

# Wipro AWS Launchpad | Use Cases

June 10, 2019



Raghuveer Varahagiri





USE CASE #1

# **Virtual Reality / Augmented Reality Walkthrough of AWS Launchpad using Amazon Sumerian**

## 1

# Virtual Reality / Augmented Reality Walkthrough of AWS Launchpad using Amazon Sumerian

## Overview / Objective

Create a VR/AR walkthrough experience of visiting the AWS Launchpad facility. You will visit a virtual lounge that replicates the AWS Launchpad center and the kiosks setup for the various demos. A virtual concierge will accompany you on your visit, provide an overview, and assist you in visiting each of the virtual demo kiosks – which are linked to actual demos running on the web/AWS through HTML3D entities and other AWS services like Lambda. The assistant will interact with you intelligently using natural language and adapt the experience to your preferences.

## Benefits

Demonstrate the Virtual Concierge capability of Amazon Sumerian, and integration with other AWS services. Showcase our AWS capabilities to a wider audience among our clientele across the globe.

## Key AWS Services

- Amazon Sumerian
- Amazon Lex, Amazon Polly, AWS Lambda

## Resources

- AWS Cloud account
- [Optional] Head-Mounted display (HMD) like Samsung Gear VR or Oculus Go
- [Optional] Layout of the AWS Launchpad facility and kiosks
- [Optional] Ability to call / invoke the demos planned for each kiosk over the web, preferably within AWS
- [Optional] Text narration / explanation of the demo to be voiced-over by the virtual assistant





USE CASE #2

# **Real-time Assistant for Text Recognition, Translation, and Read-out**



## 2

# Real-time Assistant for Text Recognition, Language detection & Translation, and Read-out

## Overview / Objective

Create a proof-of-concept application that uses AWS services for text-detection, translation and text-to-speech read out. This can then be packaged into a mobile app.

## Benefits

Users who are visually challenged can use this app to read public signs / product labels or other text matter to immediately make sense of the visual world around them wherever they are without depending on a guide or interpreter.

Tourists / Travelers / Users who do not speak a language can quickly detect a foreign language and have its meaning translated into a language they understand, without needing a translator.

## Key AWS Services

- Amazon Rekognition
- Amazon Polly
- AWS Amplify [Optional]

## Resources

- AWS Cloud account
- Web-Cam enabled laptop for image capture
- [Optional] Raspberry Pi [~\$35] + Raspberry Pi Camera [~\$10]

to demonstrate use on a portable device





USE CASE #3

# **Real-time Assistant for Face/Scene/Object Detection and audible read-out**

## 3

# Real-time Assistant for Face/Scene/Object Detection and audible read-out

## Overview / Objective

Create a proof-of-concept application that uses AWS services for face-detection, scene-detection and text-to-speech read out. This can then be packaged into a mobile app.

## Benefits

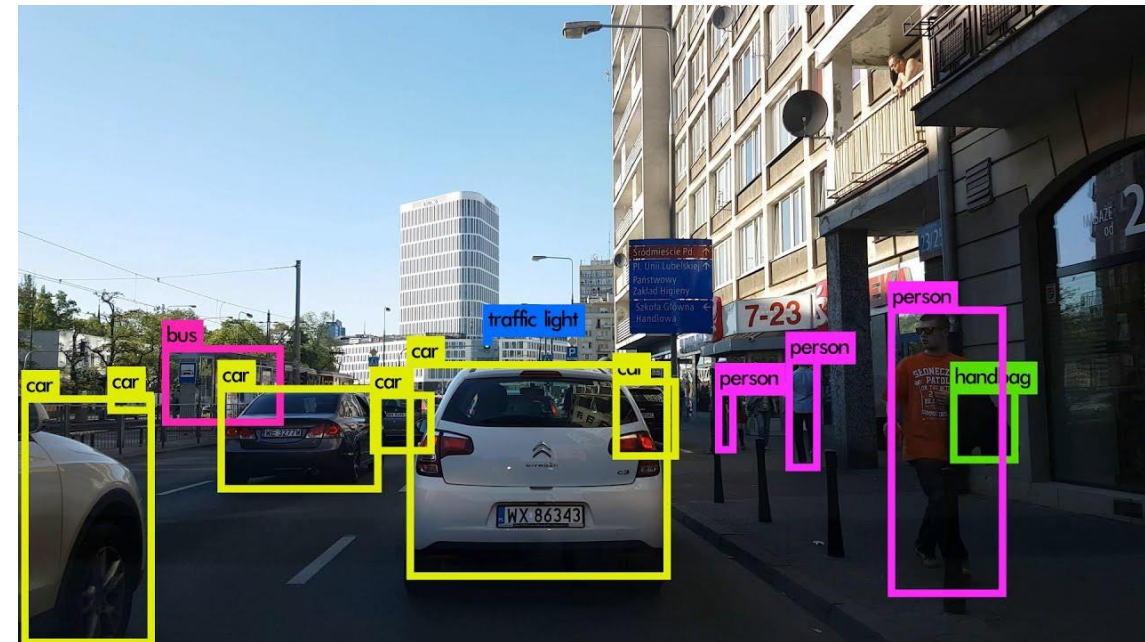
Users who are visually challenged can use this app for face/scene detection to immediately make sense of the visual world around them wherever they are without depending on a guide or interpreter.

## Key AWS Services

- Amazon Rekognition
- Amazon Polly
- AWS Amplify [Optional]

## Resources

- AWS Cloud account
  - Web-Cam enabled laptop for image capture
  - [Optional] Raspberry Pi [~\$35] + Raspberry Pi Camera [~\$10]
- to demonstrate use on a portable device







USE CASE #4

# Deep Learning at the Edge using AWS DeepLens



## 4

# AWS DeepLens: Deep Learning at the Edge

## Overview / Objective

Demonstrate deep learning at the edge using AWS DeepLens device. We will train an Apache MXNet or TensorFlow model and deploy it to the DeepLens device. There after the device can capture images and provide immediate local inference using the ML model

## Sample Use Cases

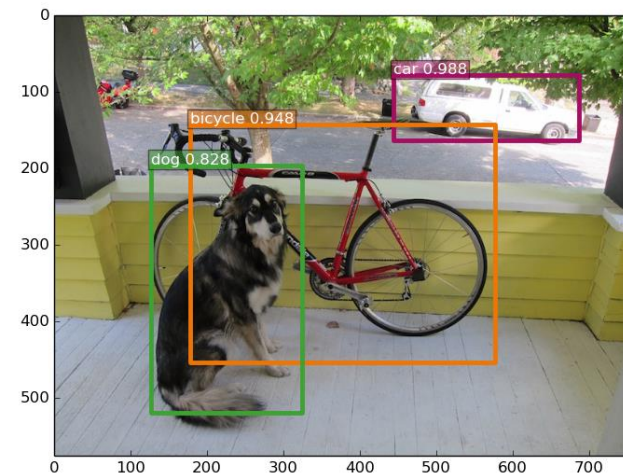
- Object Detection – a pre-trained model that automatically detects a set of objects in the live scene and labels them with a bounding box
- Artistic Style Transfer using a Convolutional Neural Network (CNN) - for example this could be used as a photo booth attraction at an amusement park or museum that automatically creates photos/videos of subjects in the style of a popular painting

## Key AWS Services

- Amazon SageMaker

## Resources

- AWS Cloud account
- AWS DeepLens - Deep learning enabled video camera [[\\$249](#)]





USE CASE #5

# **Vehicle Type and Number Plate detection for Toll Gates and Traffic Management**



## 5

# Vehicle Type and Number Plate detection for Toll Gates and Traffic Management

## Overview / Objective

We will train an Apache MXNet ResNet model to detect vehicle types and read number plates automatically. We can deploy it to a DeepLens device or run it off a laptop or Raspberry Pi.

## Sample Use Cases

- Automatically detect and record the vehicle details that pass through a particular toll gate
- Provide summary statistics on daily / weekly vehicle volume by vehicle type
- Provide summary or weekly / monthly toll charges by number plate

## Key AWS Services

- Amazon SageMaker

## Resources

- AWS Cloud account
  - Laptop equipped with web-cam for image capture
  - [Optional] AWS DeepLens - video camera [[\\$249](#)]
  - [Optional] Raspberry Pi [~\$35] + Raspberry Pi Camera [~\$10]
- to demonstrate use on a portable device





USE CASE #6

# **AWS DeepRacer: Machine Learning**



## 6

# AWS DeepRacer: Reinforcement Learning

## Overview / Objective

Train a Reinforcement Learning model deployed to the AWS DeepRacer scale model car on a printed race track. Using more than one cars, we can demonstrate the variations of different RL models and the effects of hyper-parameter tuning and fitting. We could also conduct a real world or virtual race league to encourage employees to take part in RL challenges.

## Benefits

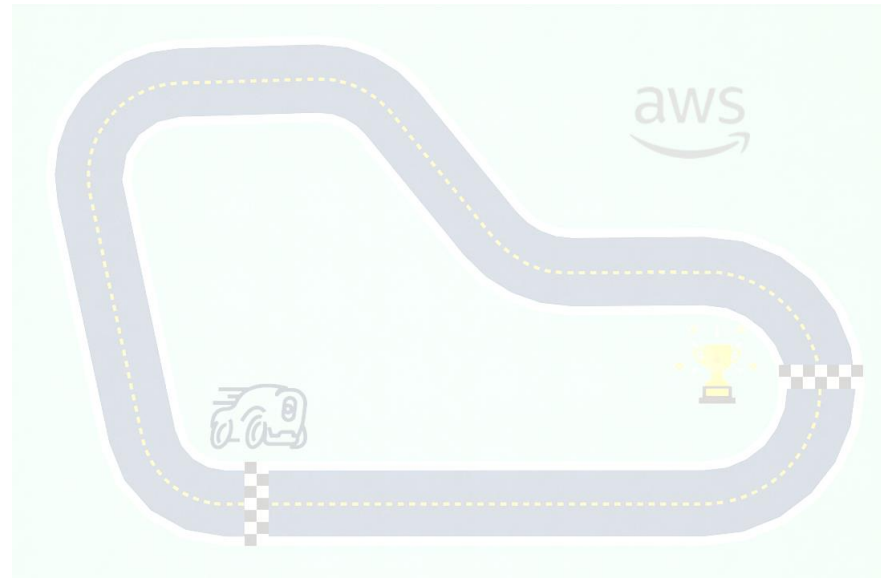
Demonstrate Reinforcement Learning using physical world inputs using a model deployed to the edge. Encourage more employees to take a dive into hands on ML implementations using AWS services like SageMaker that make it very easy to get started and show results very quickly.

## Key AWS Services

- Amazon SageMaker RL
- AWS Lambda, Dynamo DB

## Resources

- AWS Cloud account
- AWS DeepRacer 1/18<sup>th</sup> Scale Model Race Car [[\\$399](#)]
- [Optional] Printed Racetrack Carpet [[Link1](#), [Link2](#)]





USE CASE #7

# **AWS IoT Demo using Raspberry Pi**



# AWS IoT Demo using Raspberry Pi

## Overview / Objective

Raspberry Pi is a low cost (\$35) hobby computer that runs Linux. It can be used to demonstrate how an Internet-of-Things (IoT) device can work in the field. We will register the Raspberry Pi as an IoT device with AWS IoT and use Python code running on the Raspberry Pi to generate and send MQTT messages to AWS IoT. We can add some simple sensors to the Raspberry Pi – for example temperature and humidity sensors – to collect real time data from the field and send it to AWS.

## Benefits

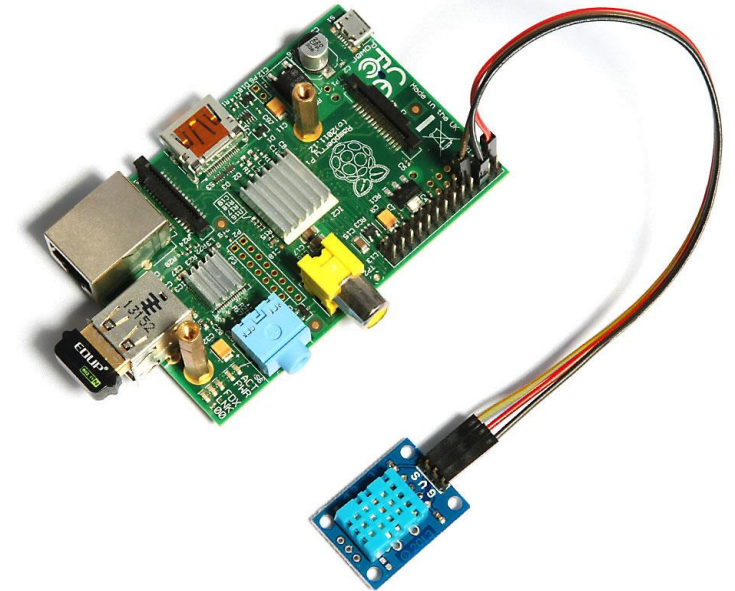
Demonstrate how AWS IoT works with the help of a physical device. Show real-world measurements like temperature, humidity, light, proximity, etc. Show how easy it is to setup and deploy IOT devices to the field and to manage them.

## Key AWS Services

- Amazon IoT

## Resources

- AWS Cloud account
- Raspberry Pi [~\$35] + Sensors [~ \$5 ]





USE CASE #8

# **AWS IoT Greengrass Demo using multiple Raspberry Pi devices**



## 8

# AWS IoT Greengrass Demo using multiple Raspberry Pi devices

## Overview / Objective

AWS Greengrass is software that lets you run local compute, messaging, data caching, sync, and ML inference capabilities for connected devices in a secure way. With AWS Greengrass, connected devices can run AWS Lambda functions, keep device data in sync, and communicate with other devices securely—even when not connected to the Internet.

A Raspberry Pi will be setup as an Internet-of-Things (IoT) Core device that communicates with the AWS Cloud. Another Raspberry Pi (say a Raspberry Pi zero W) will be setup as a sensor device that collects and sends data to the core device over the local network using MQTT even without the need for internet access.

## Benefits

Demonstrate how AWS IoT Greengrass works with the help of a physical device.

## Key AWS Services

- Amazon IoT, Greengrass

## Resources

- AWS Cloud account
- Raspberry Pi [~\$35]
- Raspberry Pi Zero W [~ \$5 ]
- Sensors [ ~ \$5 ]



Raspberry Pi 3B+



Raspberry Pi Zero W



USE CASE #9

# **Hybrid Cloud Demo: using AWS Systems Manager to manage on-prem Raspberry Pi devices**



## 9

# Hybrid Cloud Demo: using AWS Systems Manager to manage on-prem Raspberry Pi devices

## Overview / Objective

AWS Systems Manager allows one to manage virtual machines and servers not just on the AWS EC2 fleet but also on-prem servers in a hybrid cloud environment. We will demonstrate that capability by setting up one or more "on-prem" servers using Raspberry Pi devices (which are low powered computers running Linux OS)

## Benefits

Demonstrate how hybrid cloud environments can be managed using AWS System Managers using a single pane of glass

## Key AWS Services

- Amazon EC2 Systems Manager
- Amazon CloudWatch

## Resources

- AWS Cloud account
- Raspberry Pi [~\$35] or

A cluster of Raspberry Pi Zero W [~ \$5 each ]

