

# 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**

**Testing**

**Availability**

**Business  
Logic**

**Mobile**

**Natural  
Language  
Understanding**

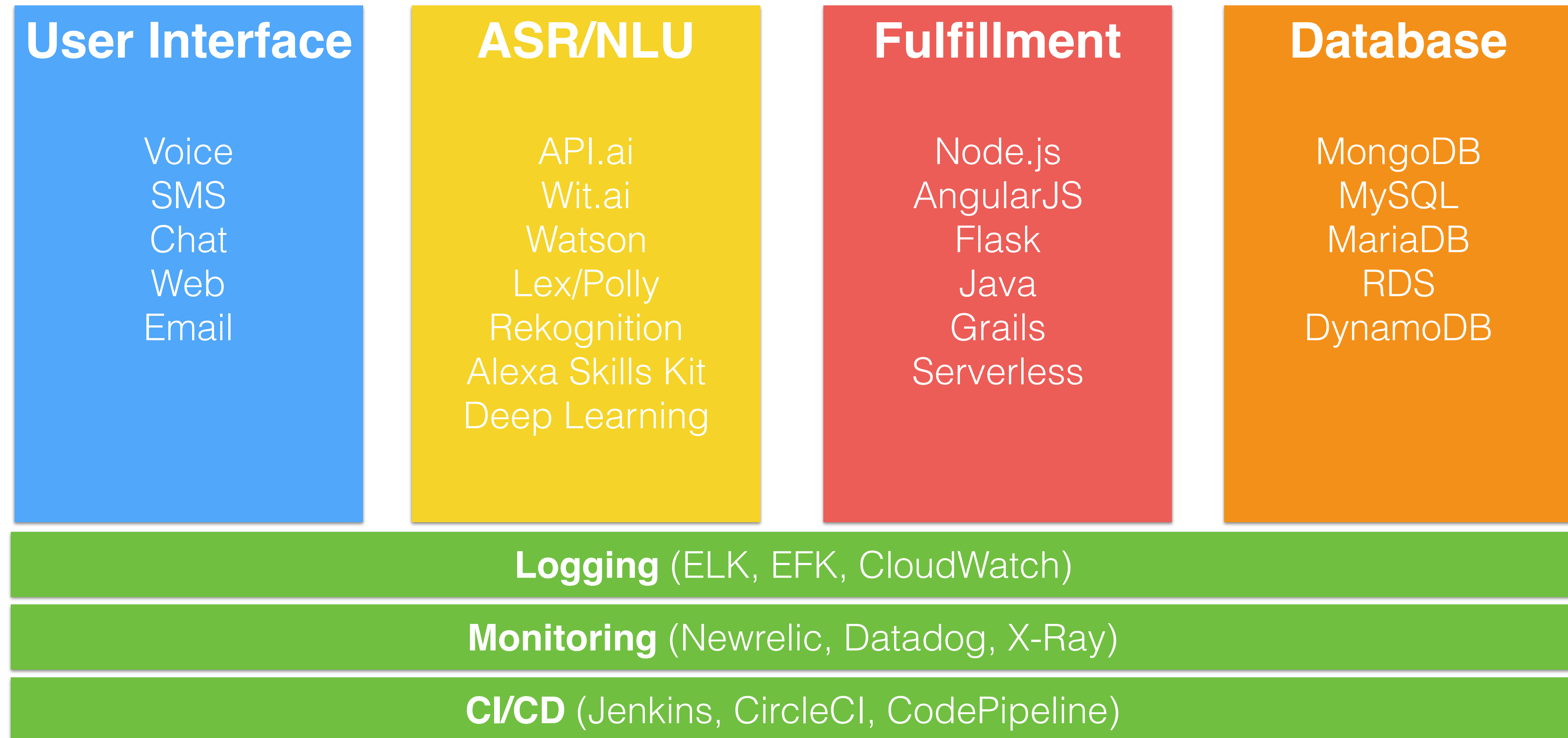
**Scalability**

**Authentication**

**Dialog Manager**

**Disparate  
Systems**

# Chatbot Components



BookHotel

### Intents

A particular goal that the user wants to achieve

I'd like to book a hotel.

### Utterances

Spoken or typed phrases that invoke your intent

Sure, which city?

New York City

### Slots

Data the user must provide to fulfill the intent

What date do you check in?

### Prompts

Questions that ask the user to input data

...

### Fulfillment

The business logic required to fulfill the user's intent



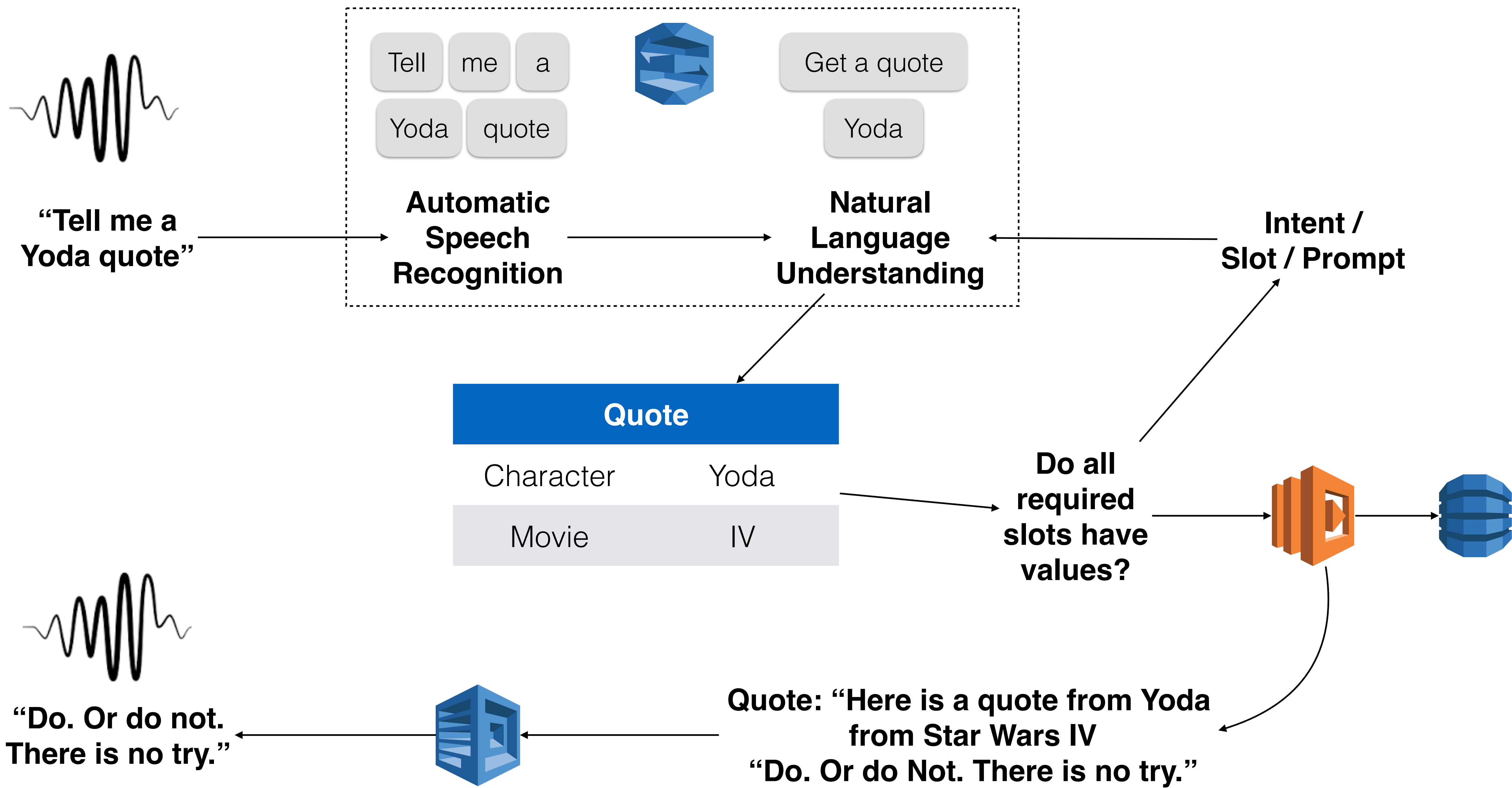
November 30th.



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

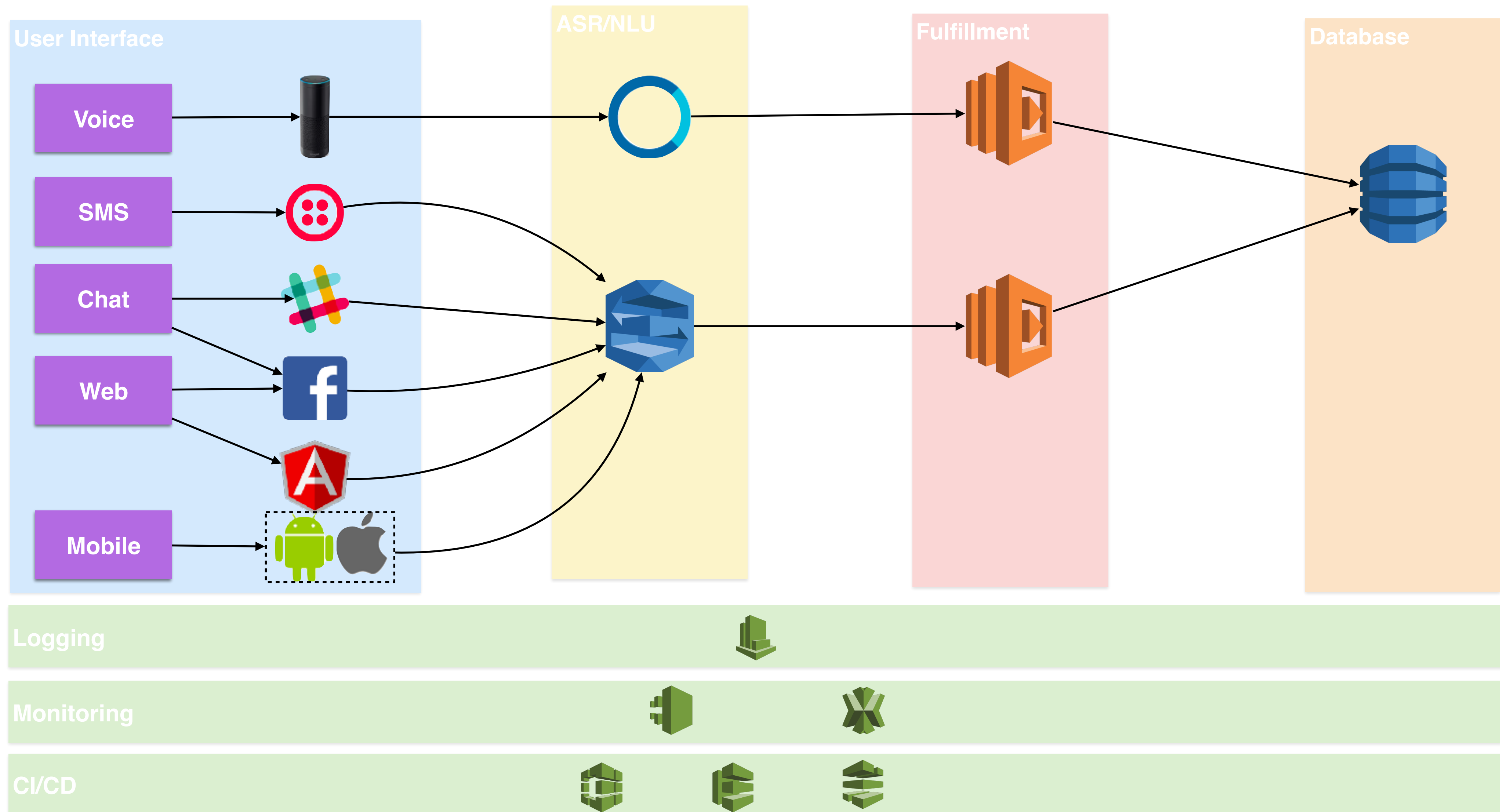
Yes.

Thank you. The reservation went through successfully.





# Chatbot 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

```
{
  "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 .",
  ],
  "weapon": "lightsaber",
  "whoami": "Yoda"
}
```

starwars [Close](#)

Overview Items Metrics Alarms Capacity **Indexes** Triggers Access control Tags

Create index Delete index



	Name	Status	Type	Partition key	Sort key	Attributes	Read capacity	Write capacity	Size
<input type="radio"/>	whoami-index	Active	GSI	whoami (String)	-	ALL	5	5	4,856

# 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 3:31:22 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



# Lex and Twilio/SMS Integration

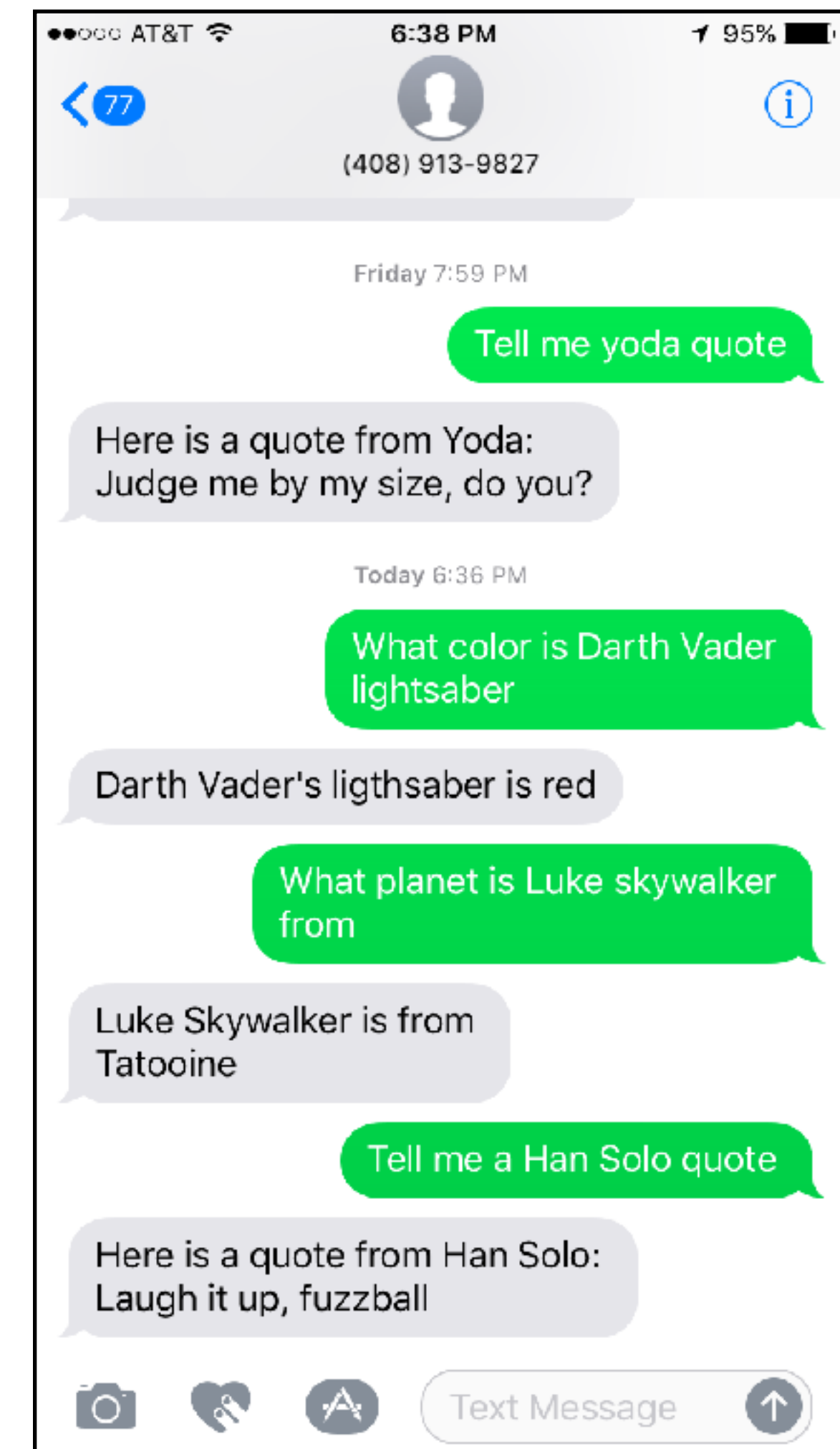
The screenshot shows the AWS Lex console interface for configuring a Twilio SMS channel. The left sidebar lists 'StarWars' as the latest chatbot, with tabs for Editor, Settings, Channels, and Monitoring. The 'Channels' tab is active, showing a list of chatbots on the left: Chatbots, Facebook, Slack, and Twilio SMS (highlighted). The main area contains a form for Twilio SMS configuration with the following fields:

- Name\***:
- Description**:
- IAM Role**: [AWSServiceRoleForLexChannels](#) (Automatically created on your behalf)
- KMS key**:
- Alias\***:
- Account SID\***:
- Authentication Token\***:

Below the form is an **Activate** button. At the bottom, there is a section for **Callback URLs** with a table containing one entry:

Callback URL	Actions
starwars_sms (starwars)	<a href="#">Delete</a>

The **Endpoint URL** is displayed as <https://channels.lex.us-east-1.amazonaws.com/twili...> with a **Copy** button.



# 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

# Lex and Slack Integration

StarWars Latest

Editor Settings Channels Monitoring

Chatbots

Facebook

Slack

Twilio SMS

IAM Role [AWSServiceRoleForLexChatbot](#)  
Automatically created on your behalf

KMS key

Alias\*

Client Id\*

Client Secret\*

Verification Token\*

Success Page URL


Activate

Callback URLs

StarWarsSlack (starwars)

Postback URL: <https://channels.lex.us-east-1.amazonaws.com/starwars-slack>

OAuth URL: <https://channels.lex.us-east-1.amazonaws.com/starwars-slack>




star\_wars\_chatbot

This is the very beginning of your direct message history with star\_wars\_chatbot.


[How does star\\_wars\\_chatbot work?](#)

Today




arungupta 9:14 PM

tell me a yoda quote




Star Wars Chatbot APP 9:14 PM

Here is a quote from Yoda: That is why you fail.




arungupta 9:14 PM

what color is Luke skywalker lightsaber




Star Wars Chatbot APP 9:14 PM

Luke Skywalker's lightsaber is blue



arungupta 9:14 PM

which planet is Darth Vader from



Star Wars Chatbot APP 9:14 PM ☆

Darth Vader is from Tatooine

19

# 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)

# Lex and Facebook Integration

The screenshot displays the AWS Lex console interface for a chatbot named "Star Wars Chatbot". The left sidebar shows the "Channels" tab selected, with "Facebook" highlighted. The main configuration area includes fields for Name, Description, IAM Role (AWS Service Role), KMS key (aws/lex), Alias, Verify Token, Page Access Token, and App Secret Key. An "Activate" button is visible at the bottom of the configuration section. Below the configuration, the "Callback URLs" section shows "StarWarsChatbot (starwars)" with an "Endpoint URL" of "https://channe".

On the right, a preview window shows the chatbot's interactions. The chatbot's profile is "Star Wars Chatbot" with 1 person liking it and a "Movie" category. The chat history shows the following sequence of messages:

- 5:21PM: User input: "tell me a yoda quote"
- 6:27PM: User input: "tell me a yoda quote"
- Bot response: "Here is a quote from Yoda: When nine hundred years old you reach, look as good you will not."
- User input: "what is the color of luke skywalker lightsaber"
- Bot response: "Luke Skywalker's lighthsaber is blue"
- User input: "what planet is darth vader from?"
- Bot response: "Darth Vader is from Tatooine"



# Web Application

# Mobile

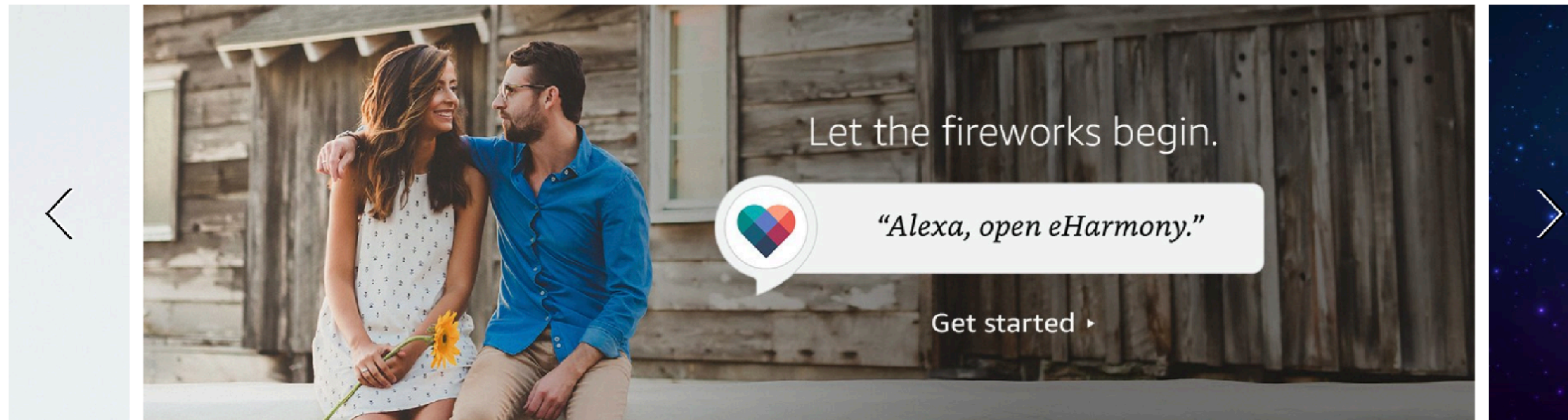
# What is Alexa?

- 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!

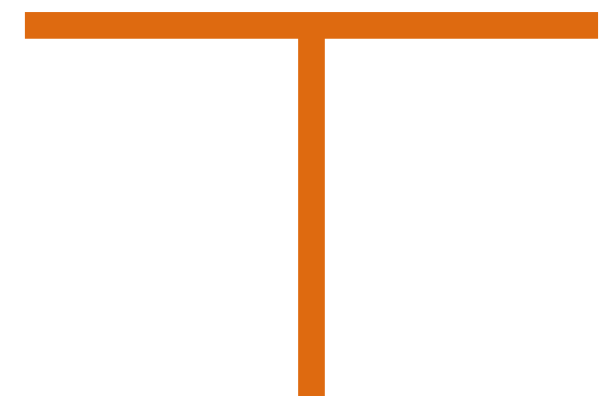


"Alexa, what are your top skills?"

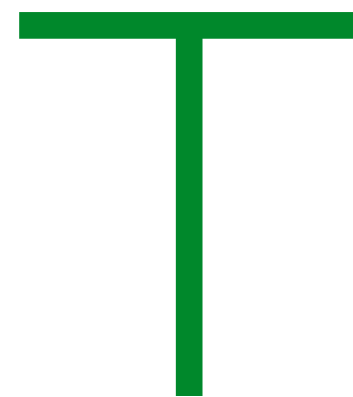
 <p>"Alexa open Sleep Sounds"</p>	 <p>"Alexa, play Classic Vinyl on SiriusXM"</p>	 <p>"Alexa, what's my Flash Briefing?"</p>	 <p>"Alexa, start Song Quiz"</p>	
<p>Sleep and Relaxation Sounds</p> <p>★★★★★ 3,298</p>	<p>SiriusXM</p> <p>★★★★☆ 1,259</p>	<p>Fox News</p> <p>★★★★☆ 595</p>	<p>Song Quiz</p> <p>★★★★★ 652</p>	<p>Jeopardy!</p> <p>★★★★☆</p>

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

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



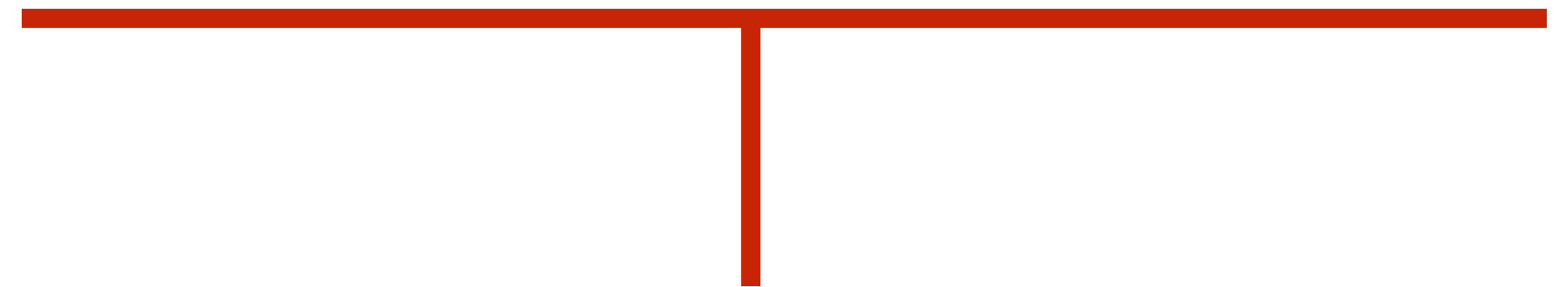
Wake  
word



Starting  
phrase



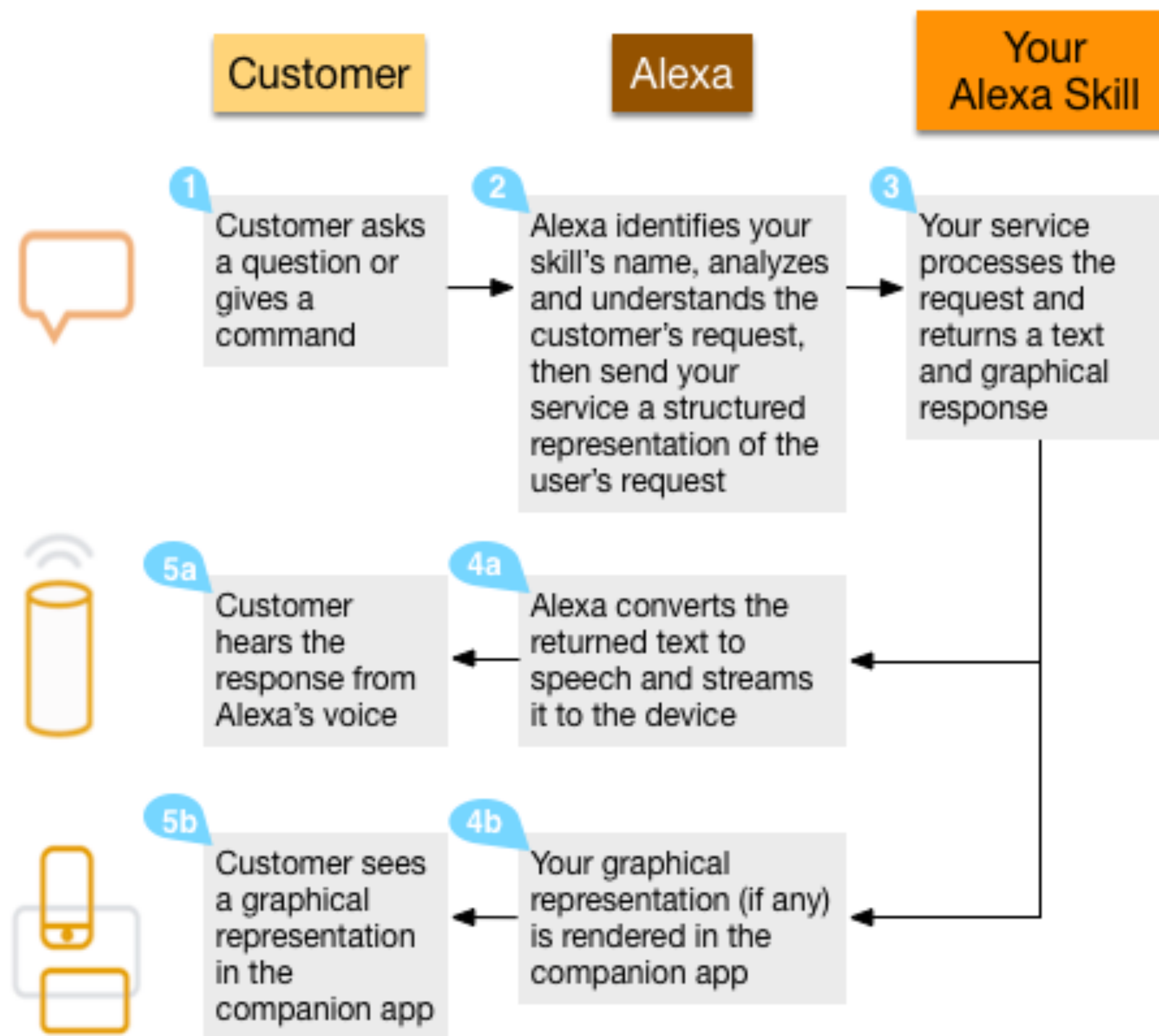
Skill  
invocation  
name



Utterance



NOTE to Alyssa:  
Can you please  
break this image  
into multiple  
images and  
animate them?



# Two sides of Alexa skill



Voice User Interface

+



Programming Logic

# Two sides of Alexa skill



+



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

[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>
```



## Alexa Skill Testing Tool

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

Expect Speech...



Clear

Collapse / Expand

SpeechRecognizer.ExpectSpeech 2017-09-21 01:21:18

```
{
  "directive": {
    "header": {
      "dialogRequestId": "dialogRequestId-18b4f182-d922-4e63-b85d-4f9993f021f0",
      "namespace": "SpeechRecognizer",
      "name": "ExpectSpeech",
      "messageId": "64b4412f-202c-4894-95a5-5a7f8f376dbd"
    },
    "payload": {
      "timeoutInMilliseconds": 8000
    }
  }
}
```

TemplateRuntime.RenderTemplate 2017-09-21 01:21:18

**Welcome to Star Wars Trivia, you  
can ask quotes**

*Star Wars*

Star Wars Welcome

```
{
  "directive": {
    "header": {
      "dialogRequestId": "dialogRequestId-18b4f182-d922-4e63-b85d-4f9993f021f0",
      "namespace": "TemplateRuntime",
      "name": "RenderTemplate",
      "messageId": "8035d9d2-ecf-4e07-b92f-f60e20ea9659"
    },
    "payload": {
```





# Security

Role ARN	arn:aws:iam::091144949931: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 relationshipsAccess AdvisorRevoke 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



Search Log Group

Create Log Stream

Delete Log Stream

Filter:	Log Stream Name Prefix	×
<input type="checkbox"/> Log Streams		Last Event Time
<input type="checkbox"/>	2017/09/21/[\$LATEST]8981e581afb9417d8d89f09b90324ec3	2017-09-20 17:12 UTC-7
<input type="checkbox"/>	2017/09/21/[\$LATEST]f0fa047b8f5d4181ae6d50dc369f7018	2017-09-20 17:08 UTC-7
<input type="checkbox"/>	2017/09/19/[\$LATEST]cbc463340fa248d5b99061d0c9648c42	2017-09-19 13:57 UTC-7
<input type="checkbox"/>	2017/09/12/[\$LATEST]96dc2bf8b7904a1b8685d1b0e3d08959	
<input type="checkbox"/>	2017/09/12/[\$LATEST]827ad8dea94a444c9206f218d0268163	
<input type="checkbox"/>	2017/09/11/[\$LATEST]576b9d3e5c2b41c5a45d4f11d6570258	
<input type="checkbox"/>	2017/09/11/[\$LATEST]a56d79f17ec84e0ca2770893348cea50	
<input type="checkbox"/>	2017/09/11/[\$LATEST]2a1635100f8245e3aff5ac9c73bd9ff7	
<input type="checkbox"/>	2017/09/11/[\$LATEST]6b5ca42c842f44ffb8a7042d7167e39f	
<input type="checkbox"/>	2017/09/09/[\$LATEST]c107c593907642b6994136139dab2c2a	
<input type="checkbox"/>	2017/09/09/[\$LATEST]df2c0fbf61bf4e5392529c3fa854eb81	
<input type="checkbox"/>	2017/09/09/[\$LATEST]beea3b8a5be34868ae39c2df9c46b7a5	
<input type="checkbox"/>	2017/09/08/[\$LATEST]7fd04de005594d5d9e2940971129b75d	
<input type="checkbox"/>	2017/09/08/[\$LATEST]b97bf5fc565d4e43b15ed38e988a5949	
<input type="checkbox"/>	2017/09/08/[\$LATEST]f51c4e28e23c462596a8dab26eb7c255	







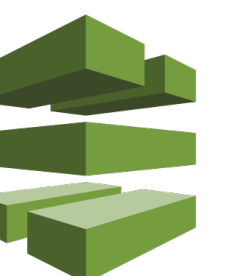
# Analyze & Debug - AWS X-Ray

- 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

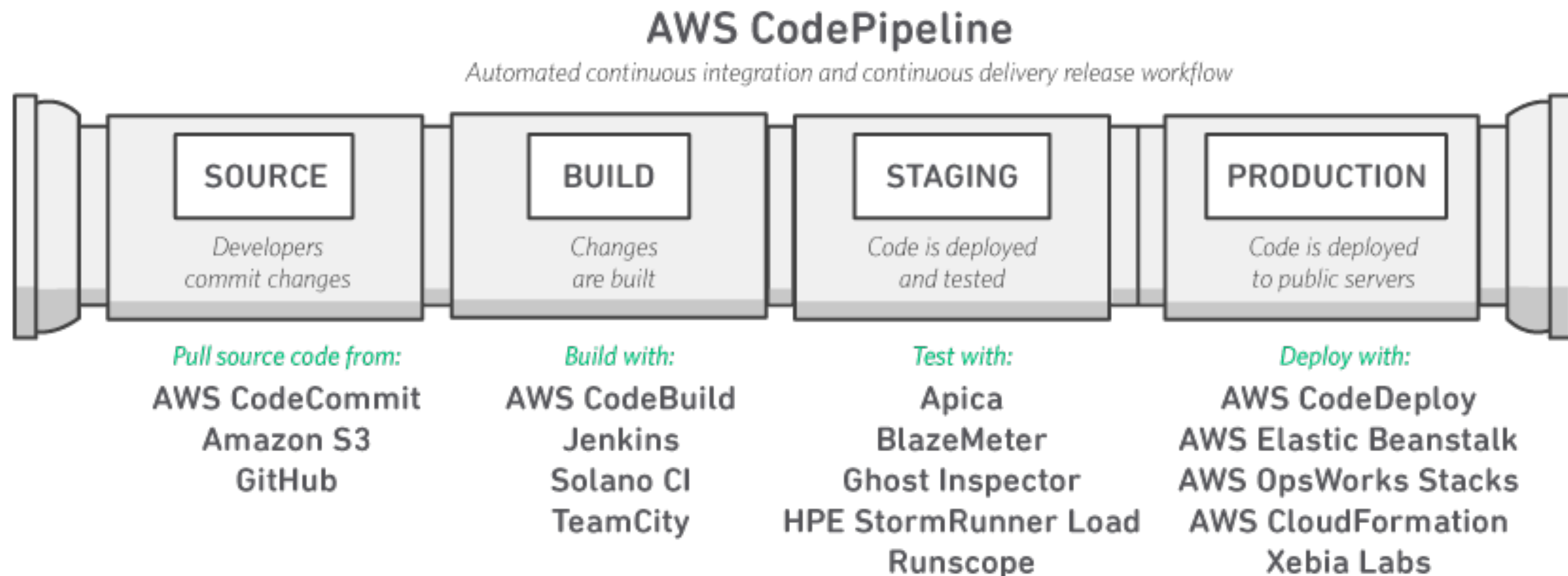


# Add a snapshot of X-Ray dump

# 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



# 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)