenges

Automated Speech  
Recognition

Business Logic

Authentication

Testing

Mobile

Scalability

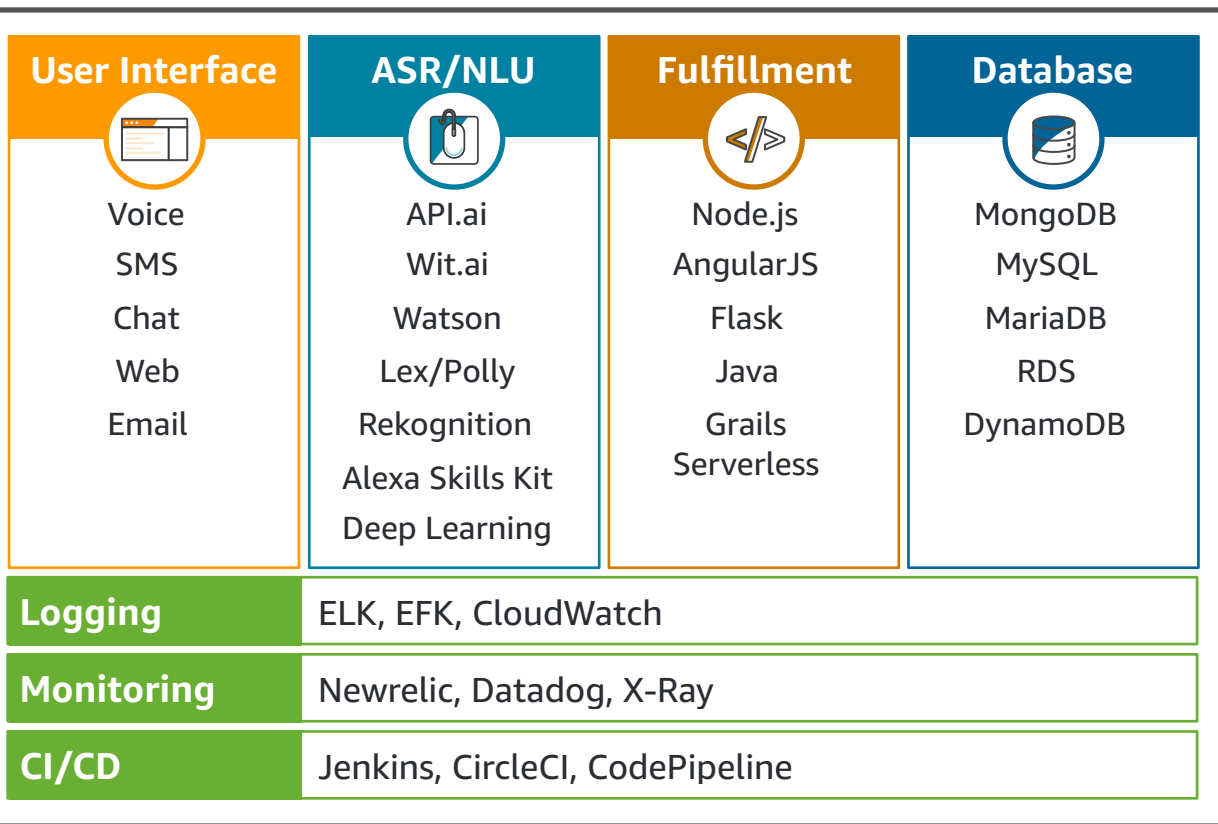
Dialog Manager

Availability

Natural Language  
Understanding

Disparate Systems

# CHATBOX Components



## BookHotel

### Intents

A particular goal that the user wants to achieve

### Utterances

Spoken or typed phrases that invoke your intent

### Slots

Data the user must provide to fulfill the intent

### Prompts

Questions that ask the user to input data

### Fulfillment

The business logic required to fulfill the user's intent

I'd like to book a hotel.

Sure, which city?

New York City

What date do you check in?

...



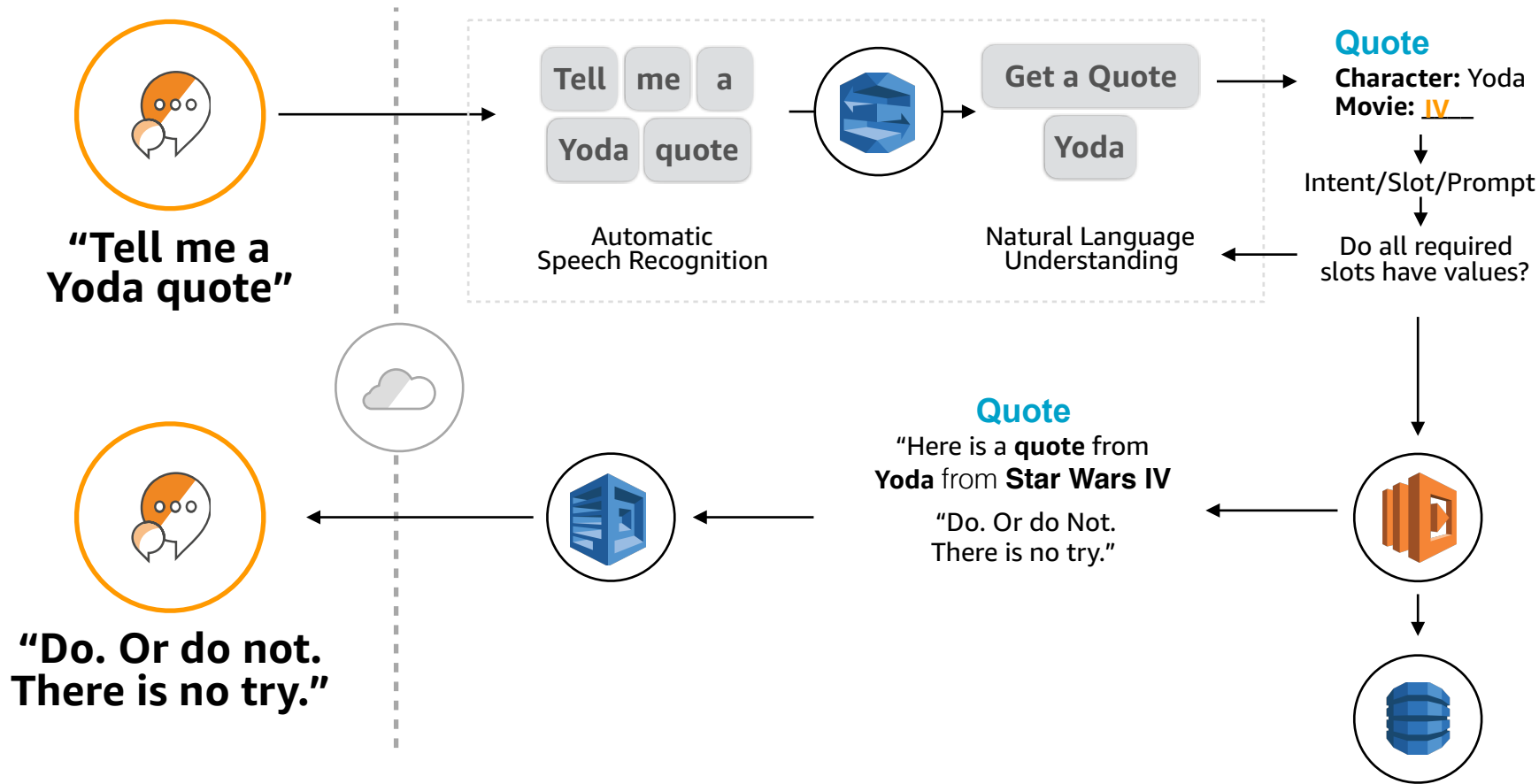
November 30th.



Are you sure you want to book the hotel in New York City?

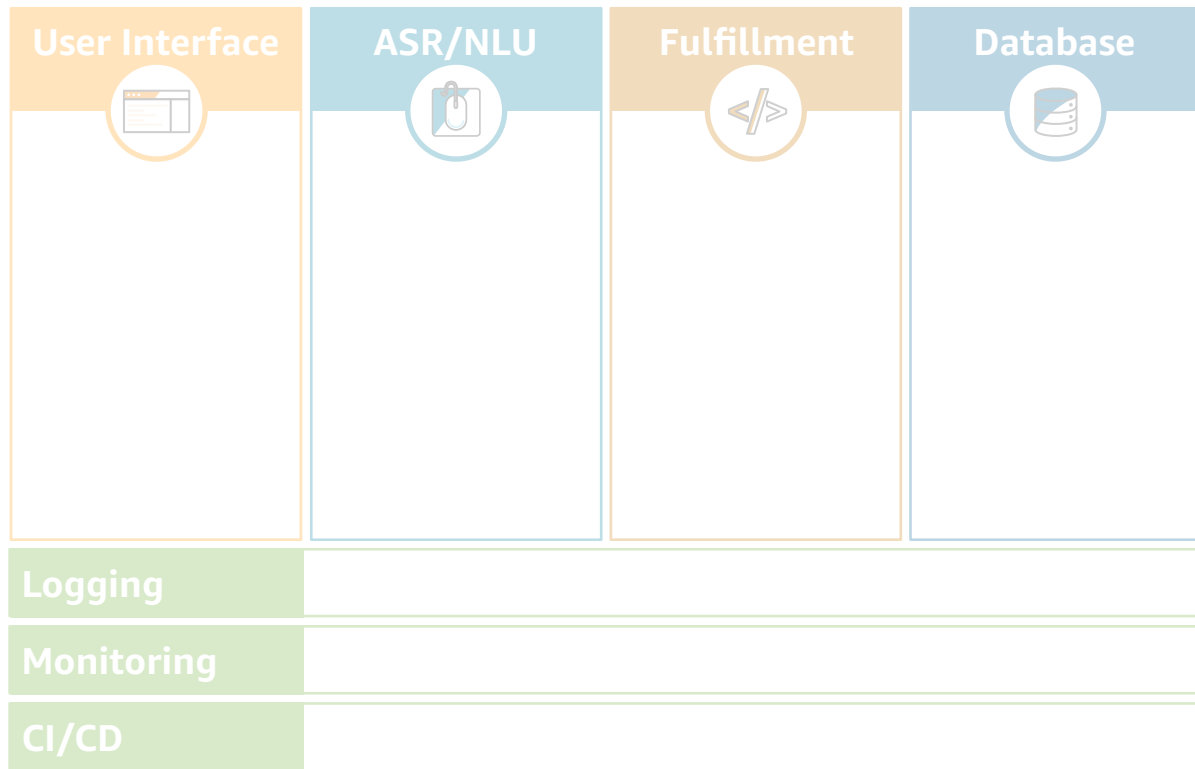
Yes.

Thank you. The reservation went through successfully.

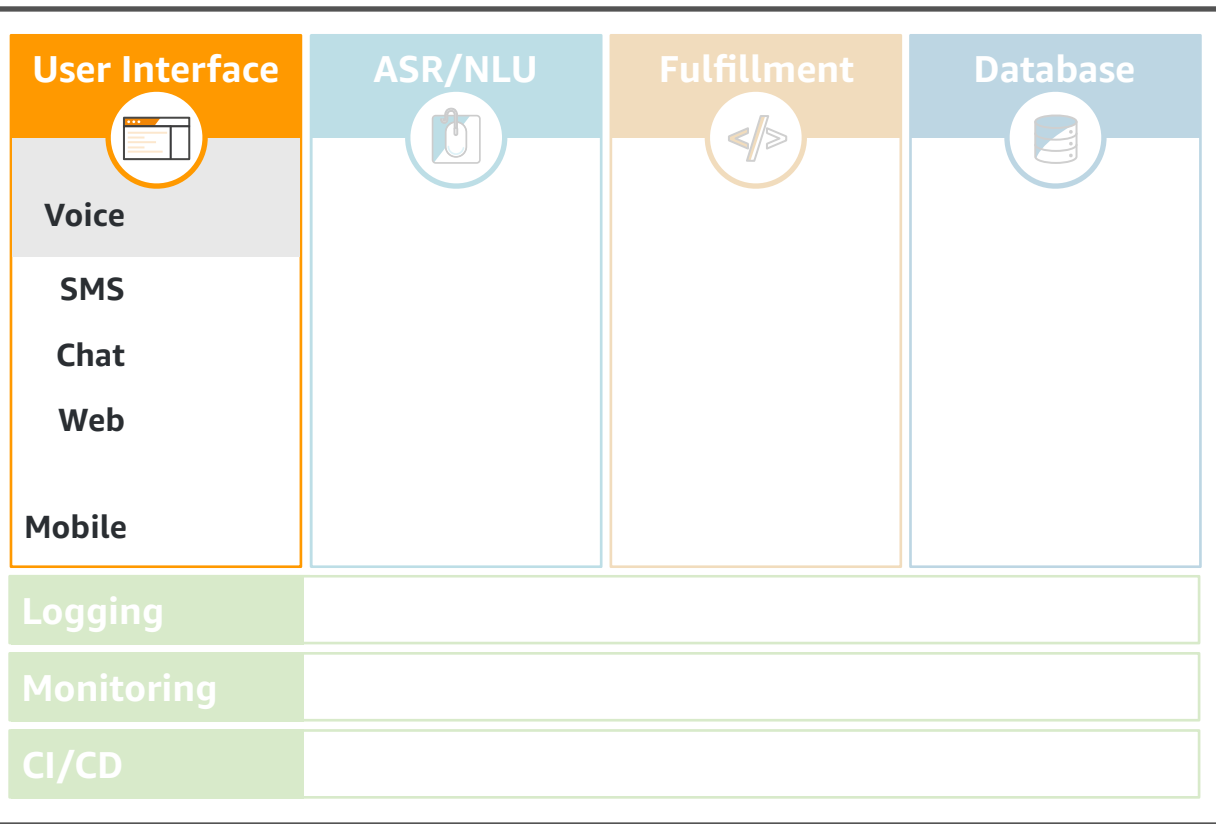




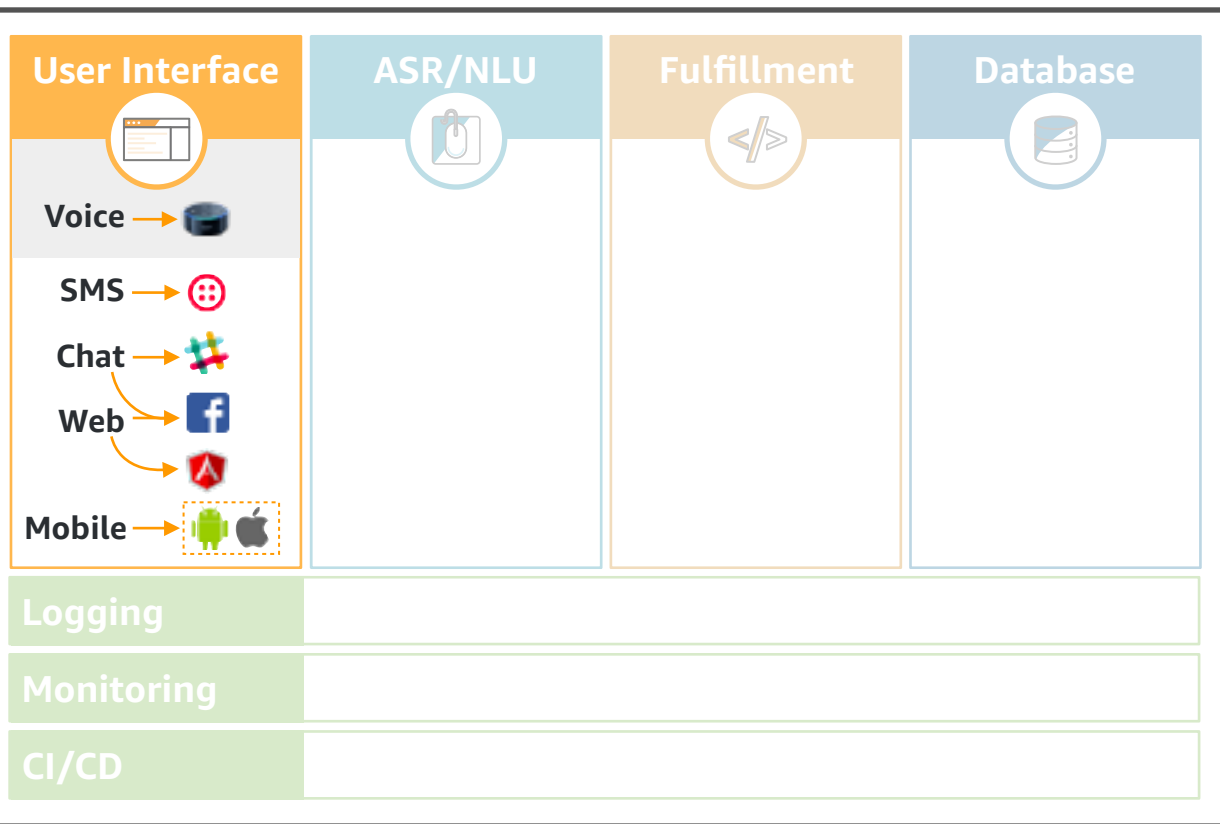
# CHATBOX Architecture



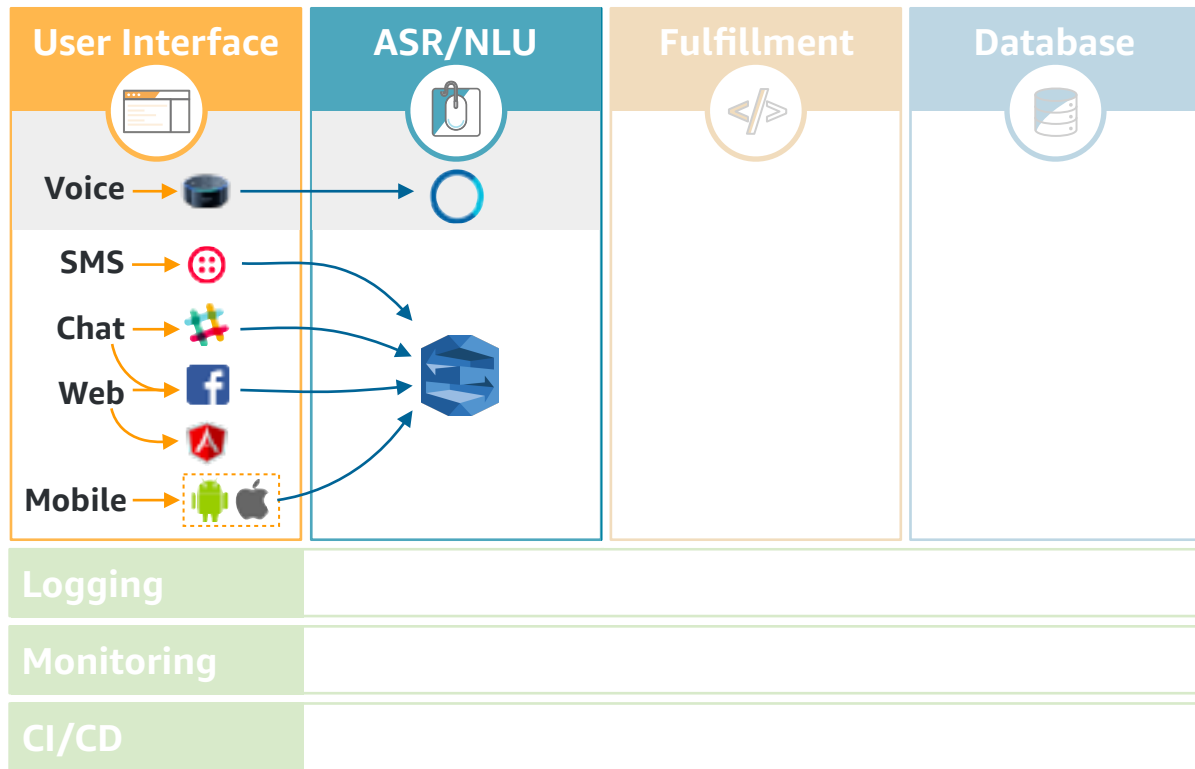
# CHATBOX Architecture



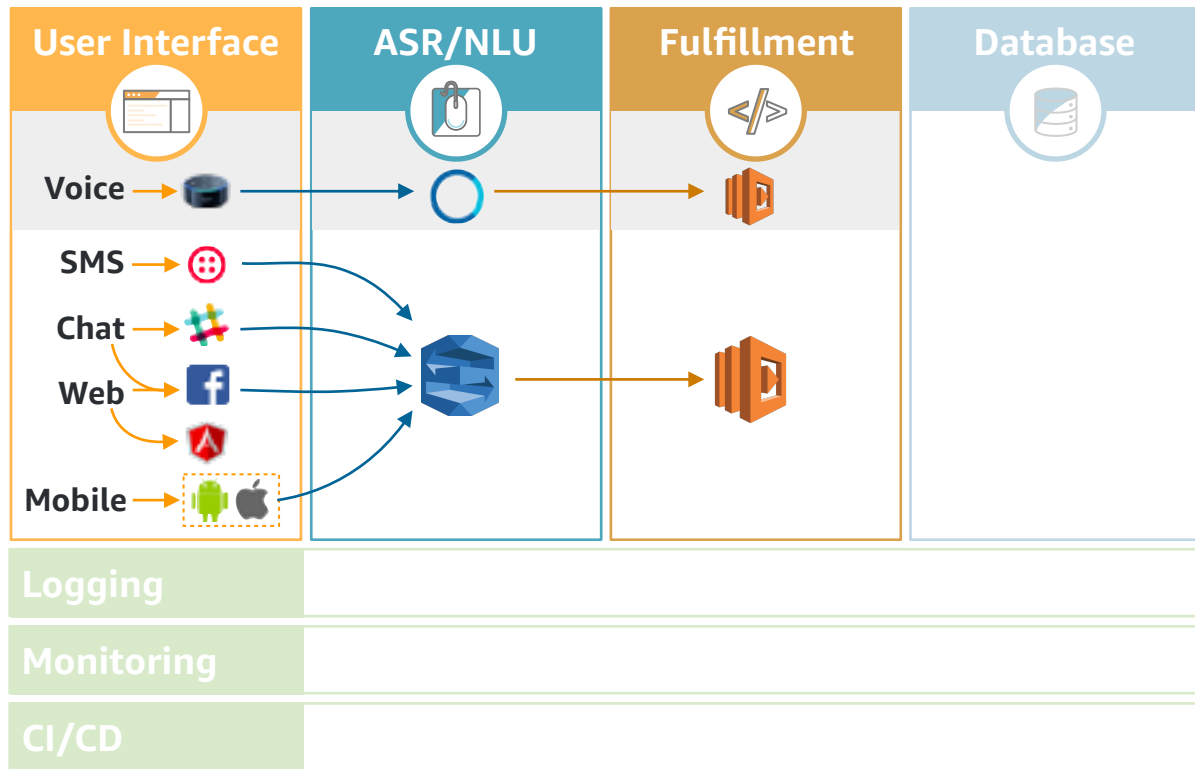
# CHATBOX Architecture



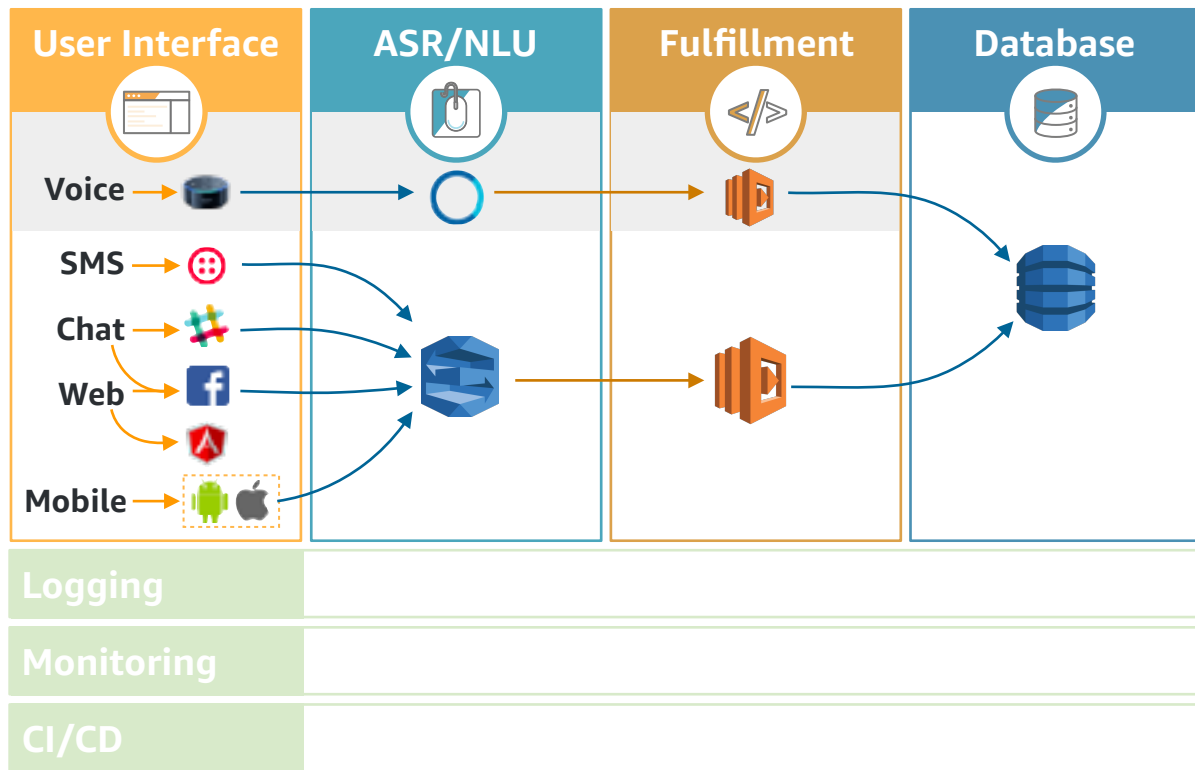
# CHATBOX Architecture



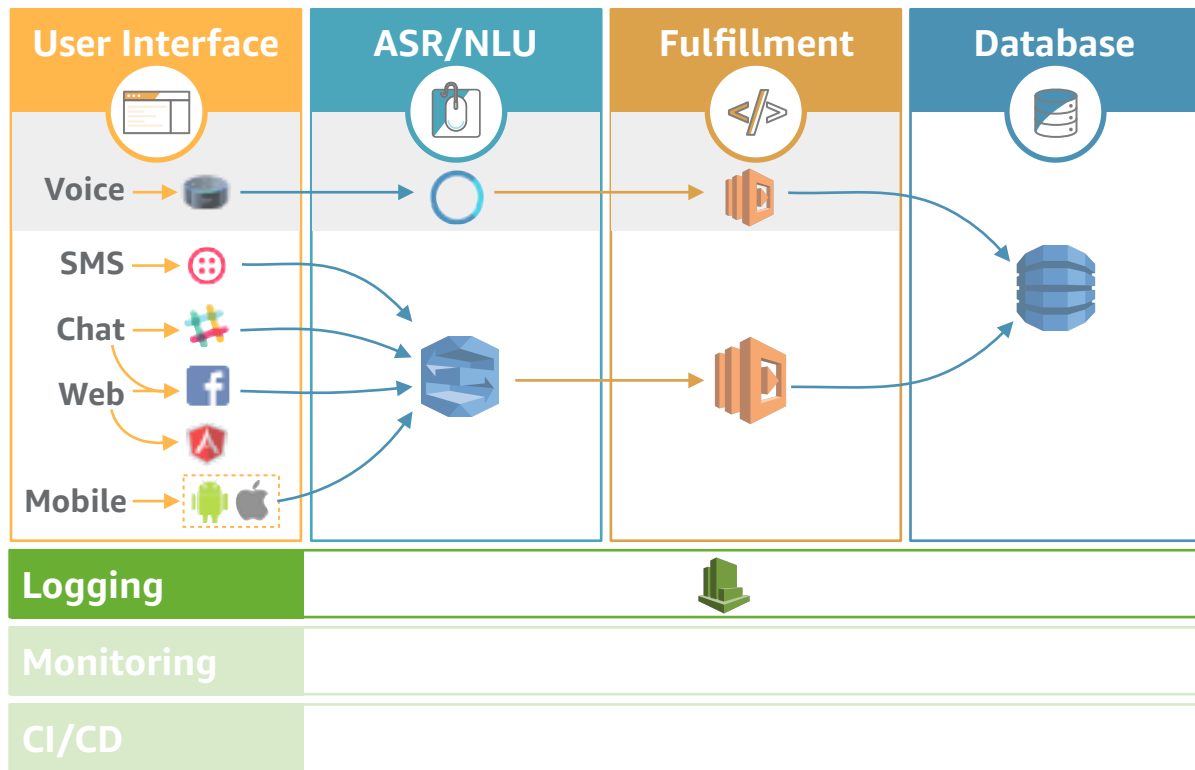
# CHATBOX Architecture



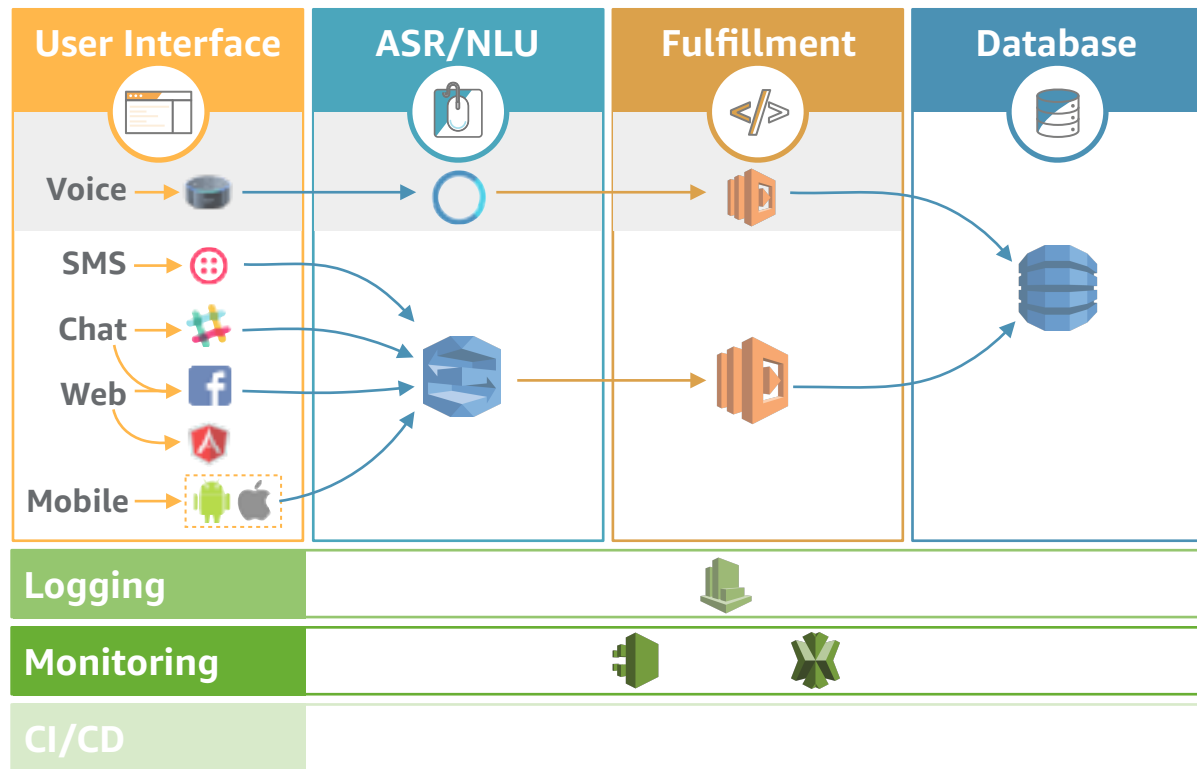
# CHATBOX Architecture



# CHATBOX Architecture

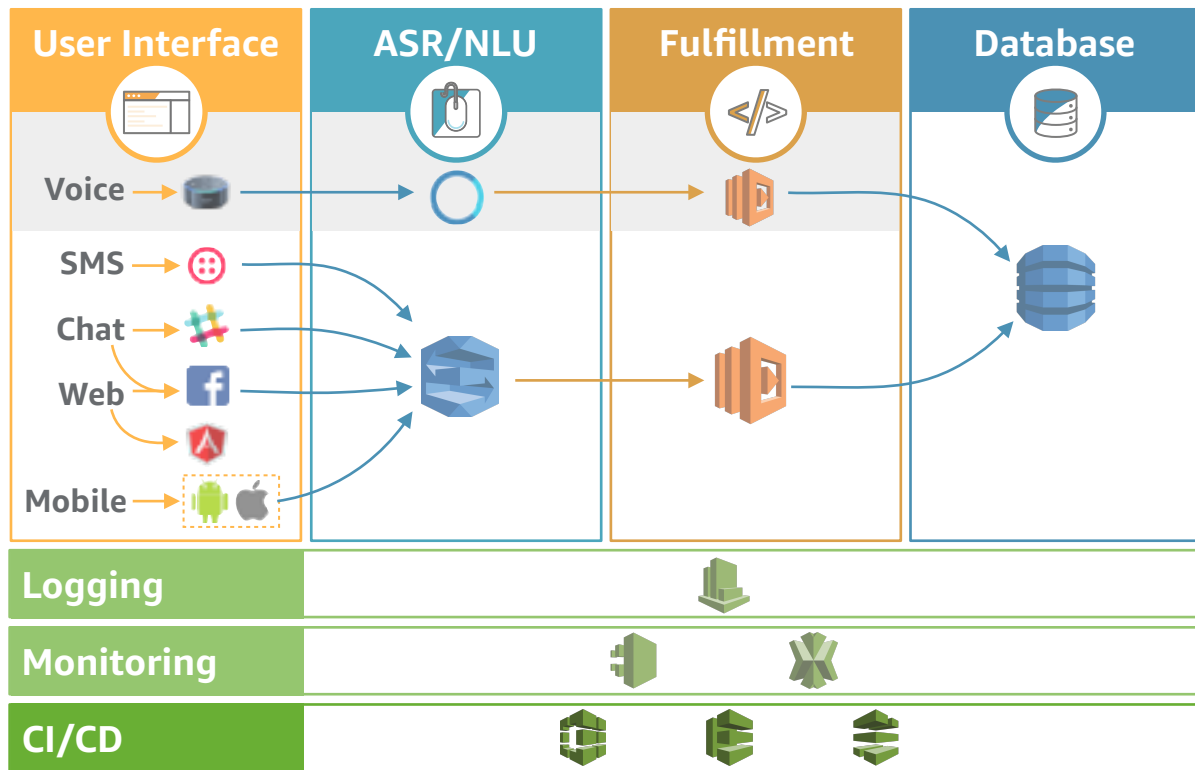


# CHATBOX Architecture





# CHATBOX Architecture



# BYOC Build Your Own Chatbot



## Build a Lex bot

### Voice User Interface (VUI)

Define intent, utterances,  
slot types, slots, prompts

Write Lambda function  
for business logic  
(intent → database)

Prepare database, setup indexes

Integrate with Twilio/SMS,  
Slack, Facebook, Web, mobile



## Build an Alexa skill (optional)

Export Lex bot  
configuration

Write Lambda function  
for business logic  
(intent → database)



---

# DynamoDB

Fast, fully-managed NoSQL database service

Both key/value and document

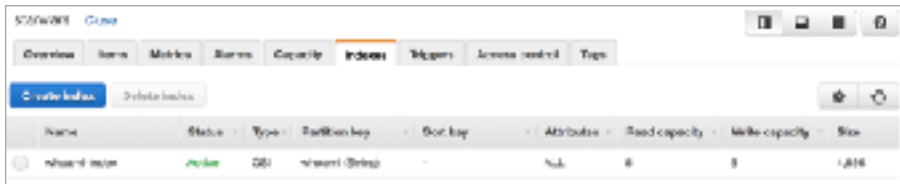
Capable of handling any amount of data

Durable and highly available

All SSD storage

Simple and cost effective

# JSON Document



Name	Status	Type	Path	Group	Attributes	Last capacity	Size
root	Active	Root	/	root	root	0	0

```
{
  "dead": true,
  "force-sensitive": true,
  "force-side": "light",
  "id": 1,
  "lightsaber": "green",
  "planet": "Dagobah",
  "quotes": [
    "When nine hundred years old you reach, look as good you will not.",
    "Truly wonderful, the mind of a child is",
    "A Jedi uses the Force for knowledge and defense, never for attack",
    "That is why you fail.",
    "Adventure. Excitement. A Jedi craves not these things.",
    "Judge me by my size, do you?",
    "Fear is the path to the dark side",
    "Wars not make one great",
    "Do, or do not. There is no try",
    "Size matters not",
    "The dark side clouds everything",
    "Impossible to see the future is",
    "Clear your mind must be",
    "Much to learn you still have ... my old padawan"
  ],
  "weapon": "lightsaber",
  "whoami": "Yoda"
}
```



---

# Amazon Lex

Service for building conversational interfaces using voice and text

## Provides

Automatic Speech Recognition (ASR): speech → text

Natural Language Understanding (NLU): text → intent

Powered by the same deep learning technologies as Alexa

<http://aws.amazon.com/lex>



# Build a Lex Bot

Services ~ Resource Groups ~

Lex

- Bots
- Intents
- Slot types

Bots

Create Actions

Filter:

Name	Status	Last updated
<input type="radio"/> <a href="#">OrderFlowers</a>	READY	September 3, 2017 at 6:59:18 PM UTC+2
<input type="radio"/> <a href="#">StarWars</a>	READY	September 9, 2017 at 1:31:23 AM UTC+2

## Multi-turn Conversations

Default session timeout is **5 mins**,  
can be up to **24 hours**



# Lex and Twilio

## SMS INTEGRATION

**Twilio:** Build software that communicates with everyone in the world

Create a Twilio SMS endpoint

Specify Twilio credentials in Lex bot

Create a Twilio SMS endpoint (phone #)

Set Callback URL in Messaging  
of Twilio SMS endpoint

When message comes in

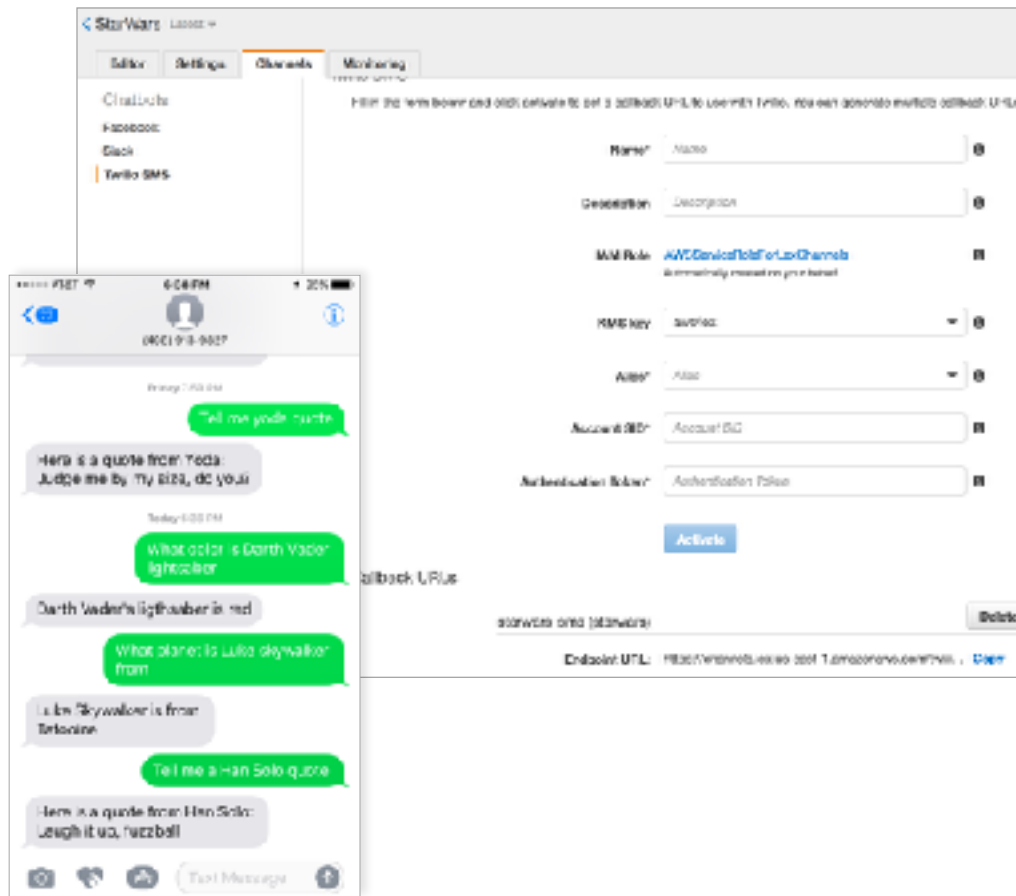
<https://docs.aws.amazon.com/lex/latest/dg/twilio-bot-association.html>

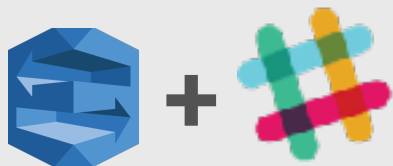




# Lex and Twilio

## SMS INTEGRATION





# Lex and Slack

## INTEGRATION

Create a Slack application: [api.slack.com](https://api.slack.com)

Add a Bot to the application

Always online

Enable interactive messages

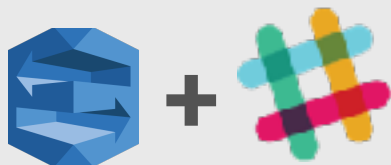
Specify Slack credentials in Lex bot

Specify URLs in Slack application

Postback: Lex bot's endpoint that listens to Slack events

OAuth: Lex bot's handshake with Slack

<https://docs.aws.amazon.com/lex/latest/dg/slack-bot-association.html>



# Lex and Slack INTEGRATION

The screenshot displays the Slack interface for a chatbot named 'star\_wars\_chatbot'. The chat history shows a conversation where the user 'arungspta' asks for a Yoda quote, and the chatbot responds with 'Here is a quote from Yoda: That is why you fail.' The user then asks for Luke Skywalker's lightsaber color, and the chatbot responds 'Luke Skywalker's lightsaber is blue.' Finally, the user asks which planet Darth Vader is from, and the chatbot responds 'Darth Vader is from Tatooine.'

The settings panel on the right shows various configuration options for the chatbot, including:

- MM Flow: [AddSkillToSkillSet-C16E7A7608](#)
- MM ID:
- MM ID:
- Client ID:
- Secret:
- Token:
- URL:

The 'Add Skill' button is visible at the bottom of the settings panel.



---

# Lex and Facebook

INTEGRATION

Create a Facebook application

Specify application credentials in Lex bot

Gives Callback URL

Setup webhooks for Facebook application

Specify Callback URL and enable subscription

Optionally, publish the application  
(for public consumption)

<http://docs.aws.amazon.com/lex/latest/dg/fb-bot-association.html>

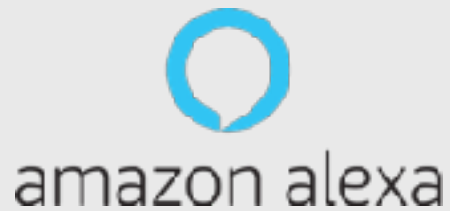


# Web Application

Text here

# Mobile

Text here



# What is Alexa



**Alexa** is a **cloud-based service** that can answer questions, play music, read the news and more



**Echo** is an **always-on always-connected hands-free device** that connects to Alexa





# Alexa Has Skills

[amazon.com/skills](https://amazon.com/skills)

Let the fireworks begin.

"Alexa, open eHarmony."

Get started

"Alexa, what are your top skills?"

Skill Icon	Voice Command	Skill Name	Rating
	"Alexa open Sleep Sounds"	Sleepand Relaxation Sounds	★★★★☆ 5,255
	"Alexa, play Cricut Vrylon StratusM"	CricutM	★★★★☆ 1,259
	"Alexa, what's my Flash Briefing?"	Fox News	★★★★☆ 582
	"Alexa, start Song Quiz"	Song Quiz	★★★★☆ 652
		longairlyf	★★★★☆ 527

***Alexa, ask Star Wars tell me a Yoda quote***

Wake  
word

Starting  
phrase

Skill  
invocation  
name

Utterance

## Customer

## Alexa

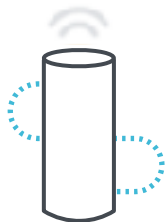
## Your Alexa Skill



**1** Customer asks a question or gives a command

**2** Alexa identifies your skill's name, analyzes and understands the customer's request, then sends your service a structured representation of the user's request

**3** Your service processes the request and returns a text and graphical response



**5a** Customer hears the response from Alexa's voice

**4a** Alexa converts the returned text to speech and streams it to the device



**5b** Customer sees a graphical representation in the companion app

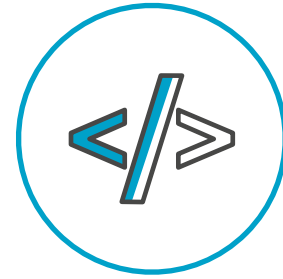
**4b** Your graphical representation (if any) is rendered in the companion app

# Two Sides of an Alexa Skill



Voice User  
Interface

+



Programming  
Logic

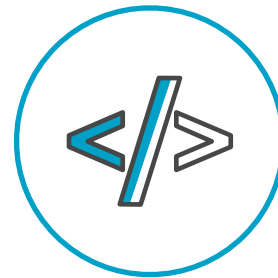
# Two Sides of an Alexa Skill



Voice User  
Interface

[developer.amazon.com](https://developer.amazon.com)

+



Programming  
Logic

[aws.amazon.com](https://aws.amazon.com)



# Build an Alexa Skill

Design a **Voice User Interface**

**Setup** the Skill in the Developer Portal

Can be done using ASK CLI or REST API

Use the Voice Design to **Build Your Interaction Model**

Intents, sample utterances, and the dialog model

**Write and Test the Code** for your Skill

AWS Lambda or web service on any cloud provider

**Beta Test** your Skill

Submit your Skill for **Certification**





# Alexa Skills Kit

```
<dependency>  
  <groupId>com.amazon.alex</groupId>  
  <artifactId>alex</artifactId>  
  <version>1.4.0</version>  
  <scope>compile</scope>  
</dependency>
```

echosim.io provided by Quinix Media. Special thanks to Sam Mathis for his Alexa in the browser [project](#).

## Alexa Skill Testing Tool

Click and **hold** the microphone button or hold down the space bar on your keyboard to activate the microphone.

Tapnet Speech



Clear

Collapse / Expand

SpeechRecognizer.ExpectSpeech 2017-09-21 11:21:18

```
{
  "directive": {
    "header": {
      "dialogRequestId": "dialogRequestId-10b1f102-d922-4e60-b854-4f9993f011f9",
      "namespace": "SpeechRecognizer",
      "name": "ExpectSpeech",
      "messageId": "64b44121-2924-4901-05a5-847f0f1766b"
    }
  },
  "payload": {
    "timeoutInMilliseconds": 8000
  }
}
```

TemplateRuntime.RenderTemplate 2017-09-21 11:21:18

Welcome to Star Wars Trivia, you can ask quotes

Star Wars  
Star Wars Welcome

```
{
  "directive": {
    "header": {
      "dialogRequestId": "dialogRequestId-10b1f102-d922-4e60-b854-4f9993f011f9",
      "namespace": "TemplateRuntime",
      "name": "RenderTemplate",
      "messageId": "0015d9d1-ec01-4e01-b21f-f0e020a905"
    }
  },
  "payload": {
```

[echosim.io](https://echosim.io)





# Security

**Role ARN**      `arn:aws:iam::091144849831:role/lambda_services`

**Role description**      Allows Lambda Function to call AWS services on your behalf.

**Instance Profile ARNs**

**Path**      /

**Creation time**      2017-08-22 18:44 PDT

Permissions    Trust relationships    Access Advisor    Revoke sessions

[Attach policy](#)    Attached policies: 5

Policy name
▶  AmazonDynamoDBFullAccess
▶  AWSLambdaDynamoDBExecutionRole
▶  AmazonLexReadOnly
▶  AmazonLexFullAccess
▶  AmazonLexRunBotsOnly

Permissions    **Trust relationships**    Access Advisor

You can view the trusted entities that can assume the role and the

[Edit trust relationship](#)

**Trusted entities**

The following trusted entities can assume this role.

**Trusted entities**

The identity provider(s) `lex.amazonaws.com`

The identity provider(s) `lambda.amazonaws.com`



---

## LOGGING

# Amazon CloudWatch

Monitoring service for AWS cloud resources and applications

Visibility into resource utilization and operational performance with Metrics and Logs

Set alarm thresholds to send notifications or trigger Auto Scaling

Log aggregation, monitoring and troubleshooting with CloudWatch Logs

Support for Custom Metrics



# LOGGING Amazon CloudWatch

The screenshot displays the AWS CloudWatch console interface. At the top, there are tabs for 'Download Log Group', 'Create Log Stream', and 'Delete Log Stream'. Below these is a filter bar with 'Log Stream Name Prefix' set to 'x'. A table lists several log streams, each with a checkbox, its name, and its last event time. The log stream 'Star Wars: The Force Awakens' is selected. Below the table, the 'Task View' for this log stream is shown, displaying a timeline of log events. The 'Log View' shows the raw log data for the selected task, including the task ID and the log message.



---

**ANALYZE  
& DEBUG**

**AWS  
X-Ray**

AWS X-Ray provides an end-to-end view of requests

Combines the data gathered from each service into singular units called traces

View the service map to see trace data such as latencies, HTTP statuses



---

**ANALYZE  
& DEBUG  
AWS  
X-Ray**

Add a snapshot of X-Ray dump

# Deployment Pipeline



## **AWS CodeBuild**

Managed build service that compiles source code, runs tests and create packages



## **AWS CodeDeploy**

Automates deployment of packages to EC2 instances, rolling updates and health checks



## **AWS CodePipeline**

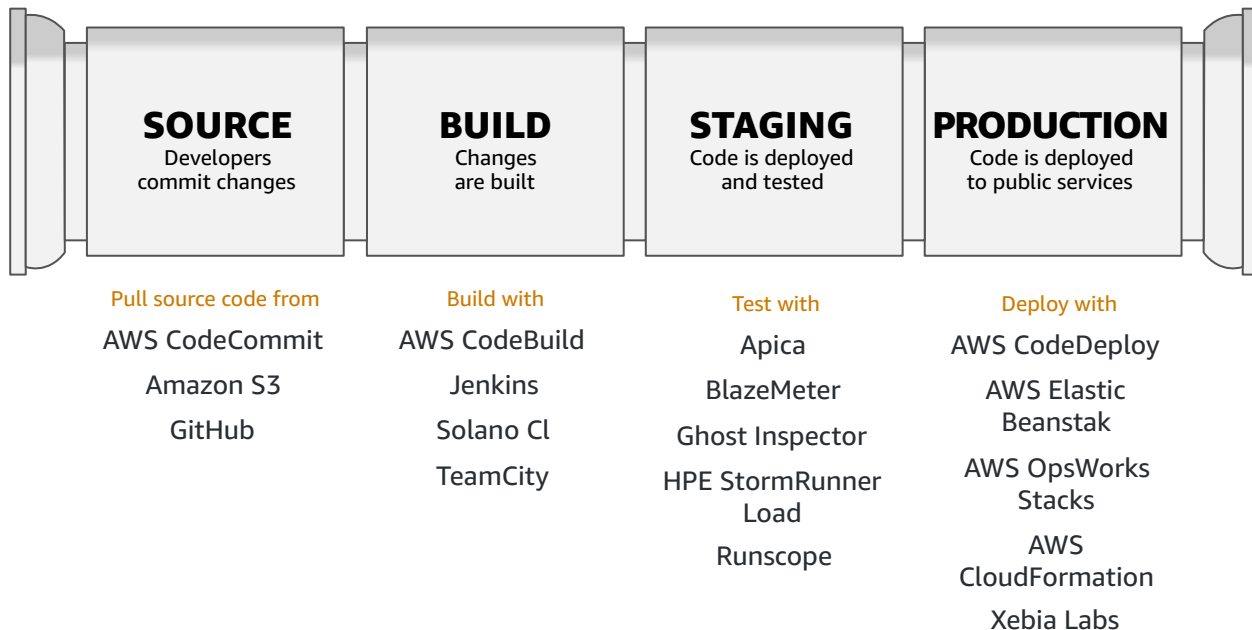
CI/CD service that builds, tests and deploy applications, visual workflow, integrate using custom plugins

# Deployment Pipeline

ON AWS

## AWS CodePipeline

Automated continuous integration and continuous delivery release workflow



# Maven Plugin

```
<groupId>com.github.seanroy</groupId>  
<artifactId>lambda-maven-plugin</artifactId>  
<version>2.2.2</version>
```





---

## References

### **Amazon Lex**

[aws.amazon.com/lex](https://aws.amazon.com/lex)

### **Developer Portal**

[developer.amazon.com](https://developer.amazon.com)

### **Slides & code**

[github.com/arun-gupta/chatbot](https://github.com/arun-gupta/chatbot)