BikeSecure Cloud Setup

Login to AWS

1. Login to your AWS root account with your email and password.

Create DynamoDB

1. Open the DynamoDB console, choose **Create table** to create our first table
2. In the **Create DynamoDB table** screen, do the following:
    a. On the **Table name** box, enter Stands
    b. For the **Primary key**, do the following:
        - In the **Partition key** box, enter Rack ID. Set the datatype to **String**
        - Choose **Add sort key**, enter Stand ID. Set the datatype to **String**
    c. Choose **Create** to create the table
3. Create 2nd table called Racks, choose **Create table** again
4. In the **Create DynamoDB table** screen, do the following:
    a. On the **Table name** box, enter Racks
    b. In the **Partition key** box, enter Stand ID. Set the datatype to String
    c. Choose **Create** to create the table
5. Now you will see two tables in the tables list. Follow Stands.csv to create the items in Stands table and follow Racks.csv to create the items in Racks table. Make sure the datatype is correct for each attribute and the data is exactly the same. (Note: make sure there is no space between Latitude and Longitude values for Location attributes)

Create a thing in IoT Core

1. In the AWS IoT console, in the navigation bar, choose **Manage** and choose **Things**
2. If you do not have any things yet, choose **Register a thing**. Otherwise, choose **Create**.
3. Choose **Create a single thing**, name your thing as ESP32, click on Next
4. Choose **Create certificate**, download private key, public key, certificate for the thing, and Amazon Root CA 1.
5. Click on **Activate** and click on Done.
6. In the navigation bar, choose **Secure** and choose **Policies**
7. Click on Create, name the policy as esp32bikepolicy
8. Click on **Advanced mode**, replace all the code by the code below

```
{
 "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Connect",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": "*"
    }
  ]
}
```

9. Click on Create to create the policy
10. Go the ESP32 thing we created just now, choose **Security**, click on the **Certificate**
11. Choose **Policies**, click on **Actions**, choose Attach policy
12. Choose esp32bikepolicy, click on Attach

Create IAM roles

We need to create two IAM roles, first one is called lambda-dynamodb-fullaccess-role.

1. Open IAM console, in the navigation pane, choose **Roles**, and then choose **Create role**
2. Choose use case as **Lambda**, and click on **Next:Permissions**
3. Select AmazonDynamoDBFullAccess, and click on **Next: Tags**
4. Click on **Next: Review**, give Role name as lambda-dynamodb-fullaccess-role
5. Click on **Create role** to create the role

Now we will create second IAM role called readdynamodb

1. In IAM console, choose **Roles**, and then choose **Create role**
2. Choose use case as **Lambda**, and click on **Next: Permissions**
3. Select AmazonDynamoDBReadOnlyAccess, and click on **Next: Tags**
4. Click on **Next: Review**, give Role name as readdynamodb
5. Click on **Create role** to create the role

Create generatePassword Lambda Function

1. Open the Lambda Functions page, choose **Create function**
2. Give **Function name** as generatePassword, choose **Runtime** as Python 3.8
3. Click on **Change default execution role,** select Use an existing role
4. Click on the box, in the dropdown lists, choose *lambda-dynamodb-fullaccess-role*

5. Click **Create function**, you will be directed to the generatePassword lambda function page you have just created
6. Under the **Code** tab, open the **lambda_function.py** file, replace all the code by the code in cloudSetup/generatePassword.py file. Change YOUR_REGION to your account region and change YOUR_ACCOUNT_ID to your account ID.
7. Create process_request.py under the same folder as lambda_function.py, copy all the code in cloudSetup/process_request.py, and paste it to process_request.py you just created
8. Create update_rackstable.py under the same folder as lambda_function.py, copy all the code in cloudSetup/update_rackstable.py, and paste it to update_rackstable.py you just created
9. Save all files and click on **Deploy**

Create readdb Lambda Function

1. Open the Lambda Functions page, choose **Create function**
2. Give **Function name** as readdb, choose **Runtime** as Python 3.8
3. Click on **Change default execution role,** select Use an existing role
4. Click on the box, in the dropdown lists, choose *readdynamodb*
5. Click **Create function**, you will be directed to the readdb lambda function page you have just created
6. Under the **Code** tab, open the **lambda_function.py** file, replace all the code by the code in cloudSetup/readdb.py file
7. Save all files and click on **Deploy**

Create iotfunction Lambda Function

1. Open the Lambda Functions page, choose **Create function**
2. Give **Function name** as iotfunction, choose **Runtime** as Python 3.8
3. Click **Create function**, you will be directed to the iotfunction lambda function page you have just created
4. Under the **Code** tab, open the **lambda_function.py** file, replace all the code by all the code in cloudSetup/iotfunction.py file. Change YOUR_REGION to your account region.
5. Save all files and click on **Deploy**
6. Under the **Configuration** tab, choose **Permissions**, click on role name
7. On the new window, choose **Permissions** tab, click on **Attach Policy**
8. Select AWSIoTDataAccess, click on **Attach policy**
9.

Create invokeiotfunction policy

1. Open IAM console, in the navigation pane, choose **Policies**, and then choose **Create policy**
2. Choose **JSON**, replace the code in the box by the following code. Change YOUR_REGION to your account region, and change YOUR_ACCOUNT_ID to your account ID.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction",
        "lambda:InvokeAsync"
      ],
      "Resource": "arn:aws:lambda:YOUR_REGION:YOUR_ACCOUNT_ID:function:iotfunction"
    }
  ]
}
```

3. Click on **Next: Tags**, click on **Next: Review**
4. Give name as invokeiotfunction, click on **Create policy**

Connect two lambda functions

1. Open the Lambda Functions page, choose **Functions**, choose generatePassword function
2. Under **Configuration** tab, choose **Permissions**, click on role name
3. In the new window, choose **Permissions** tab, click on **Attach policies**
4. Select invokeiotfunction, click on **Attach policy**

ESP32 Set-up

1. Install Arduino IDE
2. Install espressif/arduino-esp32 core
    a. Open Arduino Software (IDE), installed on your local system. Go to File in the toolbar and select Preferences.
    b. Next, enter https://dl.espressif.com/dl/package_esp32_index.json into the “Additional Board Manager URLs” field as shown in the figure below.
    c. Now select Boards Manager in the toolbar. Go to Tools > Board > Boards Manager.
    d. Here Search for ESP32 and press install button.
    e. After that, you will able to see a progress bar below (It may take a few seconds to several minutes depending on internet speed). Once installation completed you be able to see installed next to System.
3. Install relevant libraries
    a. Aws-iot library.
        i. Go to https://github.com/ExploreEmbedded/Hornbill-Examples/archive/master.zip?ref=hackernoon.com and download the AWS_IOT Hornbill library.

ii. Exact the Hornbill-Examples-master.zip & go-to Hornbill-Examples-master\arduino-esp32, copy AWS_IOT and paste it to C:\Users\{Your User Name}\Documents\Arduino\libraries.

iii. Go to the following directory, C:\Users\YourUsername\Documents\Arduino\libraries\AWS_IOT\src, open the file aws_iot_certficates.c in an editor. Here we need to attach the Thing Certificate and Private Key which we have downloaded in the previous section along with the CA Certificate.

b. ArduinoJson (by Benoit Blanchon)

i. We need to add a library in Arduino IDE, go to Tools → Manage Libraries. Search for ArduinoJson (by Benoit Blanchon), select the latest 6.x.x version only and click install.

c. ESP32Servo

i. We need to add a library in Arduino IDE, go to Tools → Manage Libraries.

4. Update code accordingly

a. Paste the iot code into the Arduino

b. Update accordingly the IoT Endpoints Address

i. retrieve from IoT Dashboard, settings (check sidebar)

c. Replace the ThingName with the actual name

d. Select the board and the port Number

5. Set up the servo motor

a. The servo motor has three wires:

i. Orange control signal connects to pin 32 of esp32

ii. Red power signal connects to 5v pin of esp32

iii. Brown/black ground wire connects to ground pin of esp32

6. Compile and Upload the Code