Test Project

Cloud Computing

Day one

Submitted by: Qiang Wang CN

WSC2022SE_TP53_Day1_acutal_en

Contents

Contents

This Test Project consists of the following documentation/files:

Description of project and tasks

This module is 7.5 hours (until 16:30) - **Day 1: IoT + AI + Web service**

You are a Cloud Engineer working at Unicorn Company which delivers Enterprise AI service. Recently, your company plan to build a new AI service which aimed to analyze short messages that collecting from IoT devices to detect the emotions hidden behind the messages. Then share the result to public by hosting a web service.

You are assigned as solutions architect for this project; Your responsibility is to design and build the technical solution. The IoT devices are managed by a partner company, so you don't need to worry about the configuration of these devices. But you have to build the rest of stuffs on a Public Cloud Platform as per business requirement.

The IoT devices will keep sending messages to you from the start time of the event. You will get points if you can successfully receive these messages. Then you will need to analyze these messages by utilizing AI service and store the result in a NoSql database. In the end, you need to build a web application to serve the result in NoSql database to customers.

As a solutions architect, you may be required to perform many different types of tasks. This challenge will gauge your ability to change tasks and respond to ambiguity.

Background

Today's business world is changing with the adoption of IoT (Internet of Things). IoT is helping in prominently capturing a tremendous amount of data from multiple sources. However, wrapping around the multitude of data coming from countless of IoT devices, makes it complex to collect, process, and analyze the data.

Realizing the future and full potential of IoT devices will require an investment in new technologies. The convergence of AI (Artificial Intelligence) and IoT can redefine the way industries, business, and economies functions. Unicorn Company value this project very much.

Out of security concerns, you can use any existing IAM user or role to handle the challenges. But please don't create any other IAM user/role from IAM console during the challenge.

According the feedbacks from operation team, using t2.micro instances are enough to run Unicorn Service reliably, using instance beyond t2.micro or other instance family may increase cost significantly. But you need to monitor the traffic closely in case of any unexpected traffic and make sure Unicorn Service is available.

Initial state

Your fist task would be finding a way to receive the messages sending from IoT devices.

The IoT devices will keep sending messages to you from the start time of the event. As soon as you find a way to receive these messages, you will start to get points.

Re-platform schedule

Phase I

- Find a way to receive the messages sending from vendor IoT devices, you may need to consider the following AWS Services: IoT Core, S3 etc. You need to provide S3 bucket name and IoT Core endpoint to GameDay Dashboard.

Phase II

- Detect the sentiment from the messages, store the analytic result to NoSql database. You may need to consider the following AWS Services: DynamoDB, Lambda, Comprehend etc.

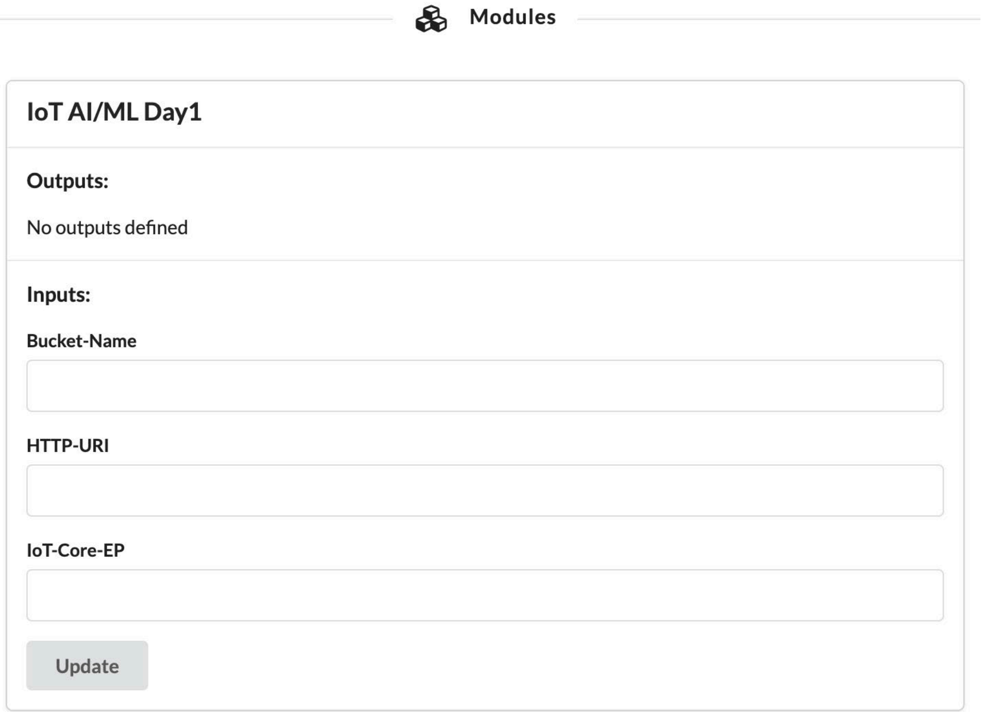
Phase III

- Serve the result by web Service, you need to provide http URL to GameDay Dashboard.

Tasks

1. Log into Gameday with your assigned hash (Provided on the day)
2. Set your team/competitors' name on the Dashboard – (Format: NAME SURNAME)
3. Read the documentation thoroughly (Outlined below)
4. Log into the Amazon Web Services console (link provided from the Dashboard)
5. Connect IoT device via IoT Core service. Store keys and certifications to a S3 bucket, provide bucket name and IoT Core endpoint to GameDay Dashboard.
6. Detect the sentiment from the messages via Comprehend, store the analytic result to DynamoDB database.
7. Build a web service to serve the data in the DynamoDB. Provide http URL to GameDay Dashboard.
8. Monitor performance of the web servers in the "Score Events and Scoreboard" and through the Amazon Web Services Console with CloudWatch.
9. Gain points, reference the "Score Events and Scoreboard" to ensure you are scoring positively by serving the requests.
10. Monitor costs and do not scale up the infrastructure excessively to minimize penalties.
11. This section of the competition will continue for 7.5 hours.

Submit your answer



Modules

IoT AI/ML Day1

Outputs:

No outputs defined

Inputs:

Bucket-Name

HTTP-URI

IoT-Core-EP

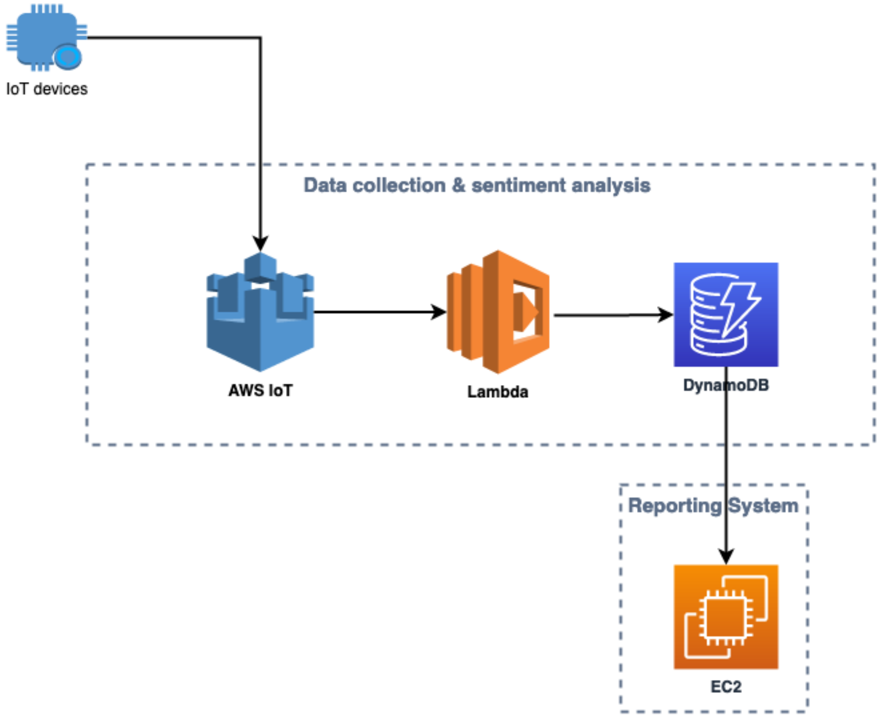
Bucket-Name: The bucket you created which would be used to store your keys and certificates.

HTTP-URI: URL for the web service you built

IoT-Core-EP: Your IoT Core endpoint

Technical Details

1. The server application is deployed as a Go binary compiled from source. Do not alter the binary or use non official binary in any way as that will be grounds for disqualification or leads to penalty. (The download url of the binary file could be found in the readme file in your dashboard)
2. For this day of competition focus on EC2 as your computing resource. Do not use ECS or Fargate or EKS.
3. The server application can handle about several connections before starting to get really slow. Be careful about overloading and watch for HTTP 503 responses from the server when the queue fills.
4. The server application is an x86 statically linked, unstripped ELF executable provided before game started. The application binary can be found in Readme file of this game event.
5. The base OS that has been chosen is **Amazon Linux 2** (https://aws.amazon.com/amazon-linux-ami/). This distribution was selected for its broad industry support, stability, availability of support and excellent integration with AWS.
6. Use of the AWS CLI (Command Line Interface) can be very helpful. Information on installing this tool locally can be found at http://docs.aws.amazon.com/cli/latest/userguide/installing.html. The AWS CLI is already installed on Amazon Linux EC2 instances.



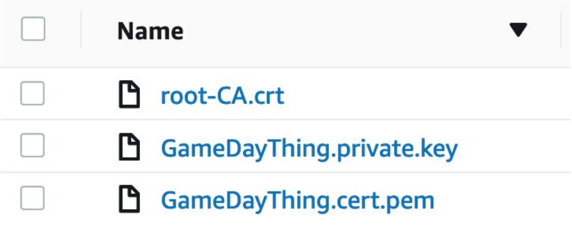
The above example illustrates one simple architectural design for the deployment of the application. This shows some segments needed to serve requests. As an architect of Unicorn company, you have the responsibility to design a modern architecture for Unicorn Service by leveraging powerful cloud services.

Service Details

IoT core

TO SUCCESSFULLY RECEIVING THE MESSAGE FROM IOT DEVICES, YOU NEED TO GENERATE AND UPLOAD FOLLOWING THREE KEYS OR CERTIFICATIONS TO A S3 BUCKET. (PLEASE ENSURE THE FILE NAME ARE CORRECT)

1. GameDayThing.cert.pem
2. GameDayThing.private.key
3. root-CA.crt



□	Name ▼
□	root-CA.crt
□	GameDayThing.private.key
□	GameDayThing.cert.pem

Database

You will need to deploy a DynamoDB server that is highly available and centralized to store the analytic result. The table name should be **unicorn**, structure would be like following:

Key_Name	**Key_Description**	**Comment**
id	Partition Key	
sentiment	Sort Key	The result after sentiment analysis
message		The messages received from IoT devices

Web SeRver

The development team has built a binary file server-stub for you to host the web service, the binary file will serve the data which you stored in the DynamoDB to customer. You need to consider how to build a high available and scalable web service with lower cost for customer.

Gentle reminder

1. Messages are sending from IoT devices in a circular manner, so no need to worry about missing any messages at beginning of the event. You still have many chances to store all the messages from IoT devices in a full cycle. The topic is ‘sdk/test/Python’.
2. After One hour of the event, result searching request will be sent to your web service to fetch out the data in your DynamoDB table. You will get points if you store the right answer inside.
3. In summary, you can gain points in case any of below get configured correctly:

- Message sent from IoT devices can be successfully received.
- Your web service is up and reachable.
- Your web service can response correct answer for the customer query requests.
- More items get configured correctly, more points you will gain.

4. Please attach TeamRole to your lambda function as the execution role.
5. Please attach EC2Role as ec2-prole for your EC2 instance.
6. The files you might need to use could be downloaded in the readme file of your game dashboard.