The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

Dan Thompson

Weather effects on 311 Calls in NYC

# Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

## Data Source

### Technology Choice

I have selected two data sources. One is a public data of every non-emergency (311) call made in the city of New York since 2010. The other is a pull of hourly weather and precipitation data from New York’s Central Park that I requested from NOAA. It goes back to 2015. Both data sets are current.

### Justification

I searched through a lot of weather sites before finding NOAA. I was enthused that it went back several years, and was accurate, hourly, and free, as well as available in an easily digestible CSV format.

The 311 data has a lot of interesting fields, and because it has a time stamp it can be matched up to the weather data. But the data set is several gigabytes in size and needs to be transformed and cleaned before it can be worked with.

## Enterprise Data

### Technology Choice

I don’t think this is going to use any enterprise data. My earlier efforts were going to look at enterprise data, but it was too difficult to come by, and would have had to have been simulated.

### Justification

No enterprise data.

## Streaming analytics

### Technology Choice

I don’t intend to use streaming, although in a later phase maybe I’ll have an events feed with live weather and 311 call information.

### Justification

Stream seems superfluous at this point.

## Data Integration

### Technology Choice

I think Keras is a good choice for me and will do the job.

### Justification

The data is huge, so I’ll need Spark. I am pretty good at Python. And Keras seemed like the easiest for me to use.

## Data Repository

### Technology Choice

I’m going to use IBM Cloud as the repository.

### Justification

It’s fast, inexpensive, and I can hold up to 25GB at no cost. Plus, the rest of the solution will run on IBM Cloud, so it makes sense from that point of view.

## Discovery and Exploration

### Technology Choice

I’ll use Python, Pandas, Numpy and one or more visualization packages from Jupyter notebooks to help plot and visualize the data.

### Justification

These are the technologies I know best, based on the preceeding IBM/Coursera courses.

## Actionable Insights

### Technology Choice

I need some kind of notification technology when the system detects an event. At this point I think I’ll just feed a kafka topic and leave it at that.

### Justification

Kafka is ubiquitous, and easy to integrate.

## Applications / Data Products

### Technology Choice

Again, I think Kafka will be an adequate choice. I’ll publish an API.

### Justification

Kafka is ubiquitous, and easy to integrate.

## Security, Information Governance and Systems Management

### Technology Choice

I don’t have anything proprietary here, nor should there be a lot of maintenance or the need for governance on this project. I’ll make everything publicly available on git.

### Justification

This seems like added cost and effort that is not needed for this project.