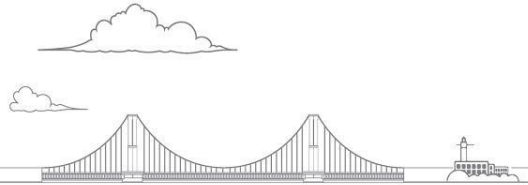
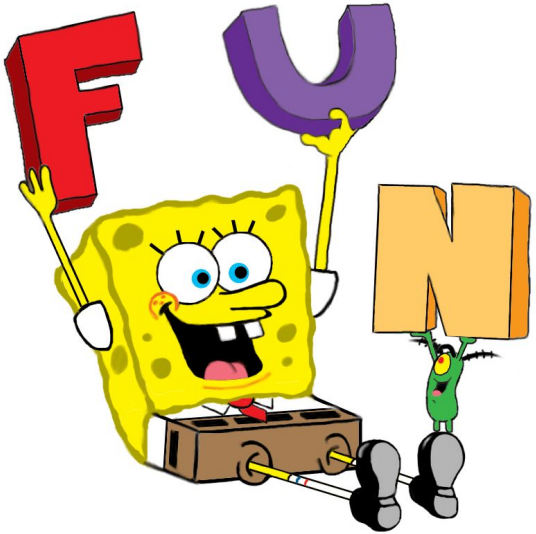
Ask Alexa: How Do I Create My First Alexa Skill?

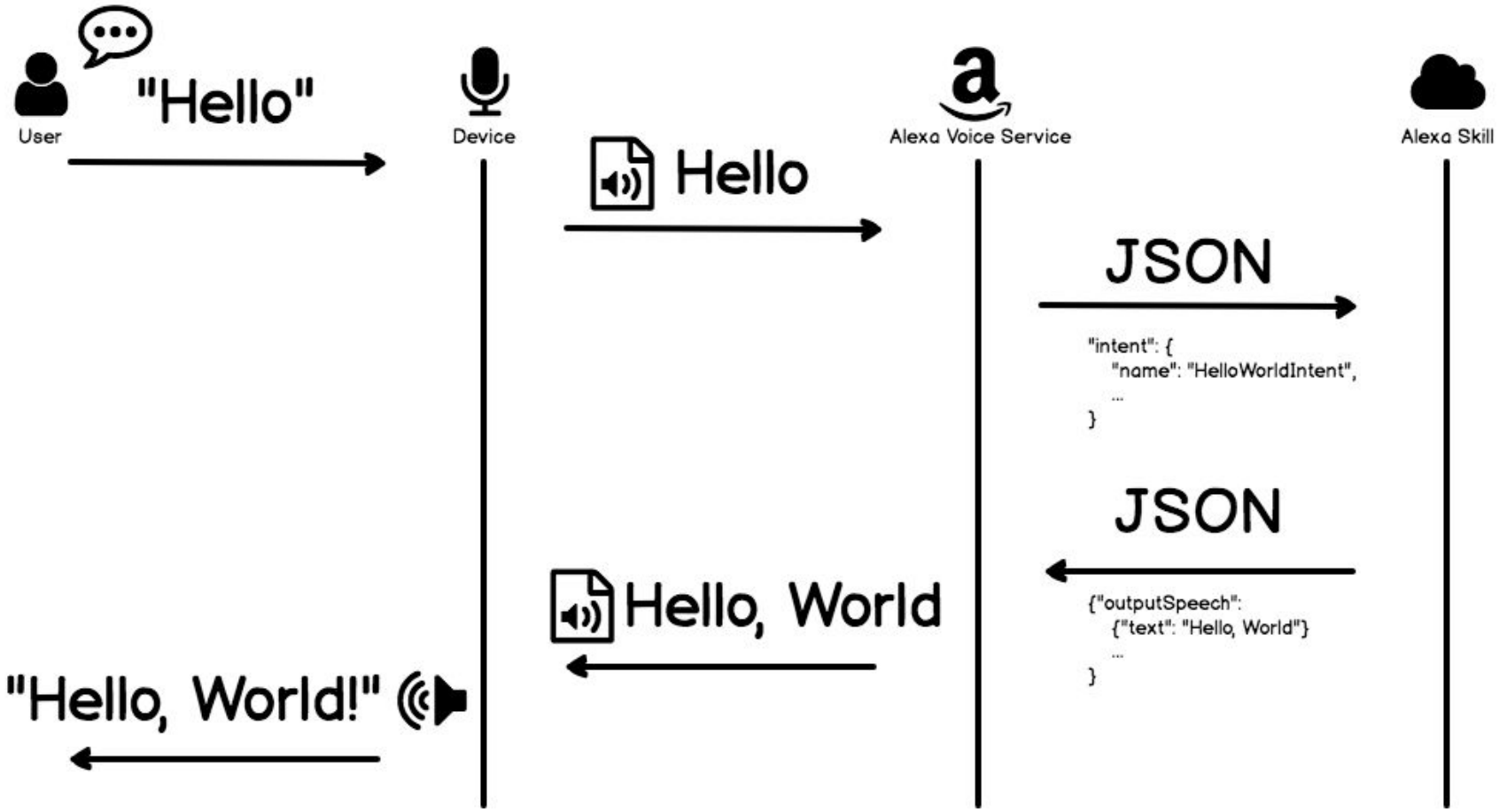
Darlene Wong
Varang Amin




FUN

Hello World

1. Frontend: VUI (Voice User Interface) with Alexa Skills Kit
2. Backend: python™
3. Test
4. Deploy
5. Certification


User
"Hello"
Device
Hello
Alexa Voice Service
Alexa Skill
JSON
"intent": {
"name": "HelloWorldIntent",
...
}
JSON
{"outputSpeech":
{"text": "Hello, World"}
...
}
Hello, World
"Hello, World!"

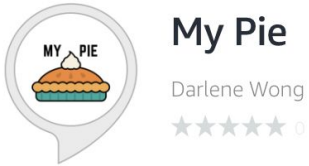
Frontend: Create VUI with Alexa Skills Kit

1. Create account on https://developer.amazon.com
2. Click “Create Skill” from the ASK Developer Console
3. Add your Skill Name and Invocation Name

...

Name That Skill!

Skill Name	● For yourself and other developers working on the skill ● Not shown to the public ○ `hello_world_pybay_2018`
Invocation Name	● What the user says to start your skill ○ “Alexa, open ***hello world***” ● Lowercase, apostrophes, periods, and spaces only
Public Name (Distribution)	● How your skill will be displayed in the Alexa app ○ “Hello World”



Frontend: Create VUI with Alexa Skills Kit

1. Create account on https://developer.amazon.com
2. Click “Create Skill” from the ASK Developer Console
3. Add your Skill Name and Invocation Name
4. Add Intents, sample Utterances, and Slots

...

Interaction Model

- Intents
  - Your skill's actions that the user can request
    - `HelloWorldIntent, HelloNameIntent`
  - Required: `HelpIntent, CancelIntent, StopIntent, FallbackIntent`
- Sample Utterances
  - Phrases the user can say
    - *"Hi", "Hello", "Hello there", "What's up"* ⇒ *"Hello world"*
  - Utterances are mapped to a specific Intent
- Slots
  - An Intent's argument
    - *"Say hello to {name}", "Greet {name}"* ⇒ *"Hello {name}"*

Frontend: Create VUI with Alexa Skills Kit

1. Create account on https://developer.amazon.com
2. Click “Create Skill” from the ASK Developer Console
3. Add your Skill Name and Invocation Name
4. Add Intents, sample Utterances, and Slots
5. Endpoint: Where to send POST requests from the Alexa Skill
   a. HTTPS web service endpoint *or*
   b. AWS Lambda Amazon Resource Name ARN

Backend Options

- Hosting
  - Host on your own web server
  - AWS Lambda
    - Supports C#, Go, Java, JavaScript, and … Python!
- Frameworks
  - Alexa Skills Kit SDK for Python

    https://github.com/alexa-labs/alexa-skills-kit-sdk-for-python

  - flask-ask

    https://github.com/johnwheeler/flask-ask

Setup flask-ask

- Setup virtual environment

```
$ python3 -m venv ~/.virtualenvs/alexa

$ source ~/.virtualenvs/alexa/bin/activate
```

- Install flask-ask

```
(alexa) $ pip install flask-ask
```

- Pin the cryptography library version

```
(alexa) $ pip install cryptography==2.1.4
```

helloworld.py (1/3)

```
import logging
import os

from flask import Flask
from flask_ask import Ask, question, statement

app = Flask(__name__)
ask = Ask(app, "/")
logging.getLogger('flask_ask').setLevel(logging.DEBUG)

@ask.launch
def launch():
    speech_text = 'Welcome to Hello World, say hello, or tell me who to say hello to.'
    return question(speech_text)
```

helloworld.py (2/3)

```
@ask.intent('HelloWorldIntent')
def hello_world():
    speech_text = 'Hello, world!'
    return question(speech_text)

@ask.intent('HelloNameIntent', default={'name': 'World'})
def hello_name(name):
    speech_text = f'Hello, {name}'
    return question(speech_text)

@ask.intent('AMAZON.StopIntent')
def stop():
    stop_text = 'Goodbye world!'
    return statement(stop_text)
```

helloworld.py (3/3)

```
@ask.intent('AMAZON.HelpIntent')
def help():
    speech_text = 'You can say hello to me, or ask me to say hello to someone!'
    return question(speech_text)

@ask.session_ended
def session_ended():
    return "{}", 200

if __name__ == '__main__':
    app.run(debug=True)
```

ngrok

- Gives you a public URL for exposing your local web server

```
$ python helloworld.py
$ ngrok http 5000
```

```
ngrok by @inconshreveable

Session Status                online
Session Expires               6 hours, 14 minutes
Version                       2.2.8
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://24044b6e.ngrok.io -> localhost:5000
Forwarding                    https://24044b6e.ngrok.io -> localhost:5000
```

- Set the skill's URI to the HTTPS ngrok forwarding URL
- Set the skill's SSL certificate type to:

  "My development endpoint is a sub-domain of a domain that has a wildcard certificate from a certificate authority"

Testing Tools

- Alexa Simulator
- Manual JSON
- https://echosim.io/

Zappa

- Deploy serverless, event-driven Python applications
- Deploy to AWS Lambda and AWS API Gateway
- Packages your flask-ask application + dependencies

Zappa

1. Install zappa, awscli, and any other libraries your skill depends on

   ```
   (alexa) $ pip install flask-ask zappa awscli
   ```

2. Create AWS account at https://aws.amazon.com
3. Create an Identity and Access Management (IAM) user https://console.aws.amazon.com/iam/home?#/users$new?step=details
   - Add user name, *e.g.* zappa-deploy
   - Set the Access type to **Programmatic access**
   - Attach the existing **AdministratorAccess** policy for the new user
   - Your Access Key ID and Secret Access Key are shown

Zappa

3. Configure AWS with the new credentials from your virtual environment

```
(alexa) $ aws configure
AWS Access Key ID [None]: YOUR_AWS_ACCESS_KEY
AWS Secret Access Key [None]: YOUR_AWS_SECRET_KEY
Default region name [None]: us-east-1
Default output format [None]:
```

4. Create zappa configuration file by selecting all the default options

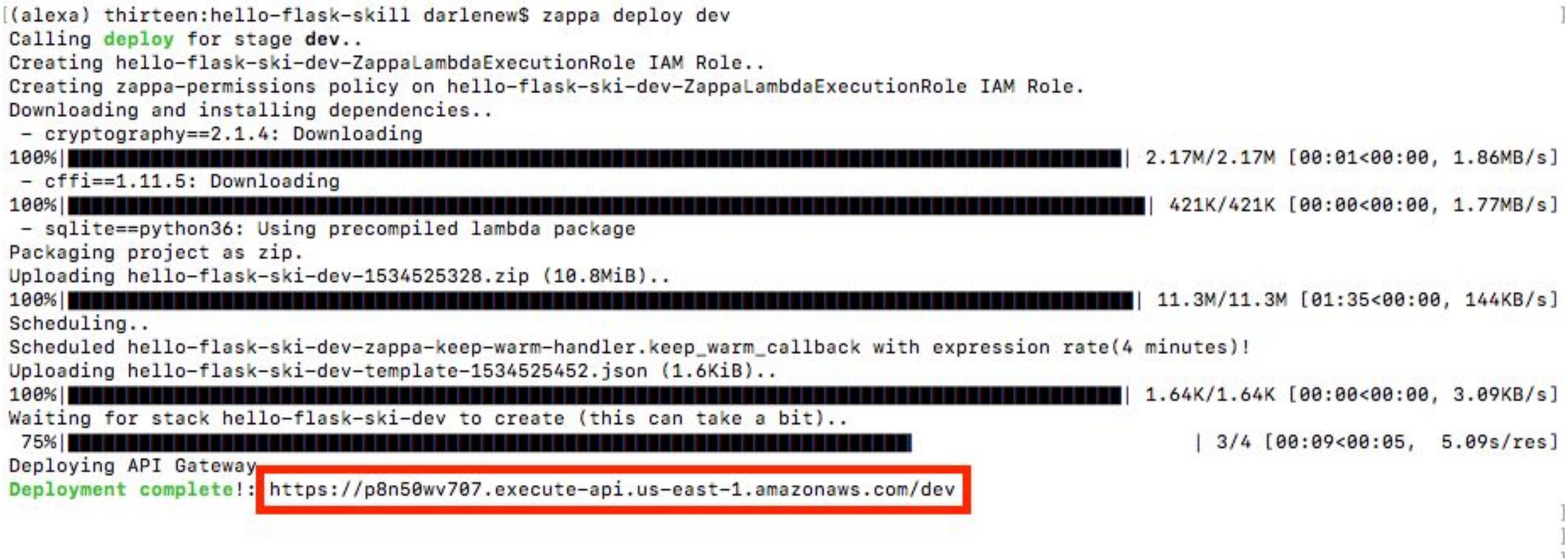
```
(alexa) $ zappa init
```

5. Deploy!

```
(alexa) $ zappa deploy dev
```

6. Deploy future updates

```
(alexa) $ zappa update dev
```



```
[(alexa) thirteen:hello-flask-skill darlenew$ zappa deploy dev
Calling deploy for stage dev..
Creating hello-flask-ski-dev-ZappaLambdaExecutionRole IAM Role..
Creating zappa-permissions policy on hello-flask-ski-dev-ZappaLambdaExecutionRole IAM Role.
Downloading and installing dependencies..
 - cryptography==2.1.4: Downloading
100%|████████████████████████████████████████████████████████████████████████████████████████| 2.17M/2.17M [00:01<00:00, 1.86MB/s]
 - cffi==1.11.5: Downloading
100%|██████████████████████████████████████████████████████████████████████████████████████████| 421K/421K [00:00<00:00, 1.77MB/s]
 - sqlite==python36: Using precompiled lambda package
Packaging project as zip.
Uploading hello-flask-ski-dev-1534525328.zip (10.8MiB)..
100%|█████████████████████████████████████████████████████████████████████████████████████████| 11.3M/11.3M [01:35<00:00, 144KB/s]
Scheduling..
Scheduled hello-flask-ski-dev-zappa-keep-warm-handler.keep_warm_callback with expression rate(4 minutes)!
Uploading hello-flask-ski-dev-template-1534525452.json (1.6KiB)..
100%|████████████████████████████████████████████████████████████████████████████████████████| 1.64K/1.64K [00:00<00:00, 3.09KB/s]
Waiting for stack hello-flask-ski-dev to create (this can take a bit)..
 75%|███████████████████████████████████████████████████████████████████████                        | 3/4 [00:09<00:05,  5.09s/res]
Deploying API Gateway
Deployment complete!: https://p8n50wv707.execute-api.us-east-1.amazonaws.com/dev
```

Certification

- Distribution
  - Public name
  - Short and long descriptions
  - Sample phrases to show the user examples of how the skill is used
  - Small and Large icons
  - Category (e.g. Games, News, Health & Fitness)
  - Privacy & Compliance
  - Testing instructions
- Certification
  - Validation
  - Functional Test
  - Submit for review! It can take a few hours or several days.

https://github.com/darlenew/2018-pybay-alexa

Darlene Wong
Varang Amin