**Project Setup instructions:**

This document describes the process of environment setup and installation various required modules for running this project.

**Prerequisites:**

Below tools should be preinstalled before proceeding further for the required environment setup.

* Hadoop cluster
* Apache Kafka
* Apache Hive
* Python 2.7
* MySQL

**Installing Python modules:**

First create a virtual environment for the project.

1. Virtual environment:

A Virtual Environment is a tool to keep the dependencies required by different projects in separate places, by creating virtual Python environments for them. It solves the “Project X depends on version 1.x but, Project Y needs 4.x” dilemma, and keeps your global site-packages directory clean and manageable.

$ mkdir analytics\_app

$ cd analytics\_app

$ virtualenv venv

1. To begin using the virtual environment, it needs to be activated:

$ source venv/bin/activate

1. Run the below command to install all required python modules. List of required modules are specified in requirements.txt file.

$ pip install -r requirements.txt

All modules should be successfully installed.

**Creating Kafka topics:**

In this section we are going to create two Kafka topics by running below list of commands.

$ kafka-topics.sh --create --zookeeper <zookeeperhost:port> --replication-factor 1 --partitions 1 --topic iot-sensor-data

Kafka replication factor and partitions can be adjusted as per the project requirement and cluster size.

Once the topic is created, list the topic with below command to cross check whether same is created or not.

$ kafka-topics.sh --list --zookeeper <zookeeperhost:port >| grep iot-sensor-data

Since our application will process every message within a minute of message creation, we do not need to store the old Kafka messages in our topic. We can alter the data retention time with below command.

This will set retention time of every message to 5 minutes.

$ kafka-topics.sh --zookeeper <zookeeperhost:port> --alter --topic iot-sensor-data --config retention.ms=5000

Likewise create another topic with name: iot-analytics-data

Below commands to manually produce and consume Kafka topic messages for debugging purpose.

# to find the IP address of any broker from zookeeper

$ zookeeper-client

$ get /brokers/id/0

$ kafka-console-producer.sh --broker-list <kafka\_topic\_broker:port> --topic iot-sensor-data

$ kafka-console-consumer.sh --zookeeper <zookeeperhost:port> --topic iot-sensor-data --from-beginning

**Creating MySQL tables:**

Create below two tables in MySQL specified schema.

SQL>create table BUILDING\_LOOKUP (

BUILDING\_ID VARCHAR(10),

BUILDING\_ZONE VARCHAR(10),

BUILDING\_MGR VARCHAR(50),

BUILDING\_AGE INT,

HVAC\_PRODUCT VARCHAR(20),

RUNNING\_FLAG VARCHAR(10),

LOCATION VARCHAR(50),

TARGET\_TEMP INT,

CUR\_OCCUP INT,

LAST\_UPDATED DATE

);

SQL> create table RULES (

RULE\_ID VARCHAR(10),

TEMP\_MIN INT,

TEMP\_MAX INT,

OCCU\_MIN INT,

OCCU\_MAX INT,

TRIGGER\_VALUE INT

);

Load Building lookup data and rules into above created tables.

SQL> LOAD DATA LOCAL INFILE '/home/IOT/building\_lookup.csv'

INTO TABLE BUILDING\_LOOKUP

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n';

SQL> LOAD DATA LOCAL INFILE '/home/IOT/RULES.csv'

INTO TABLE BUILDING\_LOOKUP

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n';

**Create below directory on HDFS home path:**

$ hadoop fs –mkdir $HOME/sensor\_data

Replace $HOME with home directory of HDFS.

**Creating Hive tables:**

$ create external table temperature\_data(BUILDING\_ID string,ZONE\_ID string,OLD\_TEMP int,NEW\_TEMP int,OLD\_TEMP int,NEW\_OCC int,RUNNING\_FLAG string,UPDATED\_TIME timestamp)

row format delimited

fields terminated by ','

lines terminated by '\n'

stored as textfile

location '$HOME/sensor\_data/'

Replace $HOME with home directory of HDFS.

**Node.js Setup:**

Install node dependencies by running below commands.

$ cd node

$ npm install

$ node index.js

This will start the node application process on port 3002. Port can be modified in index.js file.

**Running the application:**

After successfully setting up the environment, run below scripts in different windows.

1. Below script will push dummy sensor data exists in data directory into first stage of Kafka topic.

$ ./put\_sensor\_data\_in\_topic.sh

1. Below python program consumes sensor data coming from Kafka topic and process the data by comparing with lookup tables and rules and pushes the every event to HDFS log file directory we created in above steps.

$ cd IOT

$ source venv/bin/activate

$ python app2.py

1. Schedule below python script in cron for every one minute.

Replace $HOME with home directory of UNIX file system where our entire code repository resides.

\* \* \* \* \* bash $HOME/IOT/hive\_schedule.sh >> $HOME/IOT/hive\_schedule.log

1. Run the node server if same is already not running.

$ cd IOT

$ cd node

$ node index.js

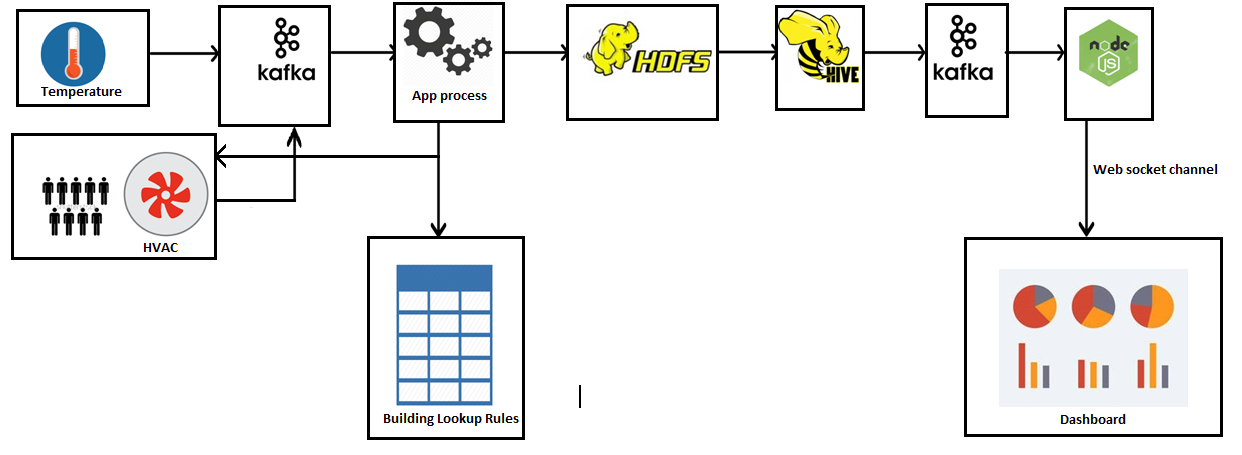
**Dashboard:**

Open browser window and enter below URL to access the dashboard screen.

URL: <http://hostname_of_server:3002/>



**Reference flow diagram:**



Kishore Garapati

kishore.garapati@tcs.com