CMPE 296A – Internet of Things: Infrastructure and Applications
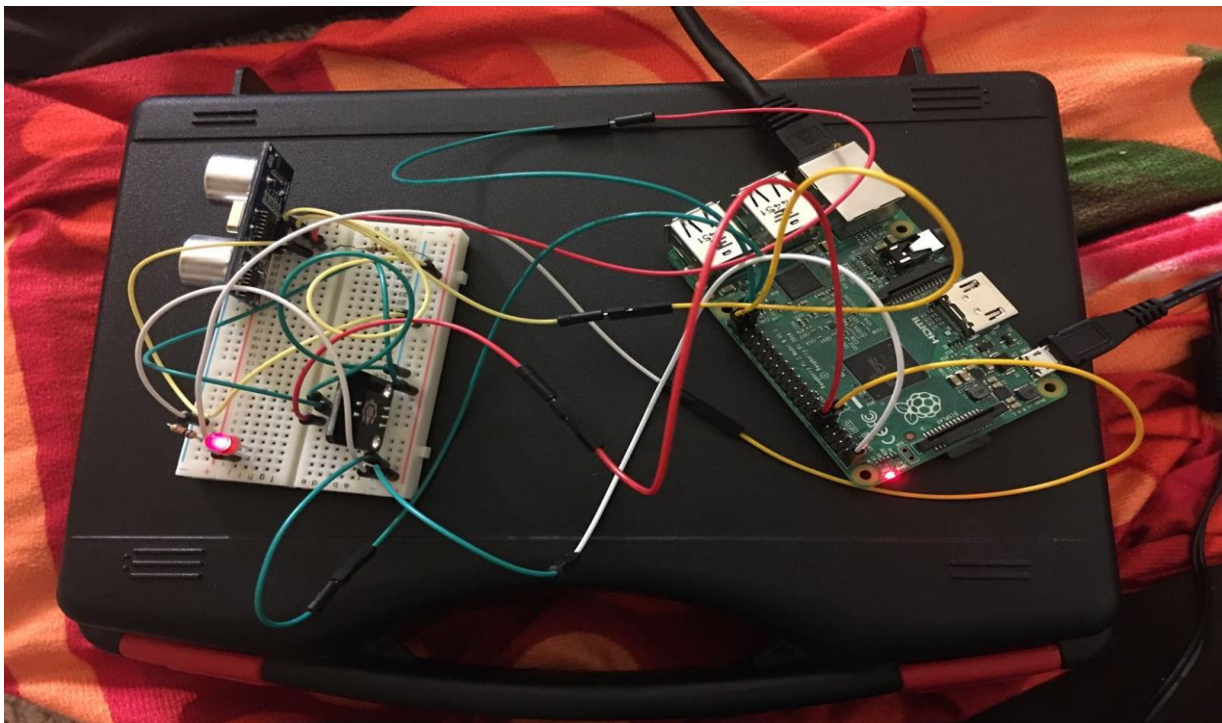
Lab Assignment – 2
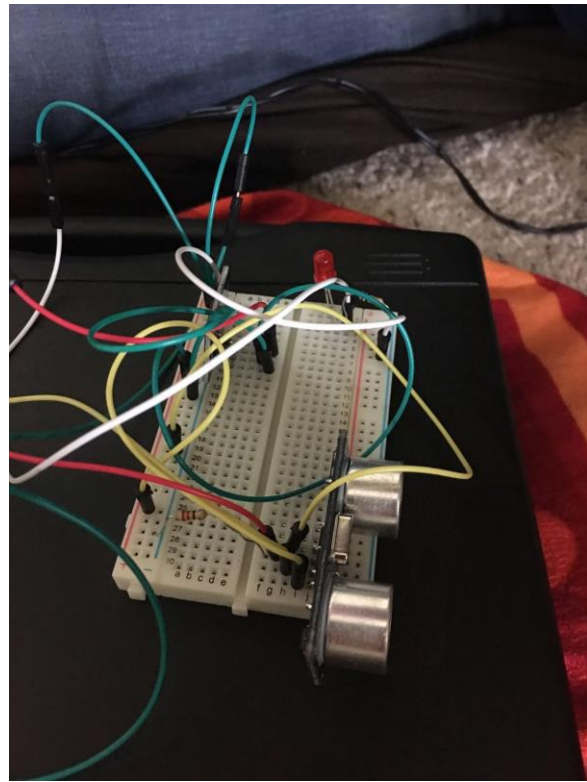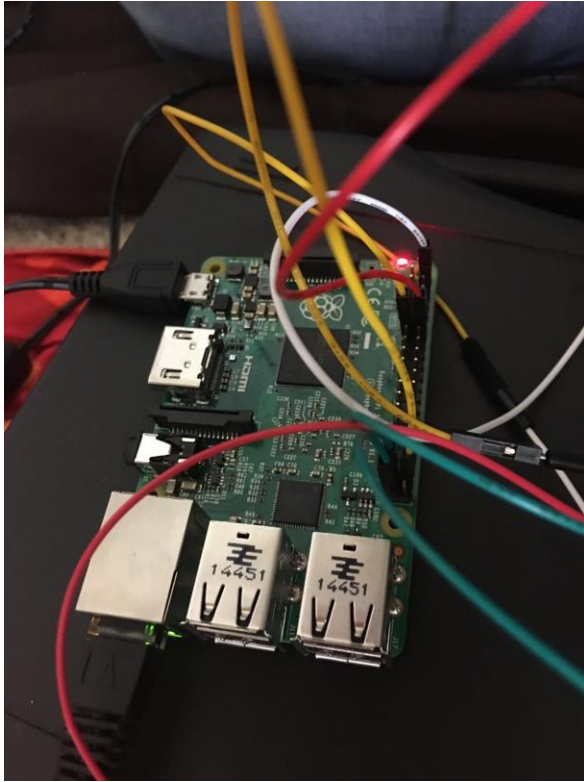
Name - Dhruv Kalaria

ID – 010100519

Date - 04/17/2016

I. Raspberry Pi Connection and Setup:

II. Instructions for Circuit Connection:

List of Pins Used in Raspberry Pi (BCM Mode):

1. GPIO Pin 18 – Connection to Light Sensor.

2. GPIO Pin 20 – Connection to Proximity Sensor's TRIG.

3. GPIO Pin 26 – Connection to Proximity Sensor's ECHO.

4. GPIO Pin 17 – LED Status pin.

Connect the pins as indicated above between Raspberry Pi and Sensors.

III. Server IP and Ports

Mosquitto Broker Listeners:

- 50.18.94.136:1883
- Websocket: 50.18.94.136:9001

Kafka:

- Server IP: 50.18.94.136:9092

- Zookeeper IP: 50.18.94.136:2181

Elasticsearch:

- IP: 50.18.94.136:9200
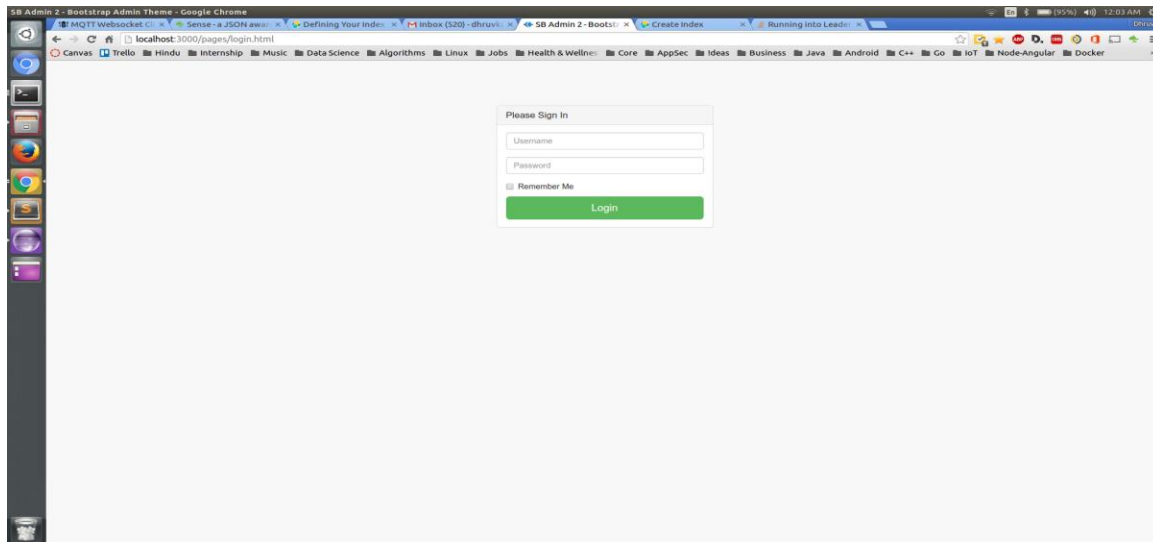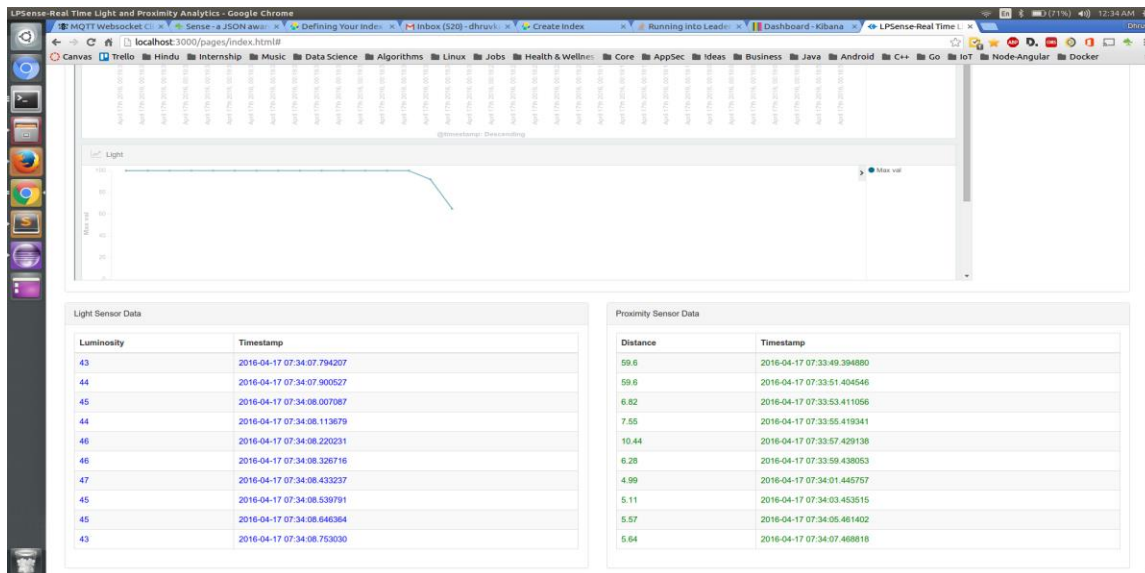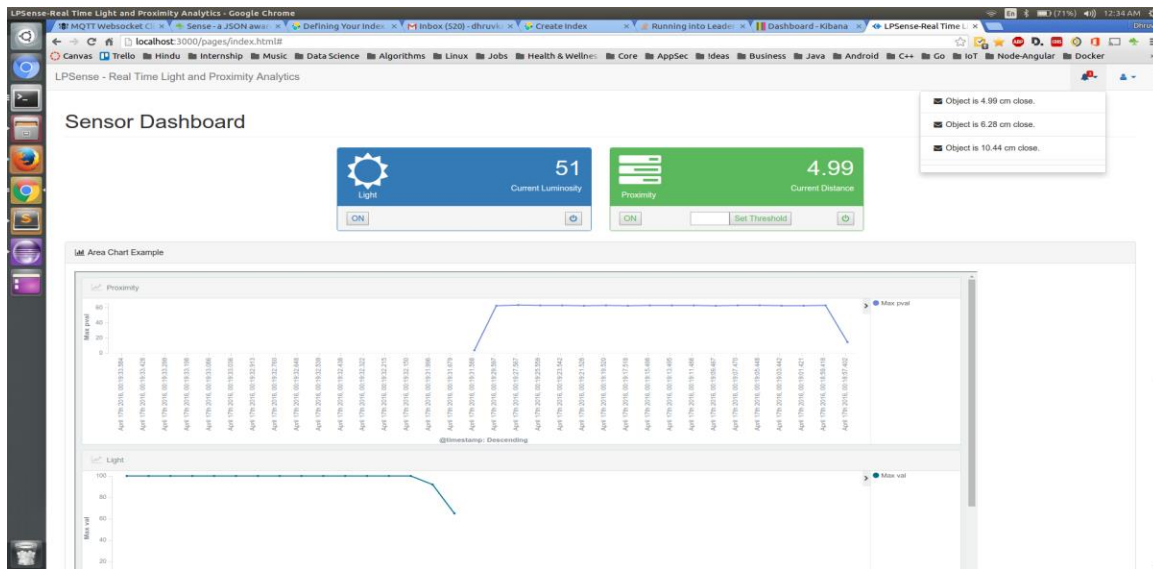
Logstash:

- IP: 50.18.94.136:9300

Kibana:

- IP: 50.18.94.136:5601

WebApp:

- IP: 50.18.94.136:3000

IV. Screenshots of Web Application:

V. Setup and Execute Code:

Instruction to Run Raspberry Pi Code:

1. There are individual python codes for light sensor and Proximity sensor.

2. Install paho mqtt client for python using command 'pip install paho-mqtt' and memcache using command 'pip install memcache'

3. Run- sudo python light_sensor.py and sudo python range_sensor.py to execute the codes in different ssh sessions.

Instruction to Run VM Code on AWS EC2 Instance:

1. Install zookeeper and then install and start kafka server using command - './kafka-server-start.sh ../config/server.properties'

2. Provide the Mosquitto broker IP:Port, kafka server IP:Port and zookeeper IP:Port to the bridge and then run MQTT-Kafka Broker jar as 'java –jar MqttKafkaBridge.jar'

3. Install and Start the Elasticsearch using command - './bin/elasticsearch'

4. Install and Start Logstash using command - './logstash agent –f ../logstash.conf'. logstash.conf file has the required filters, input and output setup.

5. Install and Start Kibana using command - './bin/kibana'

WebApp on AWS EC2 Instance:

 1. Install npm package manager

 2. Install serve using command 'npm install serve -g'

 3. Navigate to the folder containing the WebApp and then type command 'serve'

 4. Access the WebApp from url - 50.18.94.136:3000

Part 2:

1. State pros and cons of using Kafka in your IoT Infrastructure.

Pros:

  a. Kafka provides a fault tolerant stream of data to the logstash consumer. Mosquitto is not scalable, so if the velocity of data increases, kafka is able to accommodate multiple mosquitto broker streams.
  b. Logstash currently does not support MQTT protocol and Mosquitto supports only MQTT. So we needed Kafka which converts MQTT to HTTP.

Cons:

  a. Kafka server utilizes more memory footprint on the server and hence the resource utilization is very high.
  b. It also acts as a bottleneck in non HA configuration that if Kafka fails then whole pipeline fails.
  c. An additional Mosquitto-Kafka bridge is required to provide input from Mosquitto to Kafka as Kafka also doesn't support MQTT which utilizes additional resources.

2. State reasons why would you not use Apache Flume in your IoT Infrastructure. Discuss situations where you can use Flume in an IoT Infrastructure.

Reasons for not using Flume:

a. Flume does not provide broker overflowing while Kafka provides by storing data on disk.
b. Flume does not provide velocity control to consumers as it is PUSH based model which pushes data to Sink(Targeted Storage). So if the velocity of data is high then, consumers may drop some packets. Kafka is good as consumers can PULL data of it.
c. Flume does not provide replication. Hence if any of the node fails then we lose all events.

Situations to use Flume:

a. High velocity sensing data can be stored in sinks like – HDFS, Cassandra and Batch processing can be done on it using Hadoop.
b. Mosquitto can directly publish data into Flume which can be used to store data in HDFS
c. Flume can be used to store data to multiple sinks as well, so we can store data in one the sink and also use kafka as one of the sinks for flume