Deep Logs

Project Presentation

Guide:
Dr. Shelly Sachdeva
Assistant Professor

Presented By :
Amit Kumar
202211002

May 2021

Index

- Deep Logs
- Objective
- Type of Logging
- Level of Logs
- Application Logs
- Abstract Design
- Architectural Design
- Technical Details
- Screen Shot
- Data Set

Deep Logs

- Deep Logs makes log more readable, accessible and error trackable.
- Some organizations depend entirely on infrastructure of logging such as network firewall, server OS and application logs.
- Provide business intelligence through data mining.
- The purpose of logs is troubleshooting operational and availability problems.
- Reporting and Summarization.
- Visualizing Log Data.
- Statistical Analysis.

Objective

- Deep Logs concentrate more towards application logs.
- Deep Logs UI required two inputs
  - Format Of Logs(Optional)
  - Logs File/Batch Logs File(Required)
- Extract important information from application logs
  - **important feature.**
  - **total time taken in execution.**
  - **memory consumption**,
  - **demand time of that feature.**
  - **thread execution tracking.**
  - **usual exception/error**
  - many more...

Type of Logging

- These four types of logging are produced by nearly all log sources but are analysed and consumed differently and by different system.
  - **Security logging** is focused on detecting and responding to attacks, malware infection, data theft, and other security issues.
  - **Operational logging** is performed to provide useful information to system operators such as to notify them of failures and potentially actionable conditions
  - **Compliance logging** often overlaps significantly with security logging since regulations are commonly written to improve security of systems and data.
  - **Application logging** is a special type of logging that is useful to application/system developers and not system operators. Such logging is typically disabled in production systems but can be enabled on request.

Level of Logs

- Log messages can be classified into the following general categories:-
  - **Informational**: Messages of this type are designed to let users and administrators know that something kindly has occurred.
  - **Debug**: Debug messages are generally generated from software systems in order to aid software developers troubleshoot and identify problems with running application code.
  - **Warning**: Warning messages are concerned with situations where things may be missing or needed for a system, but the absence of which will not impact system operation.
  - **Error**: Error log messages are used to relay errors that occur at various levels in a computer system.
  - **Alert**: An alert is meant to indicate that something interesting has happened.

Application Logs

- Important questions for application designer which are
  - **what** to log.
  - **when** to log.
  - **how much** to log.
  - **how to control** logging.

What to log

- **Exceptions**
- **Events**
- **States of Process's Workflow**
- **Debug Information**
- **Executed SQLs**
- **User Http Requests**
- **Executing Threads**

When to logs

- **Any Process/Operation Completion.**
- **Warning from Server.**
- **Access Conflicts.**
- **User Login and Logout.**
- **Memory and Disk Utilization.**
- **Time Taken by Services.**
- **Table state change in DB such as new insert/delete/update.**
- **Monitoring core services.**

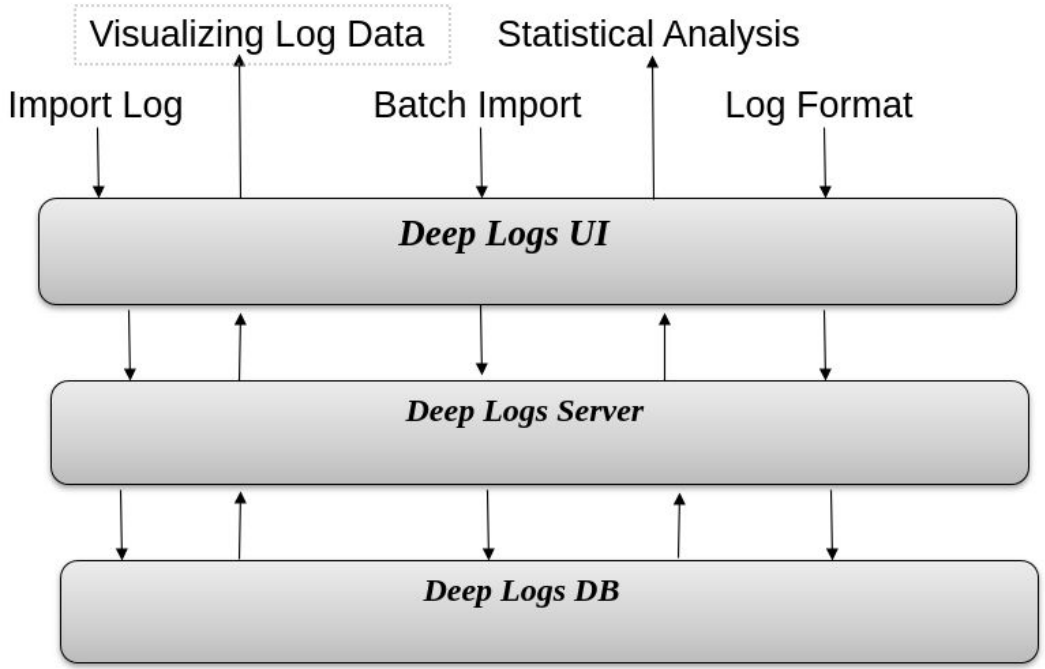
How much to logs

- **Most application have different format of logging according to the domain, components, level of access and many more.**
- **If applications begin to log a lot, application performance may severely fall down.**
- **Single Node Environment Format**
  - ***who*** *(username)* **,** ***when*** *(timestamp)* **,** ***where*** *(context, servletorpage, database)* **,** ***what*** *(command)* **,** ***result*** *(exception)*
- **Cluster Environment Format**
  - ***who*** ***(****username****)*** **,** ***when*** ***(****timestamp****)*** **,** ***where*** ***(****context, servletorpage, database****)*** **,** ***what*** ***(****command****)*** **,** ***result*** ***(****exception****), node number***

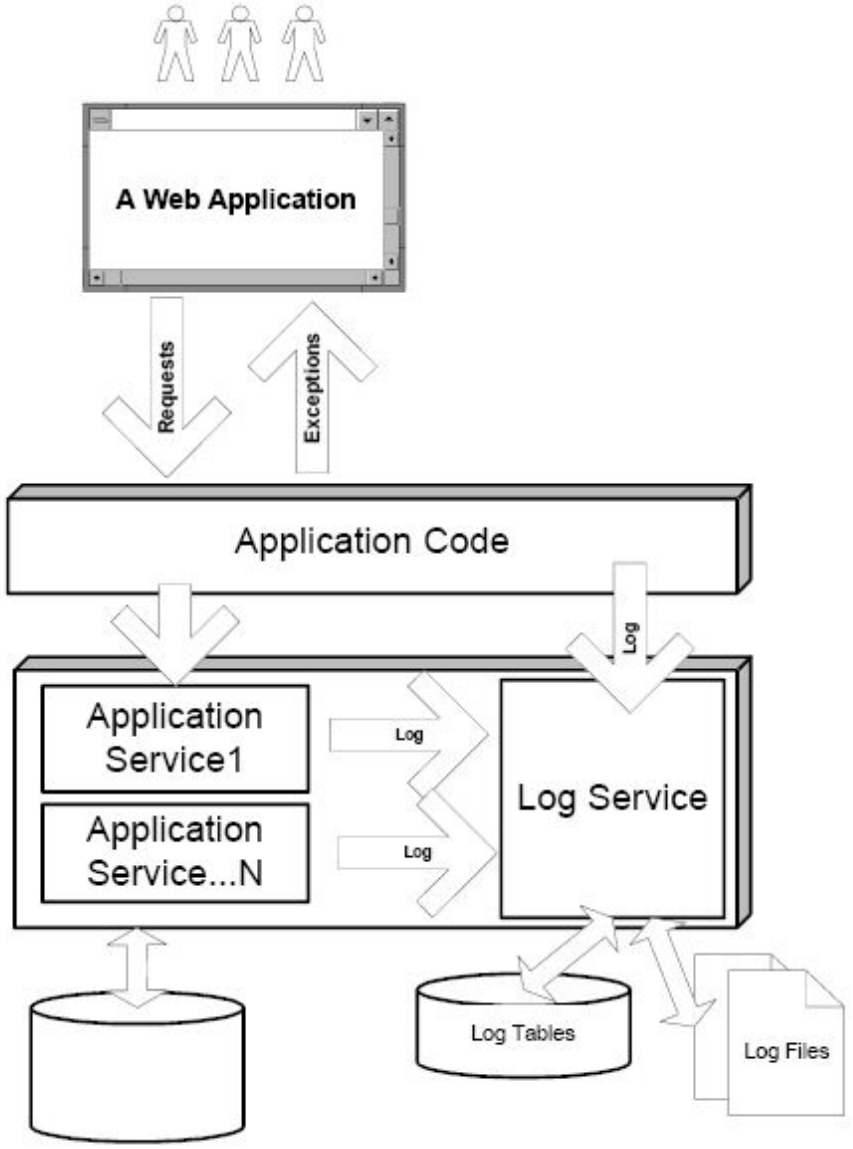
How to control logging

- **Need to know important component or feature application.**
- **Usage of application (24X7 or Scheduled Based)**
- **Using Abstraction and Separating component logs.**
- **Using Distributed Architecture.**

Abstract Design



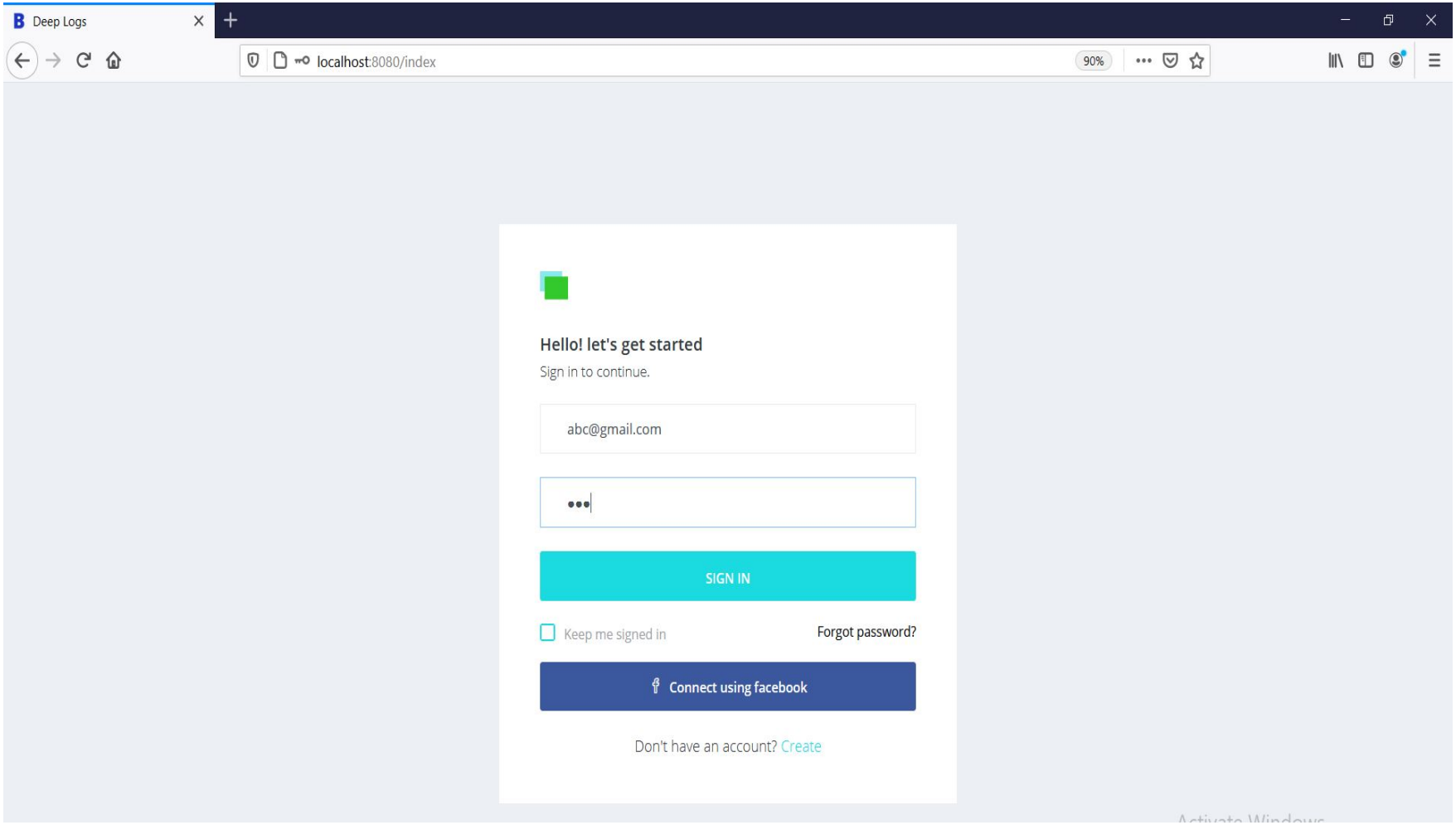
Architectural Design



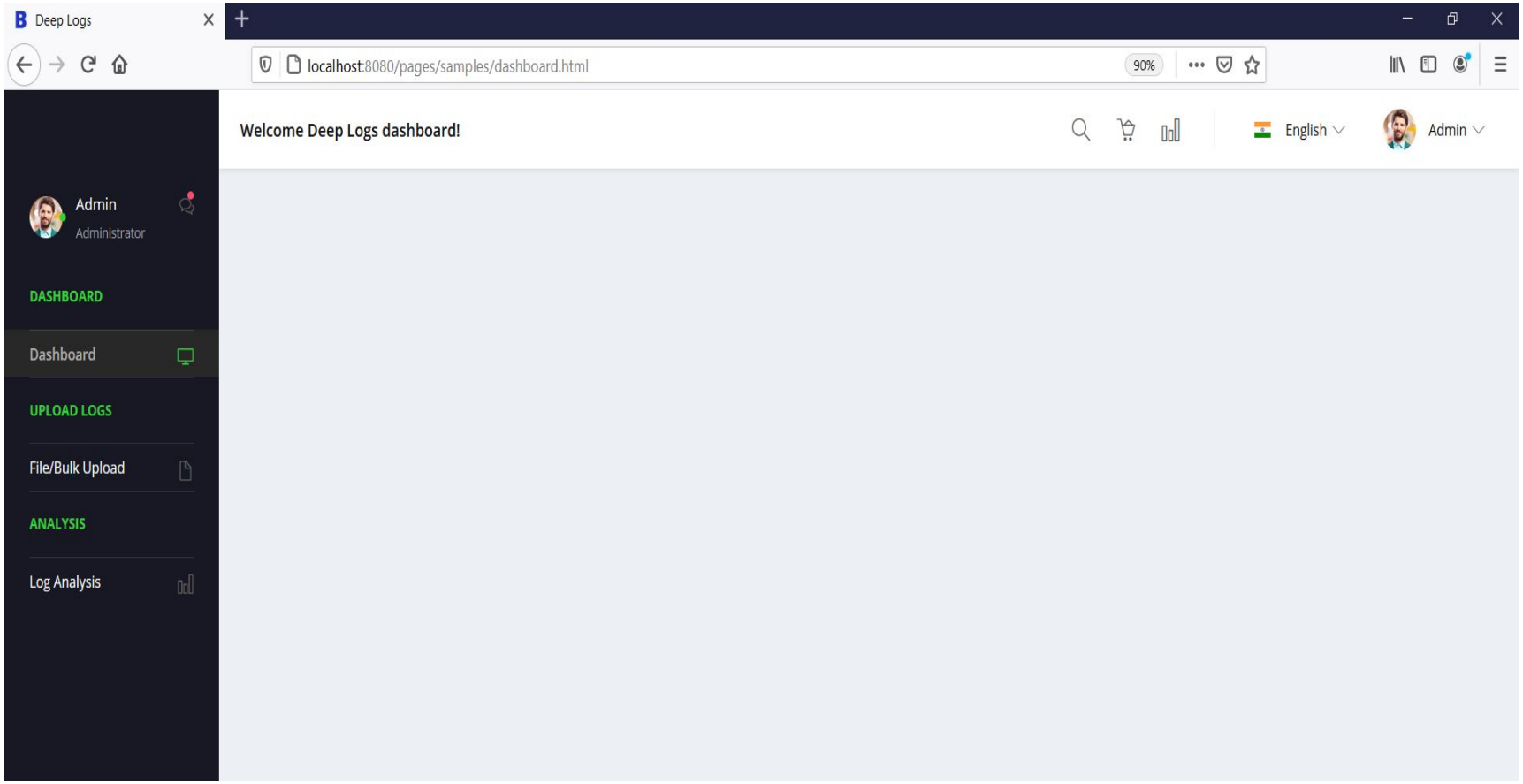
Technical Details

- User Interface
  - jQuery, Bootstrap CSS, D3 JavaScript
- Server
  - Spring Boot, REST Web Service, Hibernate
- NoSQL DB
  - MongoDB, h2 in-memory
- Apache OpenNLP
  - Tokenization, Lemmatization
- Deeplearning4j, Java-ML
  - k-means clustering, math utility methods
- Machine Learning for Language Toolkit (MALLET)
  - Natural language processing algorithms and utilities.

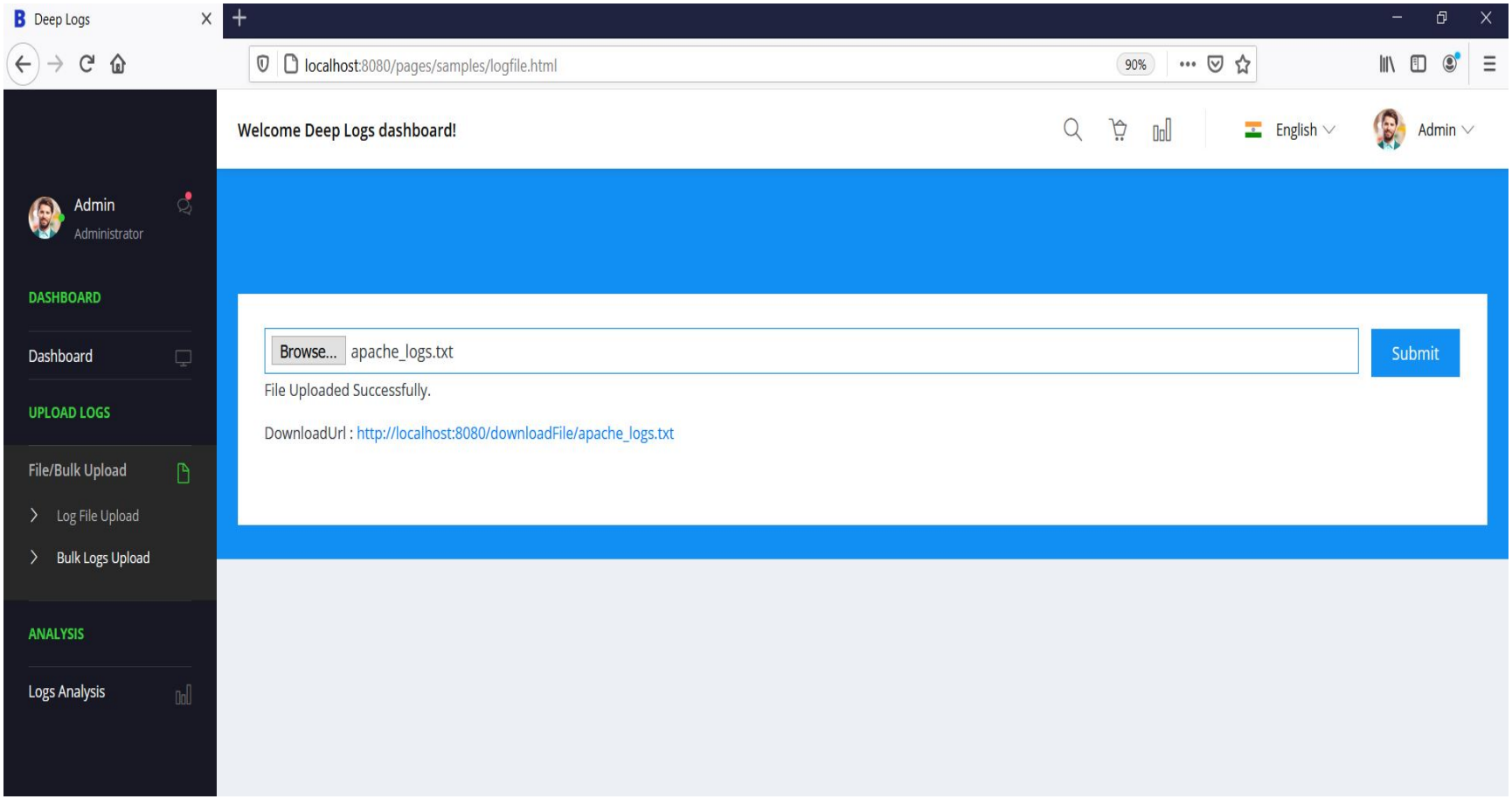
Login Screen



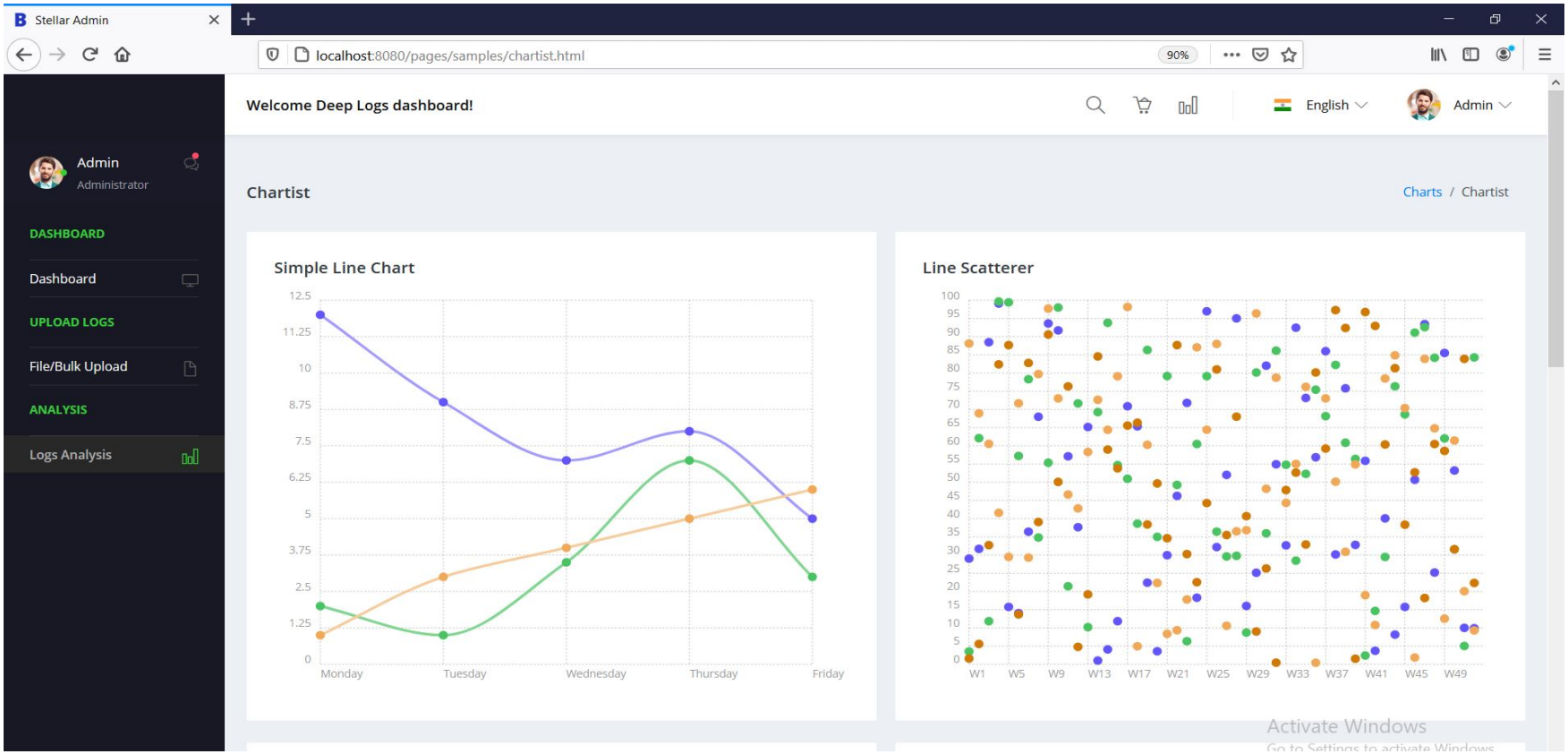
Dashboard


Deep Logs
localhost:8080/pages/samples/dashboard.html
90%
Welcome Deep Logs dashboard!
English
Admin
Admin
Administrator
DASHBOARD
Dashboard
UPLOAD LOGS
File/Bulk Upload
ANALYSIS
Log Analysis

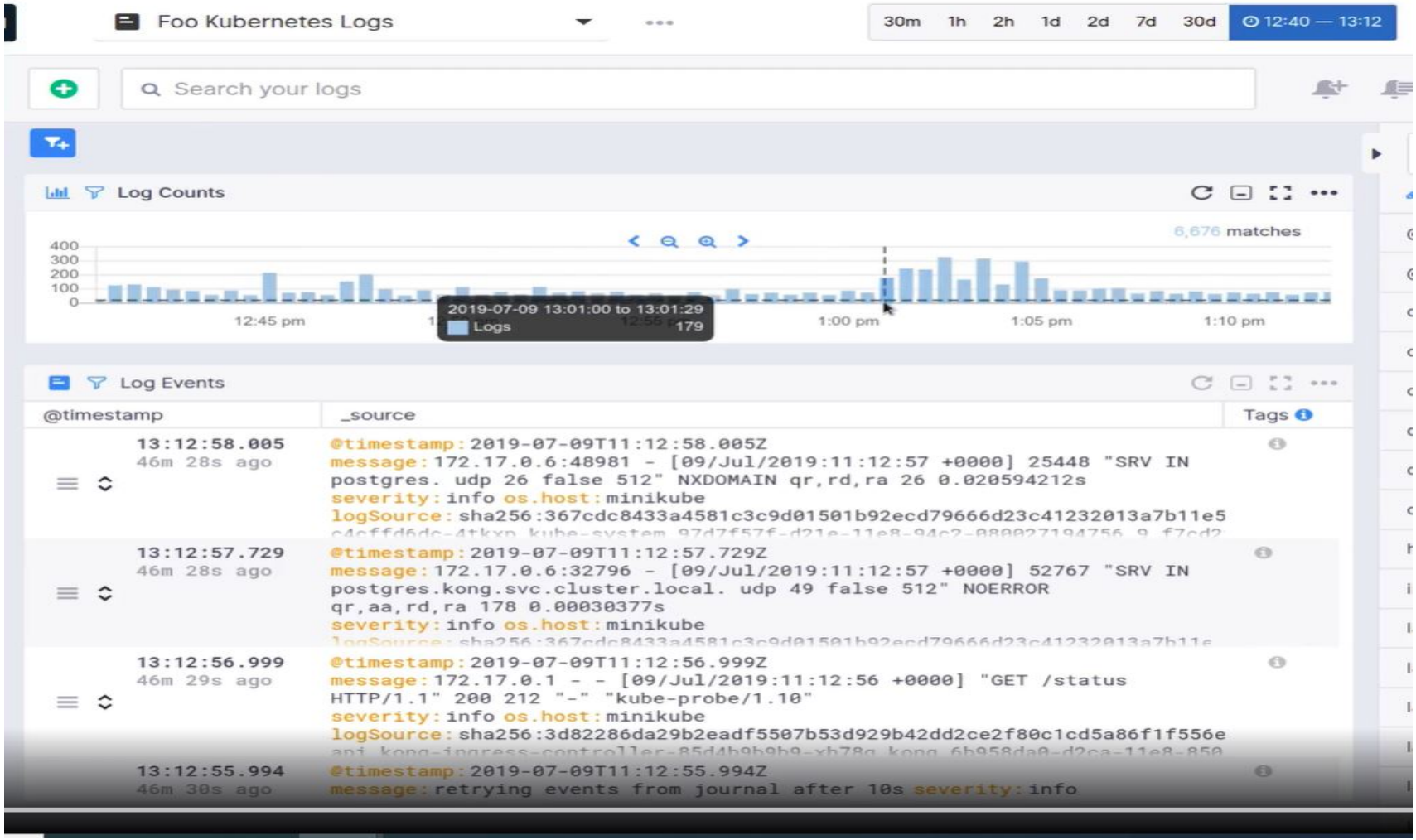
Upload and Download



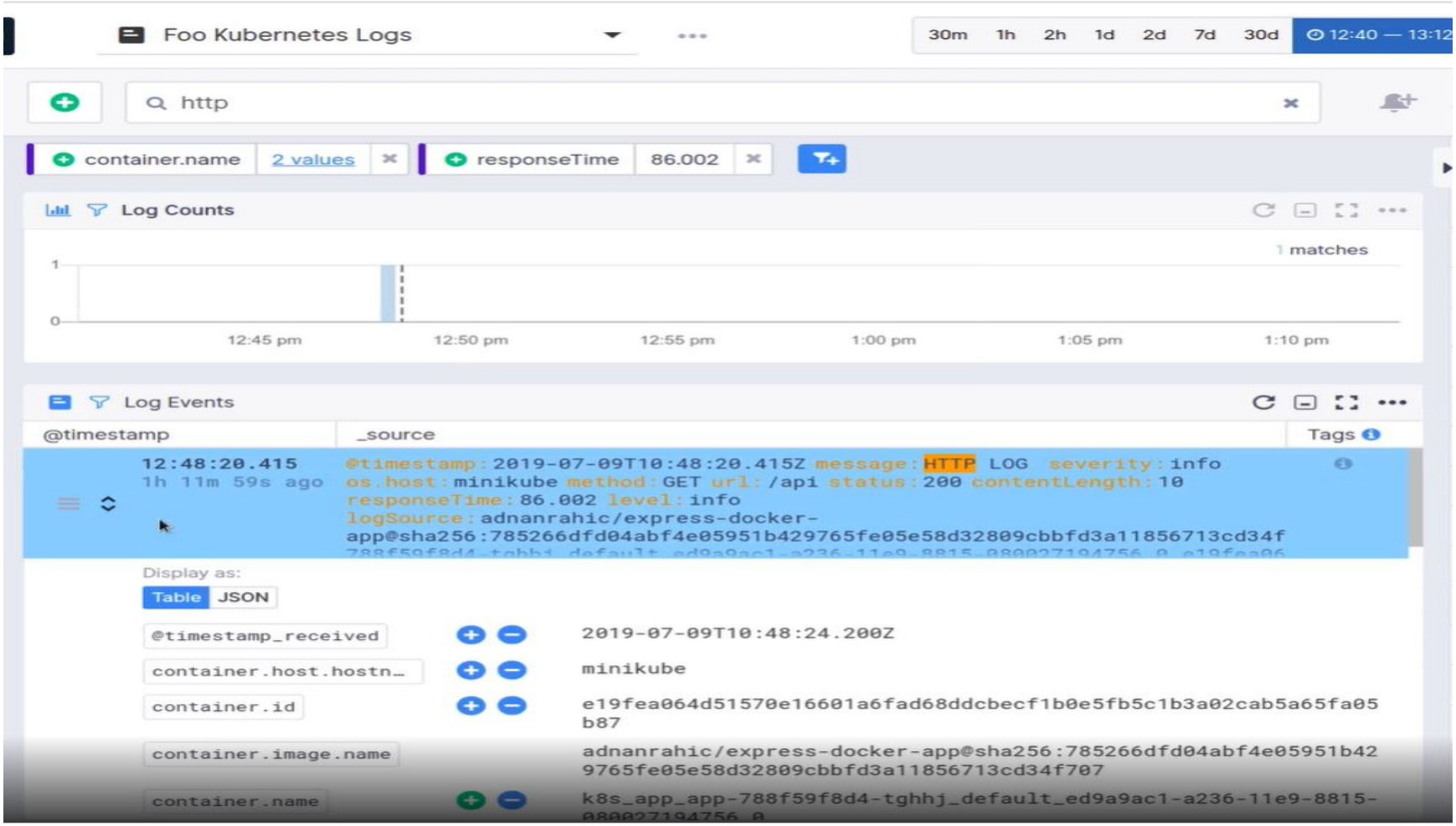
Log Analysis



Log Analysis



Log Analysis



Data Sets

- https://github.com/elastic/examples/tree/master/Common%20Data%20Formats/apache_logs