Statement of Work

Objectives

This project is the smaller scope of original production base Proposal that will allow the quick development before the demonstration in August. It will consist of:

Data Ingestion from multiple sources

Backend services and persistence modules with data store.

REST interface to web application for the dashboard.

Package the above as Docker images for better delivery and easy deployment.

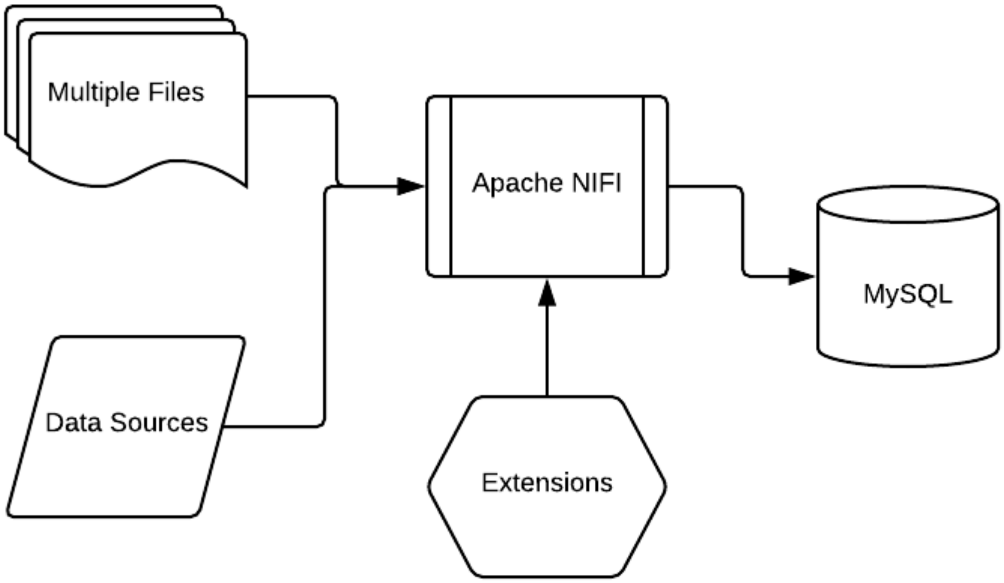
The architecture is based on the original proposal for Hypatialabs with narrow scope of it.

The schedule is trying to make it at the end of the first month that things will all get together to deploy to EC2 as the beta version. From the beta version, we will leverage the second month for alternations, additional features applied. At the end, we will make it the final demo version.

Scope of work

Data Ingestion will consist:

1. Use Apache NIFI as center to connect multiple resources and retrieve data into our system.
2. Database setup on MySQL with schema design.
3. Data format conversions in java library and customize NIFI to take conversions internally.



Backend services, data retrieval and persistence are developed in Spring framework and hibernate. They will reside in the application container as Apache Karaf. Both Spring and Hibernate are two popular frameworks in java development for backend services. With their well-structured libraries, we can speed up the development time and benefit from their stability. Apache Karaf is an enterprise level application container and bootstrapping paradigms to provide what the runtime environment needs such that we can focus on business code and application without spending much time in those runtime settings.

The third module is the **REST API interface**. To give a simple definition, REST is any interface between systems using HTTP to obtain data and generate operations on top of them in all possible formats. The features of REST are:

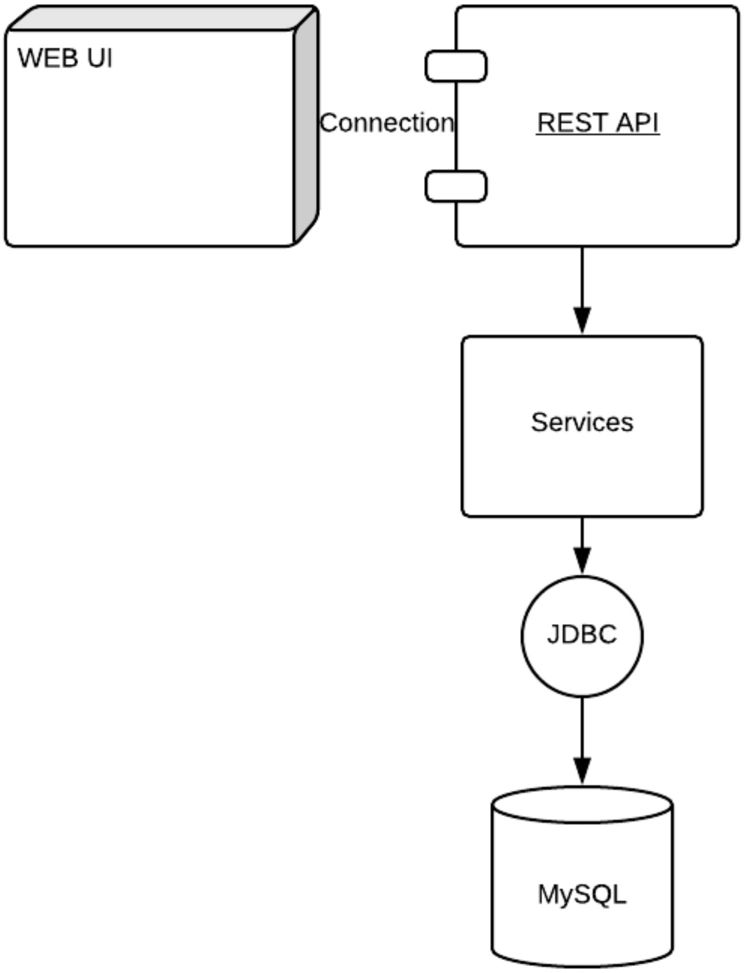
Stateless client/server protocol. We will use HTTP specification, POST for creation, GET for reading and consulting, PUT for editing and DELETE.

Uniform interface. We certainly leverage the URI that REST defined to obtain a uniform interface for data transfer. This feature provides separation between backend services and front end presentation, then decouple the complexity of data manipulation between client and server.

Furthermore, the separation between client and server has further evident advantage, that is development team can scale the products and easy migration to other servers or UI platforms.

REST API design and development will consist:

1. Open REST API standard.
2. Apache Karaf container base.
3. Swagger base implementation



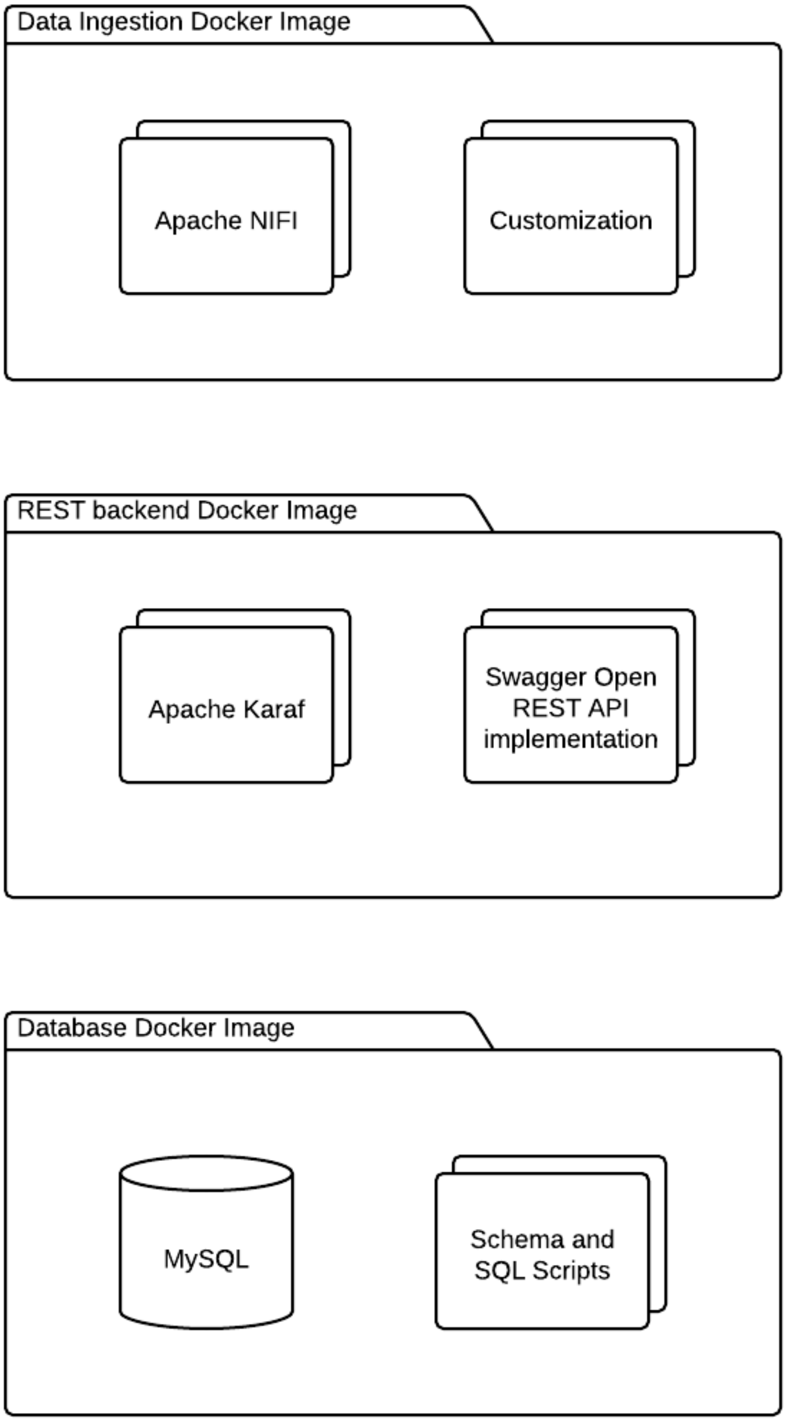
Deliverables in Docker images

It is no doubt that Docker has gained a lot of traction in the development and deliverables. With Docker containers, we can ensure our production environment setup and development environment consistently.

The portability of Docker is amazing too. Over last few years, all major cloud computing providers, including Amazon Web Services (AWS) and Google

Compute Platform (GCP), have embraced Docker’s availability and added individual support. Docker containers can be run inside an Amazon EC2 instance, Google Compute Engine instance etc. A Docker container running on an Amazon EC2 instance can easily be ported between environments. That will largely save efforts in version control and migration of our projects.

We will wrap up the modules into Docker containers - Data Ingestion, Backend services, REST and Database. Those Docker images will be the deliverables we generate either for the production or development, testing.


Data Ingestion Docker Image
Apache NIFI
Customization
REST backend Docker Image
Apache Karaf
Swagger Open REST API implementation
Database Docker Image
MySQL
Schema and SQL Scripts

Timeline

The below itemizes Deliverables and estimated time to implement. Those modules are packaged as Docker images. Each Docker instance can be tested individually and they can be under development in parallel.

	Time	Deliverable
Data Ingestion Docker Image	Estimated 25 Hours, but depends on how many data amount and sources.	NIFI data flow and Docker Image and Instance
REST Backend Docker Image	Estimated 25 Hours, but depends on how many database tables and amounts.	Java implementation and Docker Image and Instance
Database Docker Image	Estimated 25 Hours, depends on alternations and amount of SQL scripts, data format conversion.	Schema Design, data loadings and Docker Image and Instance
AMS EC2 Deployment and Web Application Integration	Estimated 25 Hours, depends on Web Application Integration.	Deployment of Docker Images to EC2 with all modules including Web App.

	Time	Deliverable
EC2 Production Environment Test and Maintenance	No Estimation.	Maintenance of production and any alternation, tuning etc.

The last phase of Production Environment Test and Maintenance is hard to estimate since many factors may involve. We can expect that the feature alternations, integration etc issues may impact the schedule and actual time.

Acceptance Criteria