# ELK POC on Java EMS application (byRucha)

1. Downloaded and unzipped elasticsearch1.5.1 file
2. Setup environment variable
3. elasticsearch service is started from command line using elasticsearch command
4. localhost:9200 is the URL to access elasticsearch information
5. install elasticsearch head plugin using this command

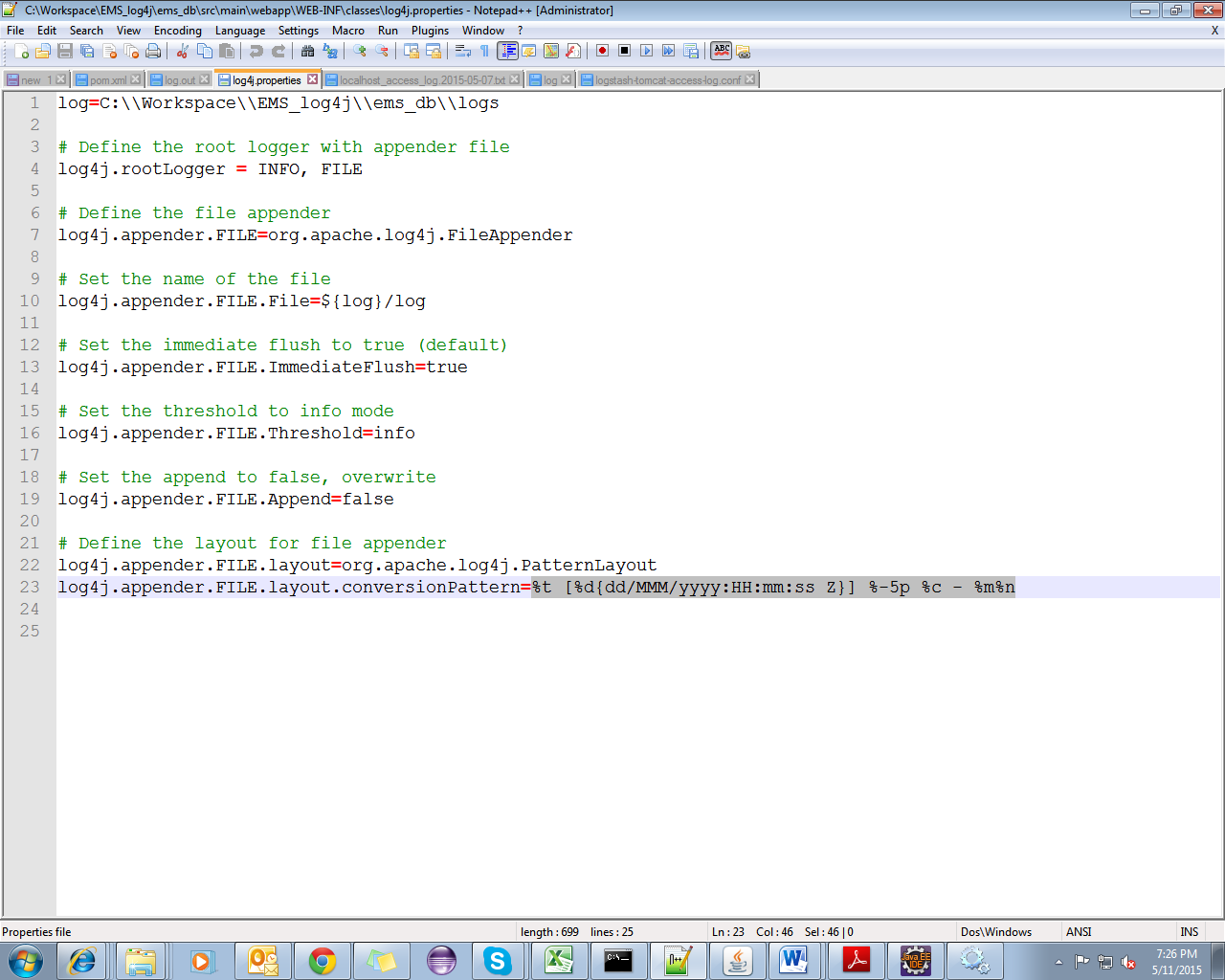
plugin -install mobz/elasticsearch-head

first navigate to bin folder in elasticsearch directory

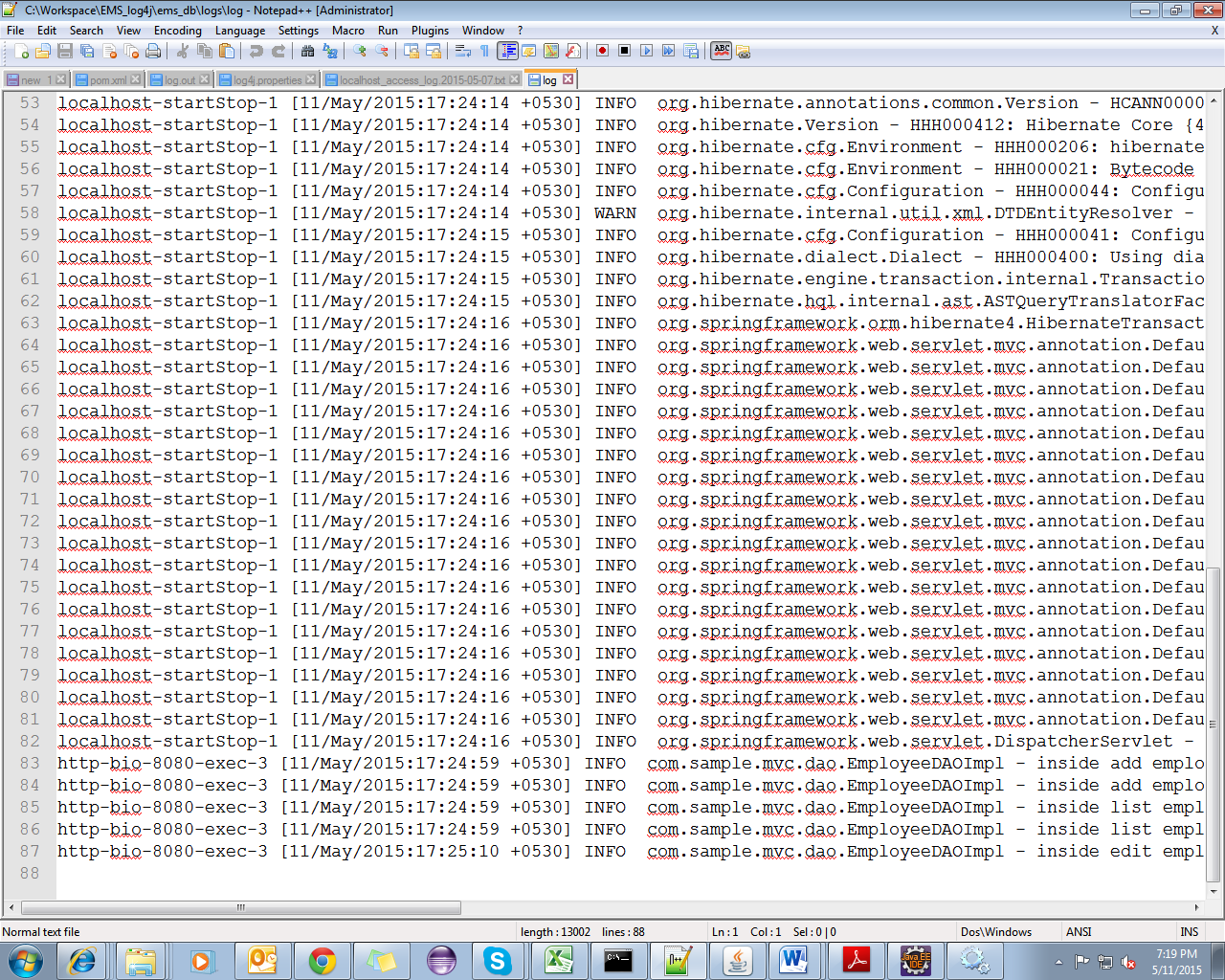
|  |  |
| --- | --- |
| **RDBMS** | **elasticsearch** |
| Database | Index |
| Table | Type |
| Row | Document |
| Column | Field |
| Primary Key | ID field |
| Schema | Mapping |
| Index | Everything is indexed |
| SQL | Query DSL |
| Select \* from table … | GET http:// |
| Update table SET … | PUT http:// |

## Indexing data into Elasticsearch from Logstash

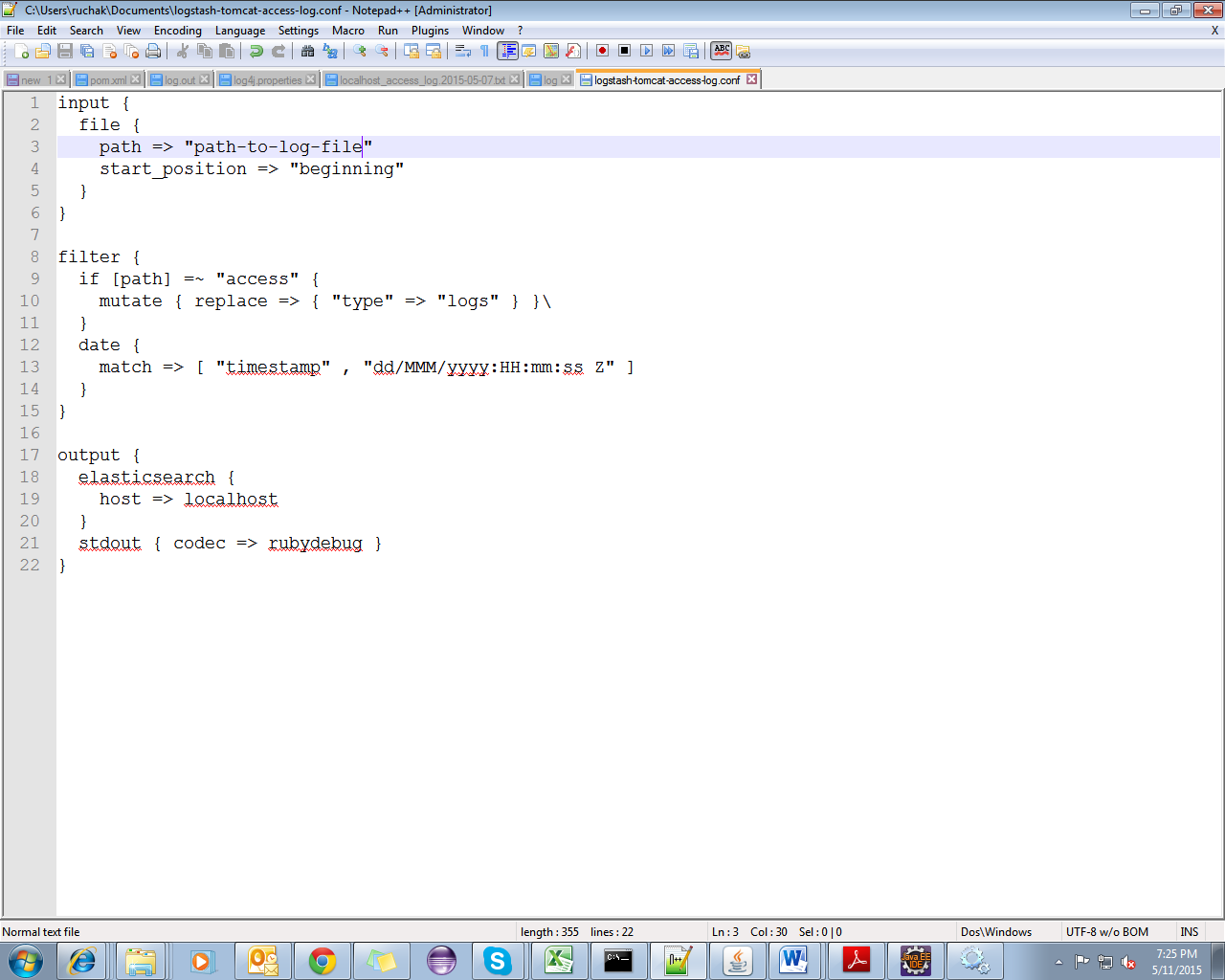
1. Created log file using pattern layout : %t [%d{dd/MMM/yyyy:HH:mm:ss Z}] %-5p %c - %m%n



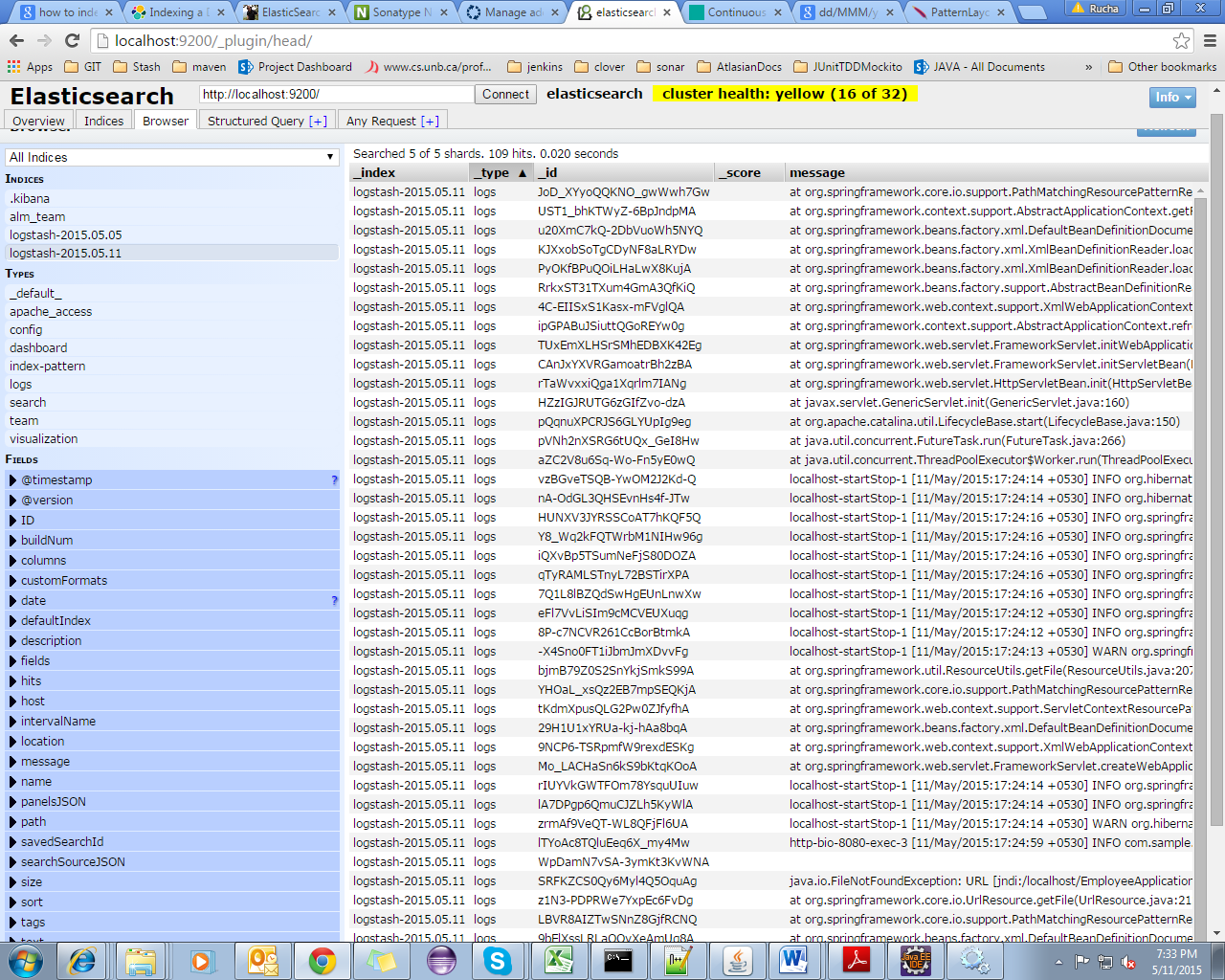
1. Generated log file by running application.



1. Created a logstash configuration file which takes input as log file and outputs data to console as well as elasticsearch. Here I’ve edited logstash-tomcat-access-log.conf file for this purpose.



1. Go to location of logstash config file. Run logstash agent –f logstash-tomcat-access-log.conf command. This will index all the data from the log file into elasticsearch.



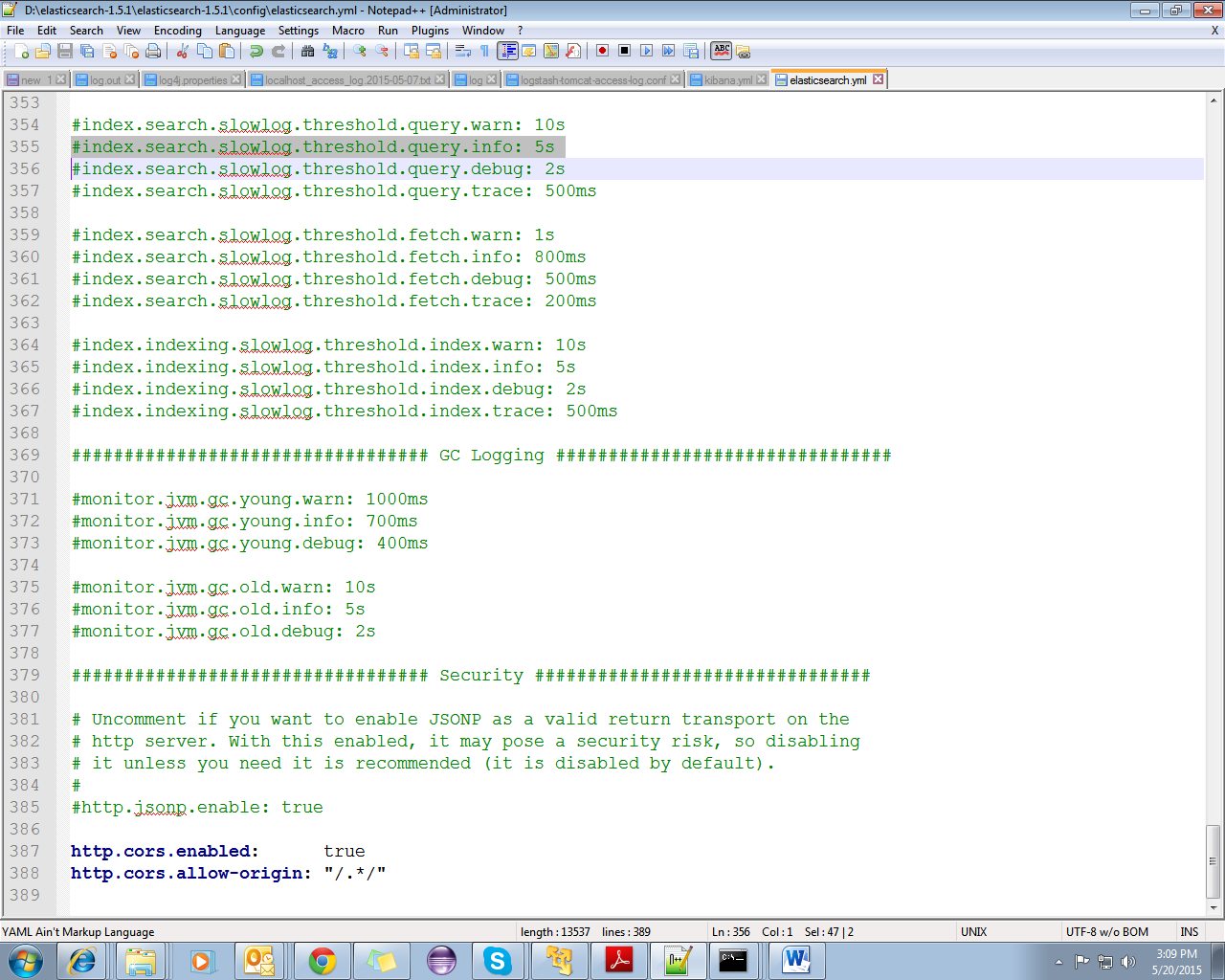
# Kibana dashboard for logstash

Prerequisites: elasticsearch, kibana and logstash are configured and up.

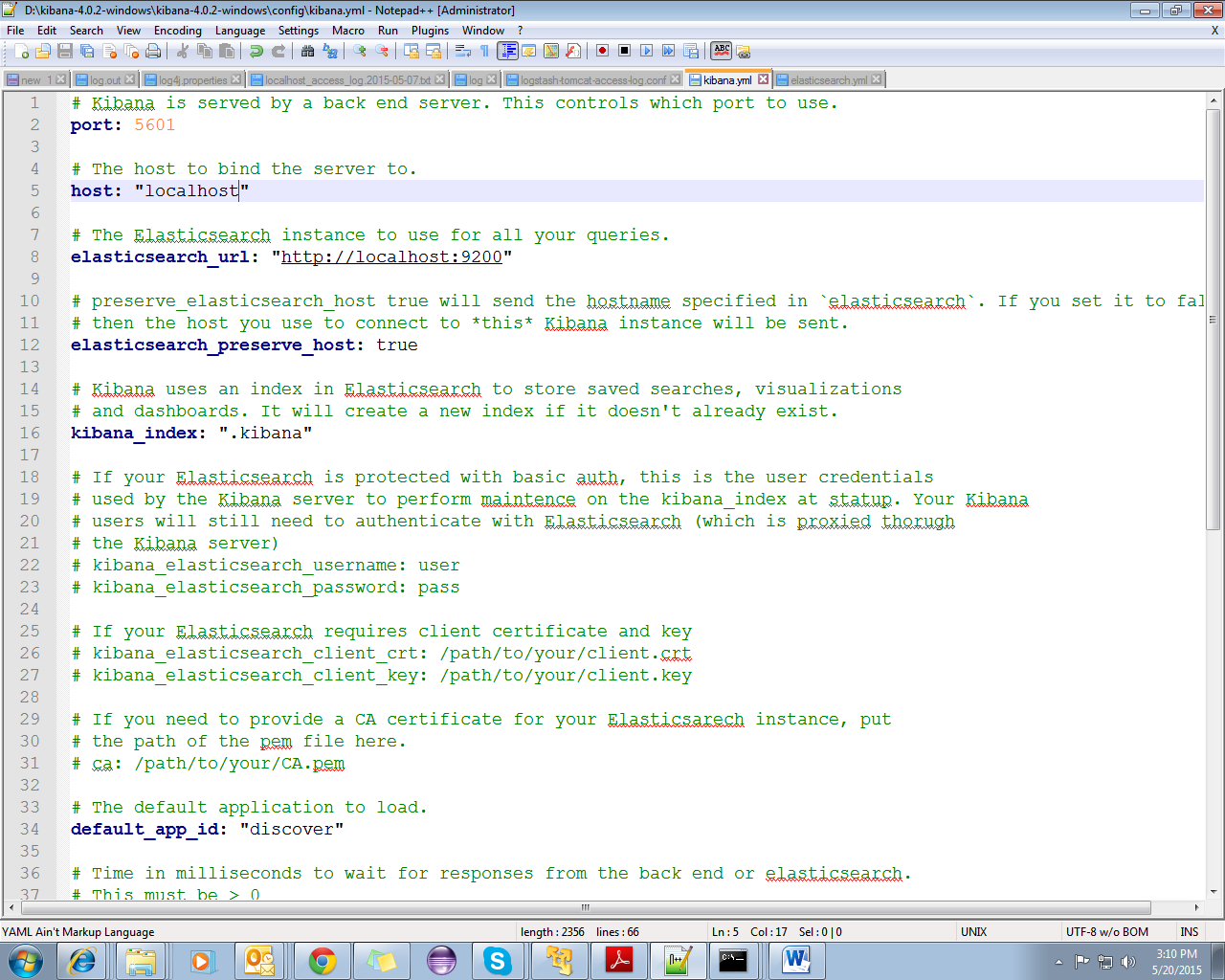
Run logstash web command through command prompt to start logstash web UI(kibana)

1. Default URL to access logastash dashboard on kibana: <http://localhost:9292/index.html>
2. Settings in yml files:

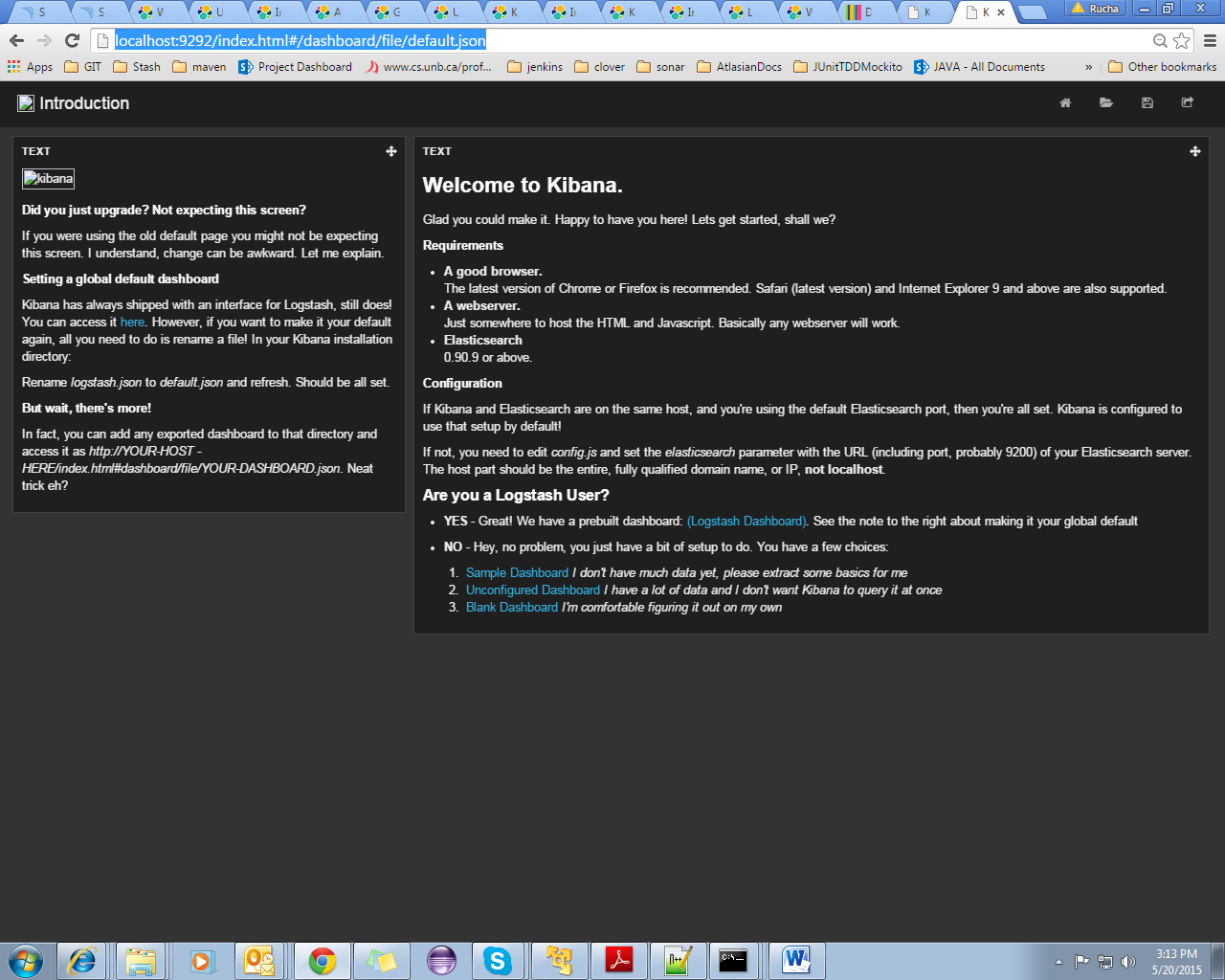
Elasticsearch.yml



kibana.yml

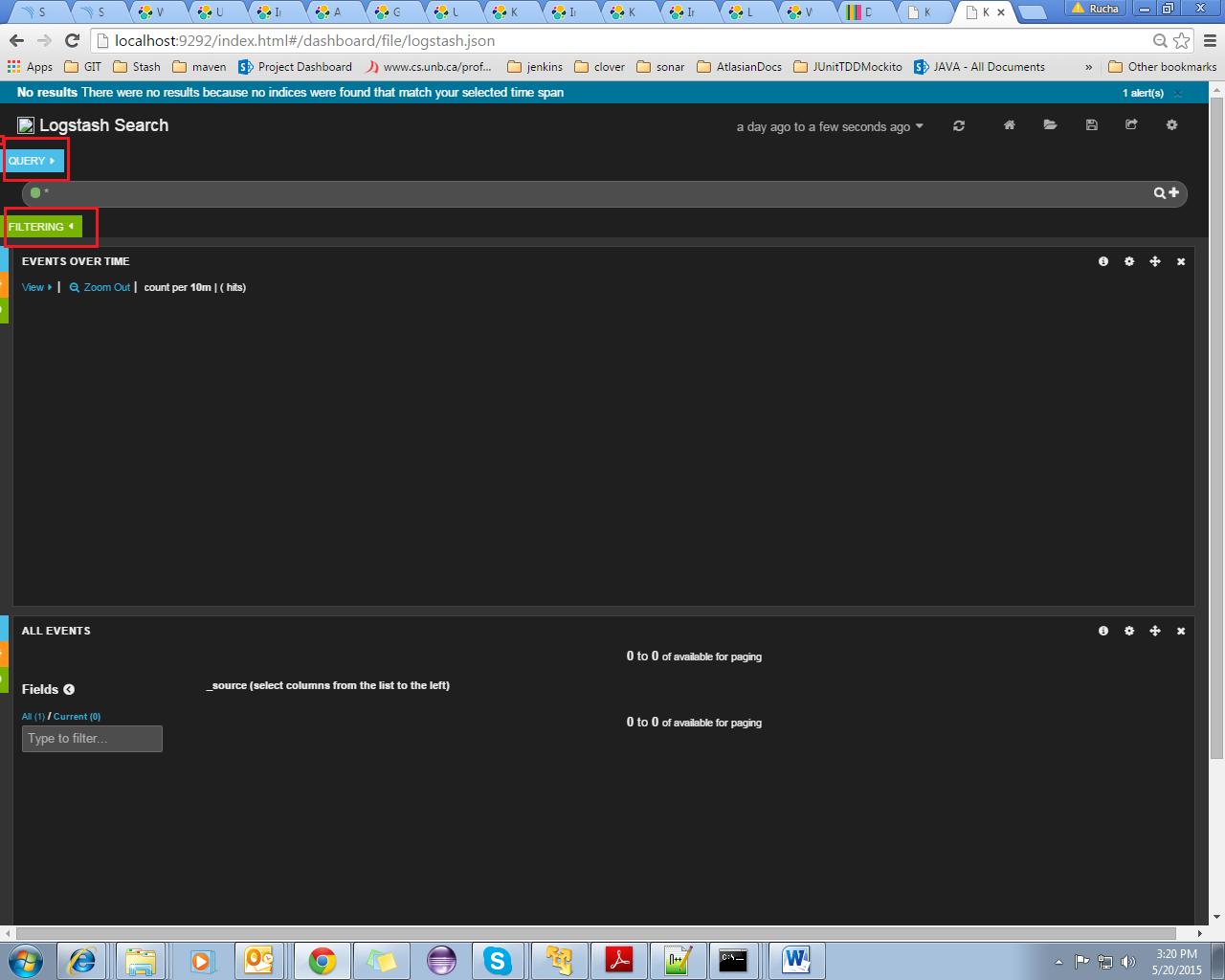


1. Kibana dashboard



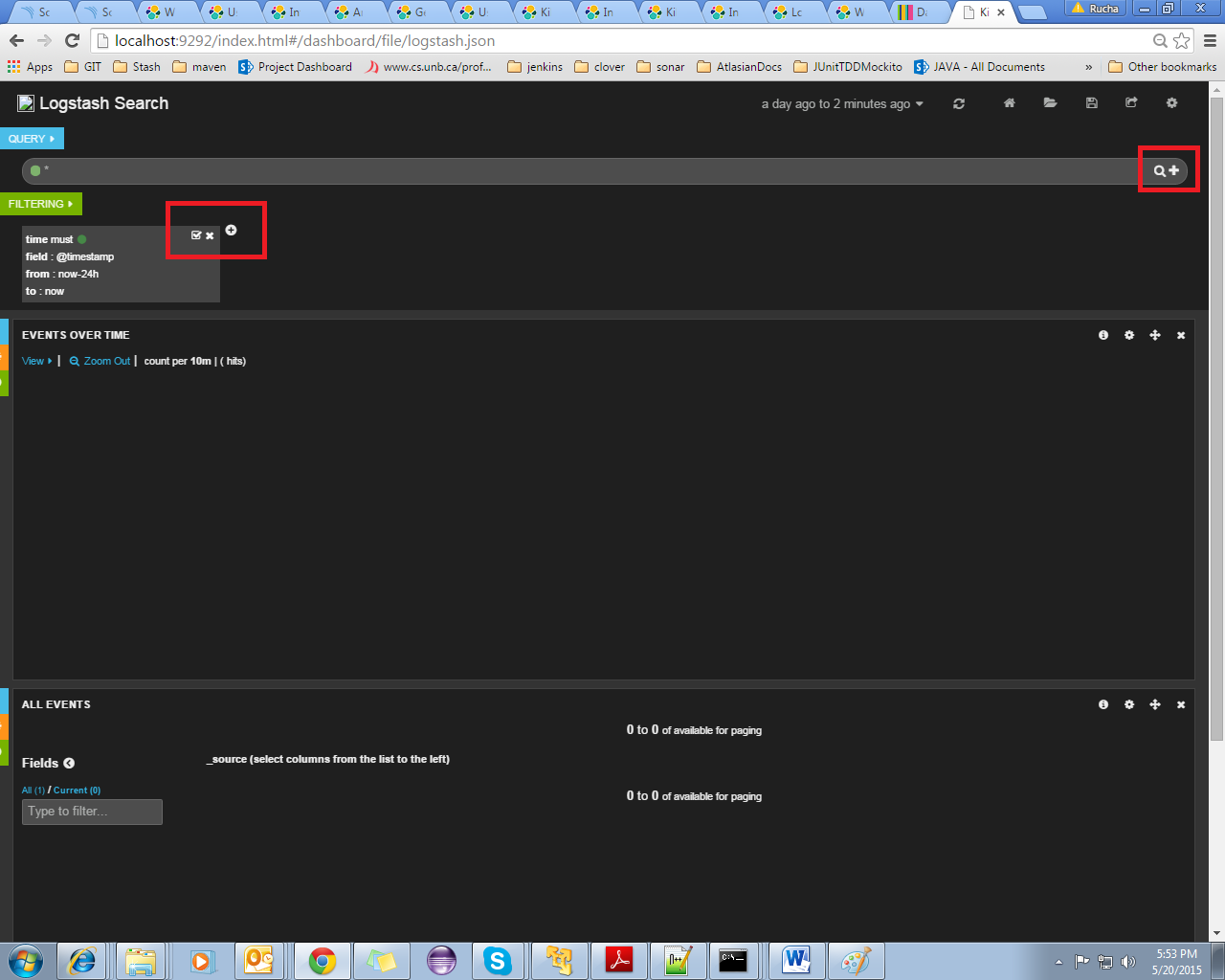
Go to logstash dashboard to see the data.

1. Logstash dashboard:



To get the information of particular event, we need to fire a query. And in filtering section we need to mention at least timestamp so that it displays the data over that period of time.

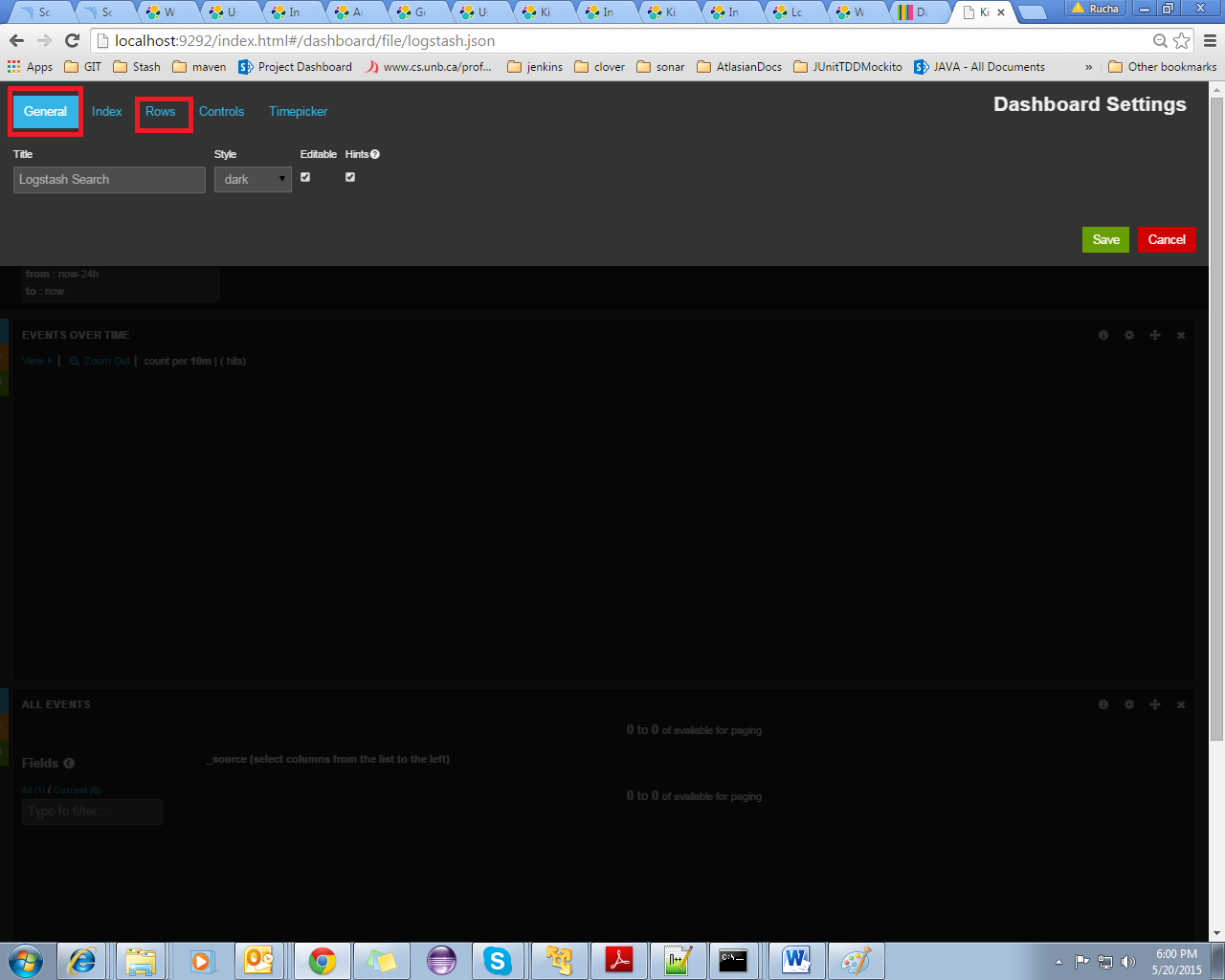
You can add multiple queries and multiple filters at the same time using add option provided.



Dashboard settings:

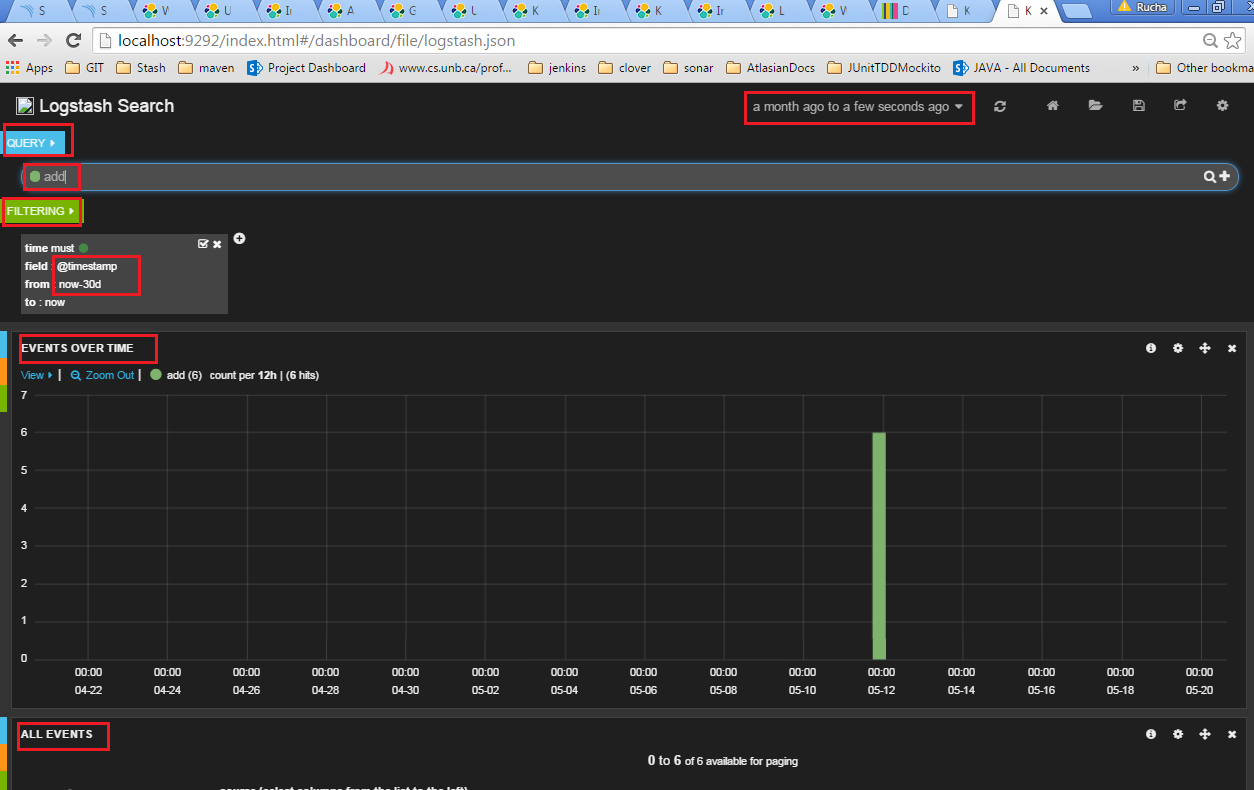
Dashboard settings can be configured using settings option in the top right corner.

In ‘general’ settings, name can be set. In ‘rows’ settings, name and sequence of rows can be configured.



1. Displaying data:

Histogram:



Query: add

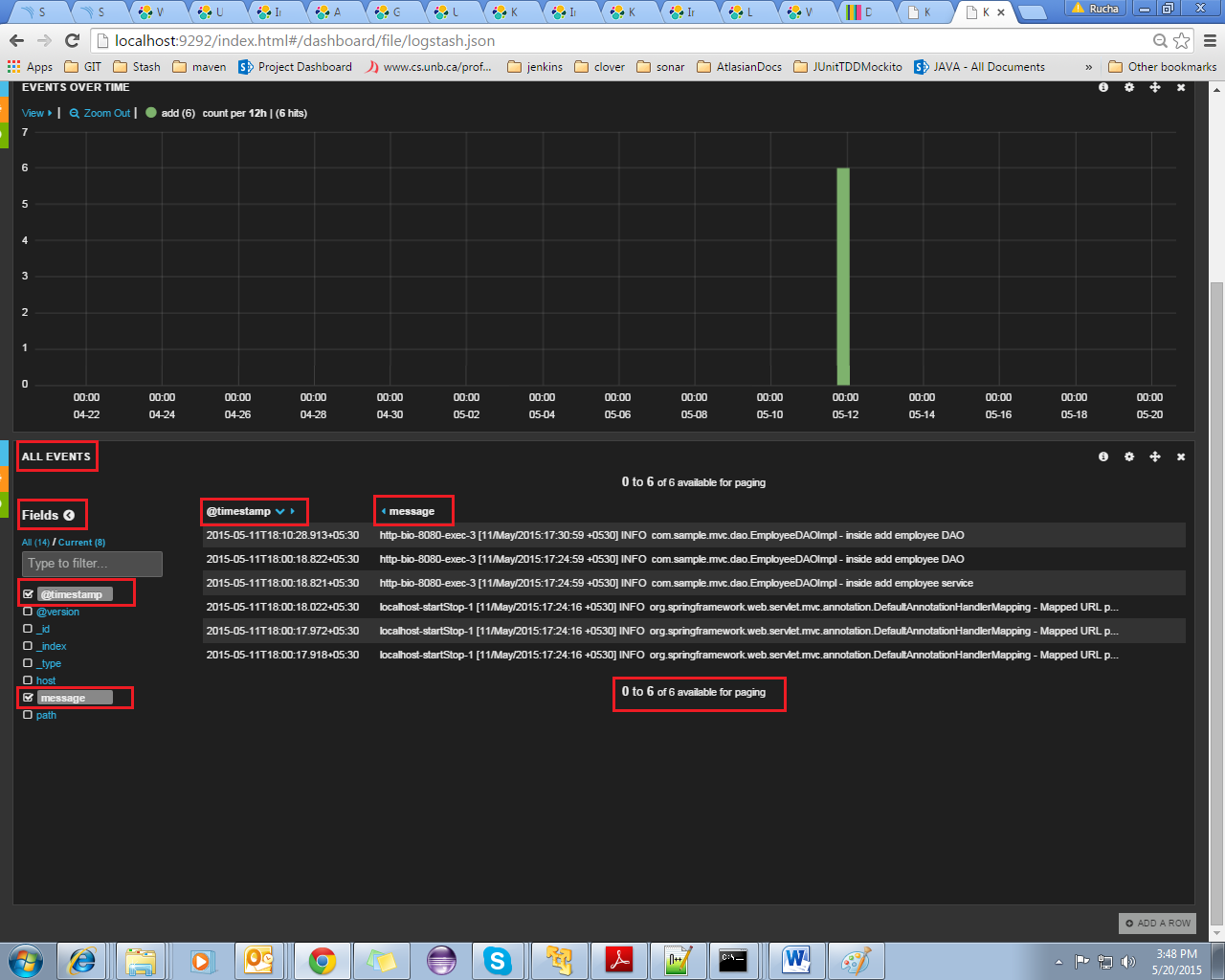
Filtering: 30 days

Events over time: displays number of events taken place in particular period of time

All events: lists all filtered events with fields specified

As you can see, add functionality has been used for 6 times on 5th May.

Listing events:

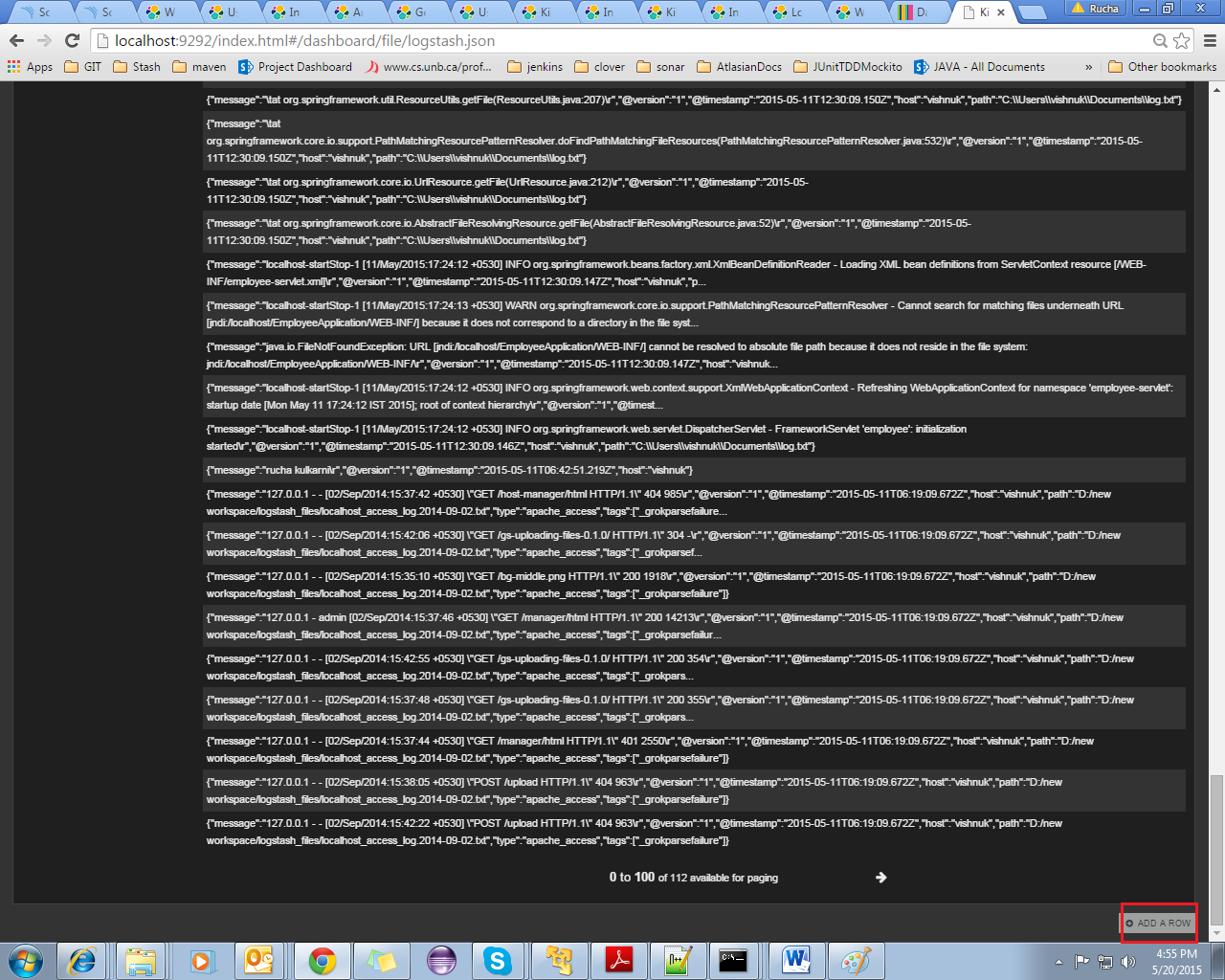


All filtered events are listed in this row with fields specified.

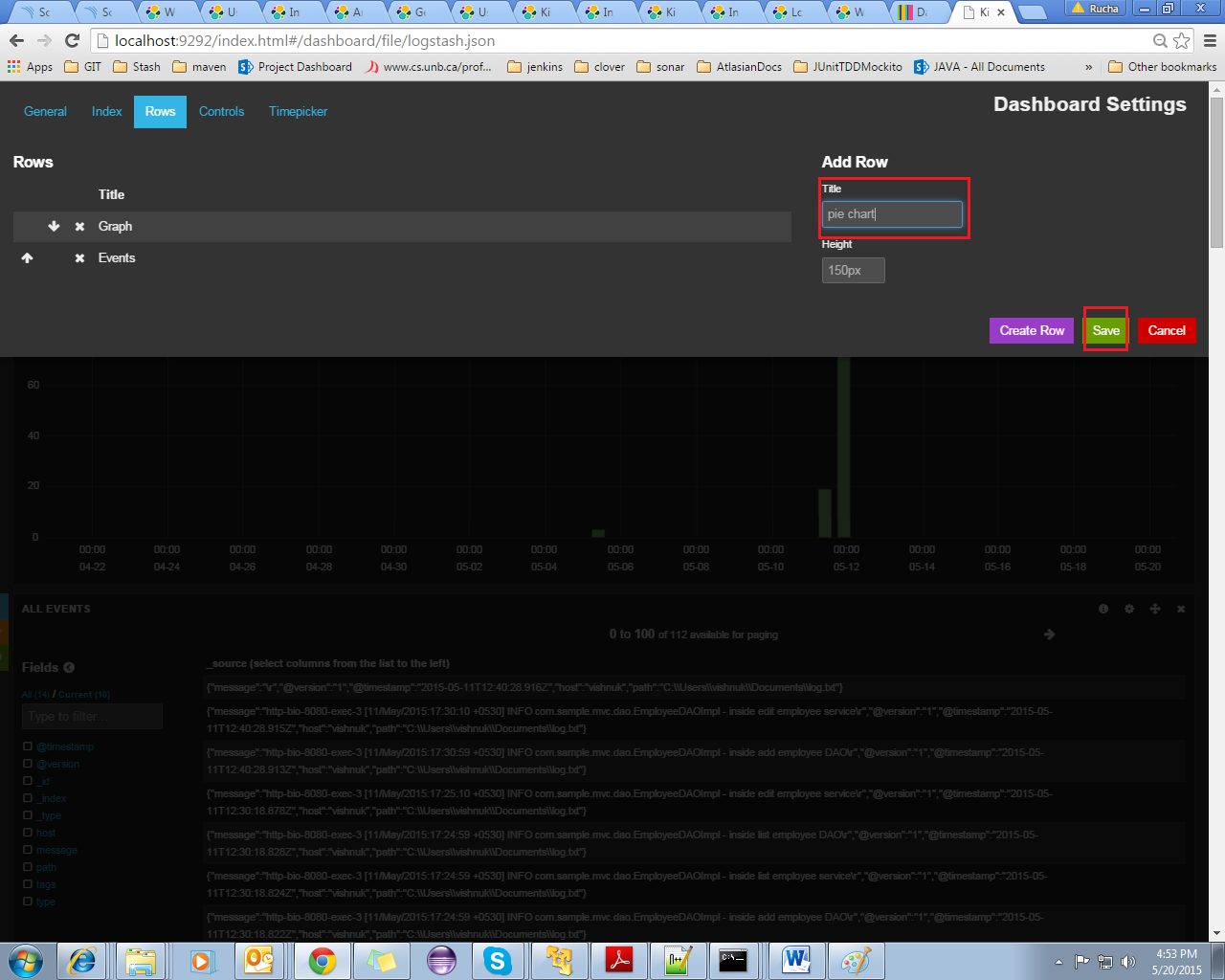
Here I have specified to display timestamp and message for an event.

1. Adding a row to dashboard:

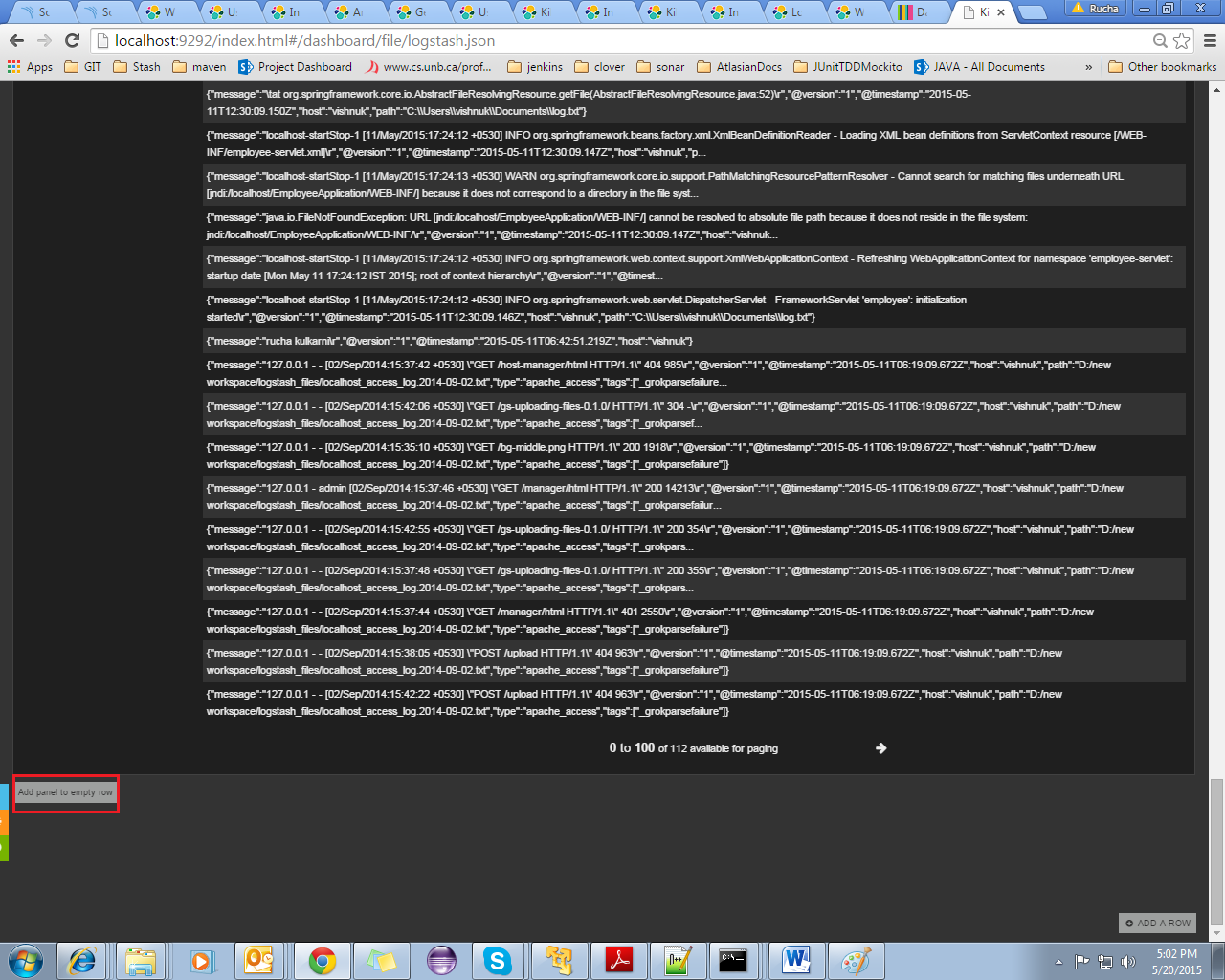
Click on the ‘create row’ option at the bottom of the dashboard.



Fill the details and save it. You can manage order of the rows on dashboard using arrows mentioned.



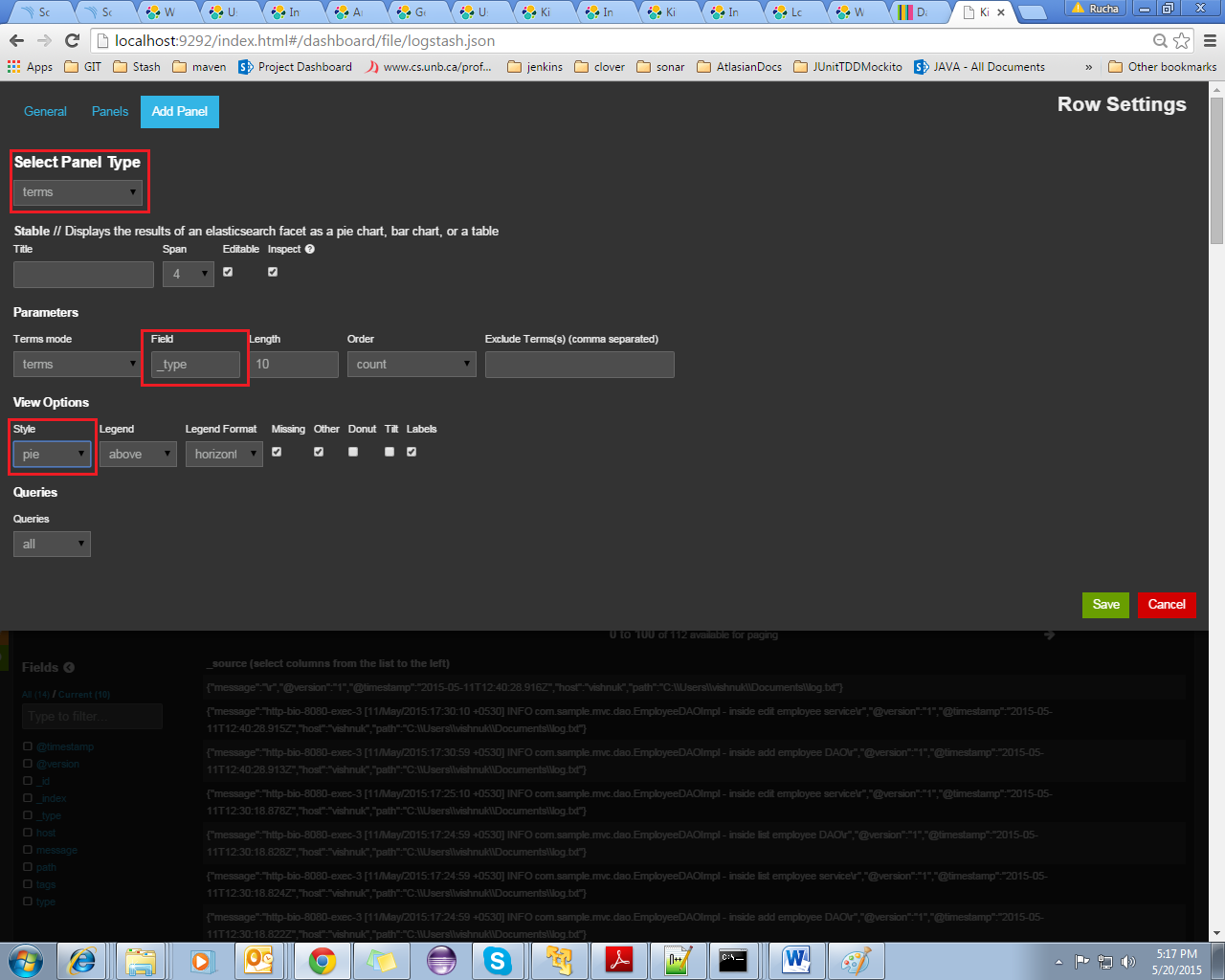
Now you can manage row settings using option ‘add panel to empty row’, which is now visible at the bottom of the dashboard.



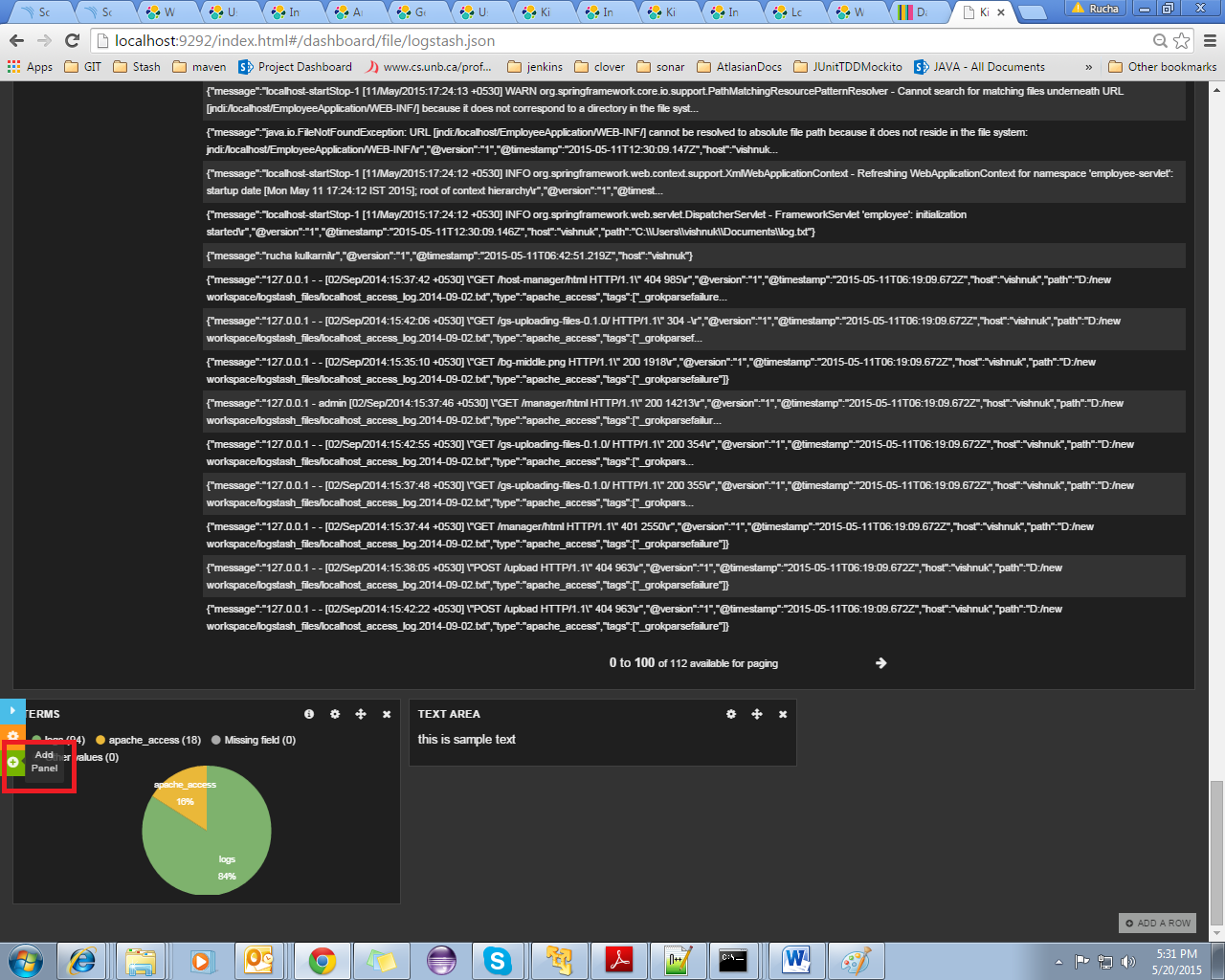
In panel settings, do the required settings.

For getting pie chart of the events, select ‘terms’ in ‘panel type’ and select ‘style’ as ‘pie’. In ‘parameters’ select field whose distribution is to be analyzed. Order can also be set. It can be set ‘count’ to display data according to size, or it can be ‘term’ to display data alphabetically.

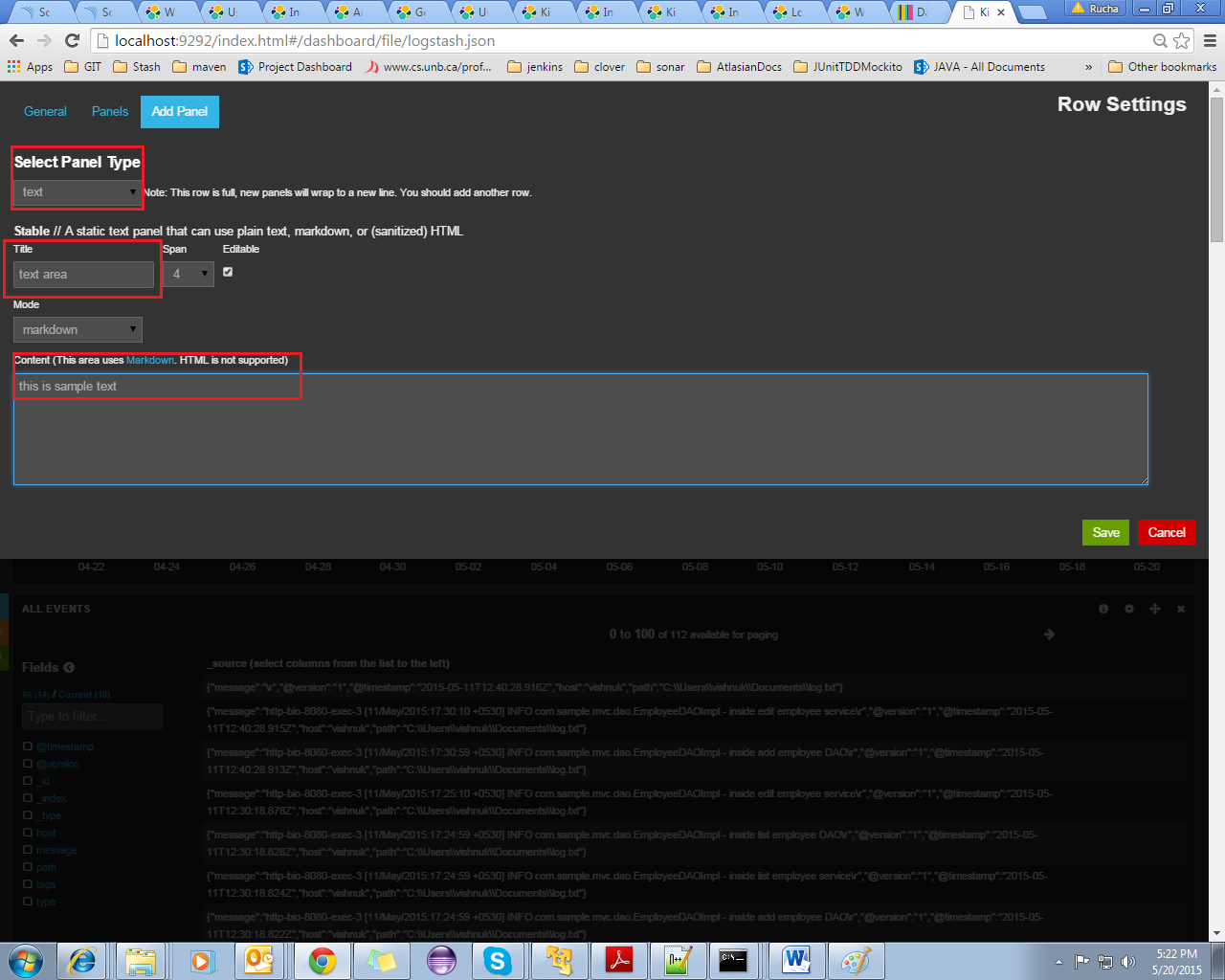
Save the configurations.

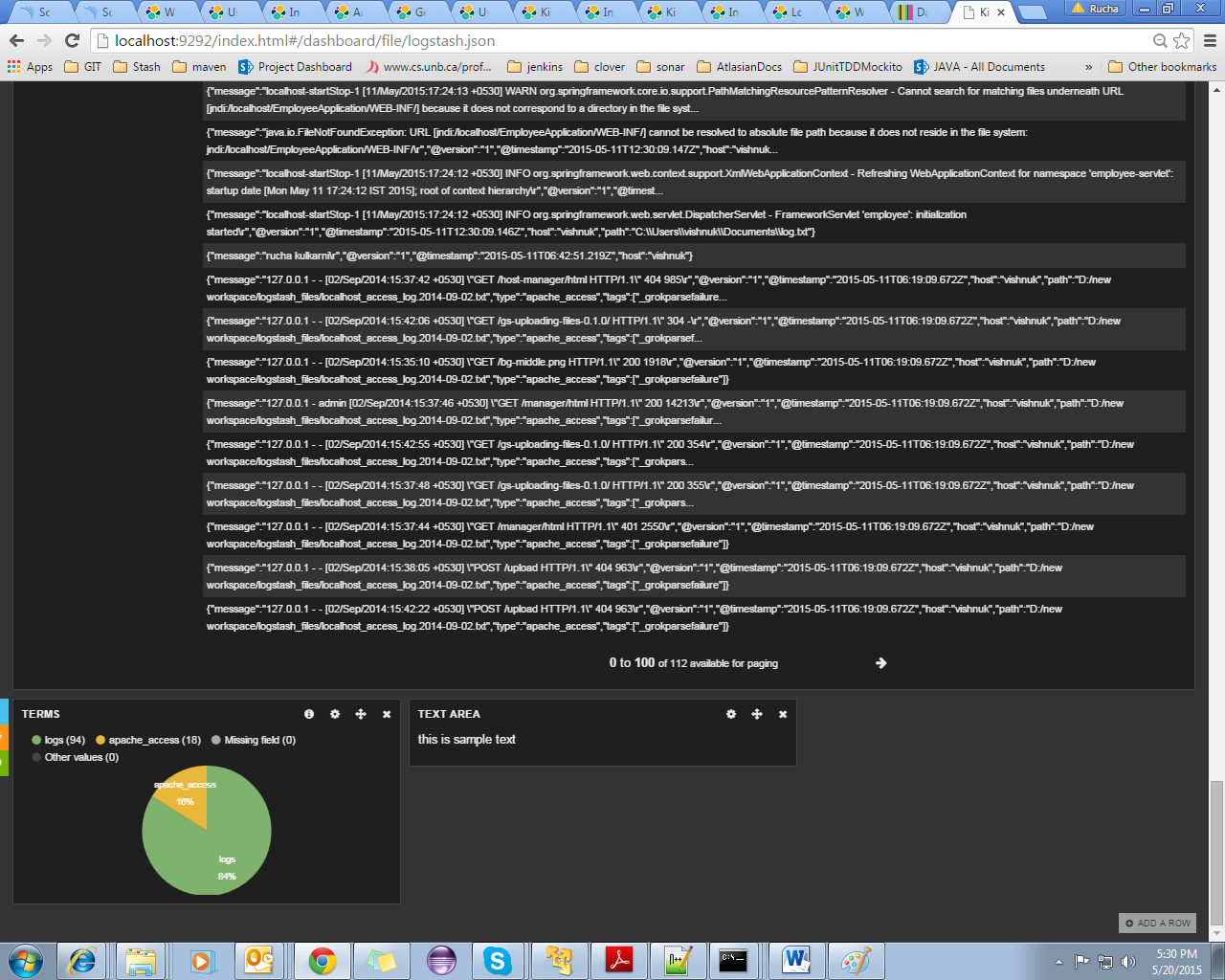


You can add new panel to row. Also here you can change row settings



Here I’ve created a ‘text’ panel for demo purpose.

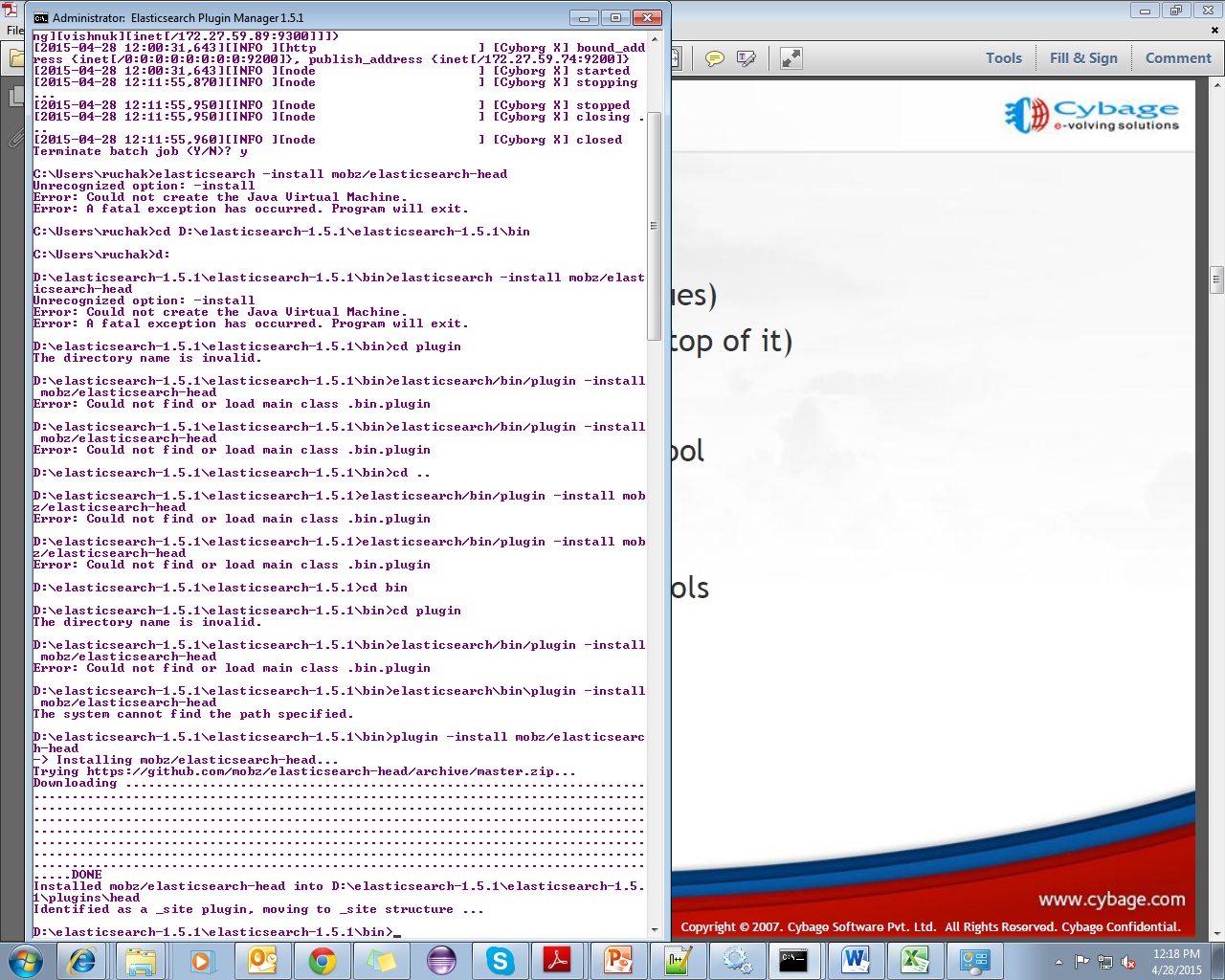




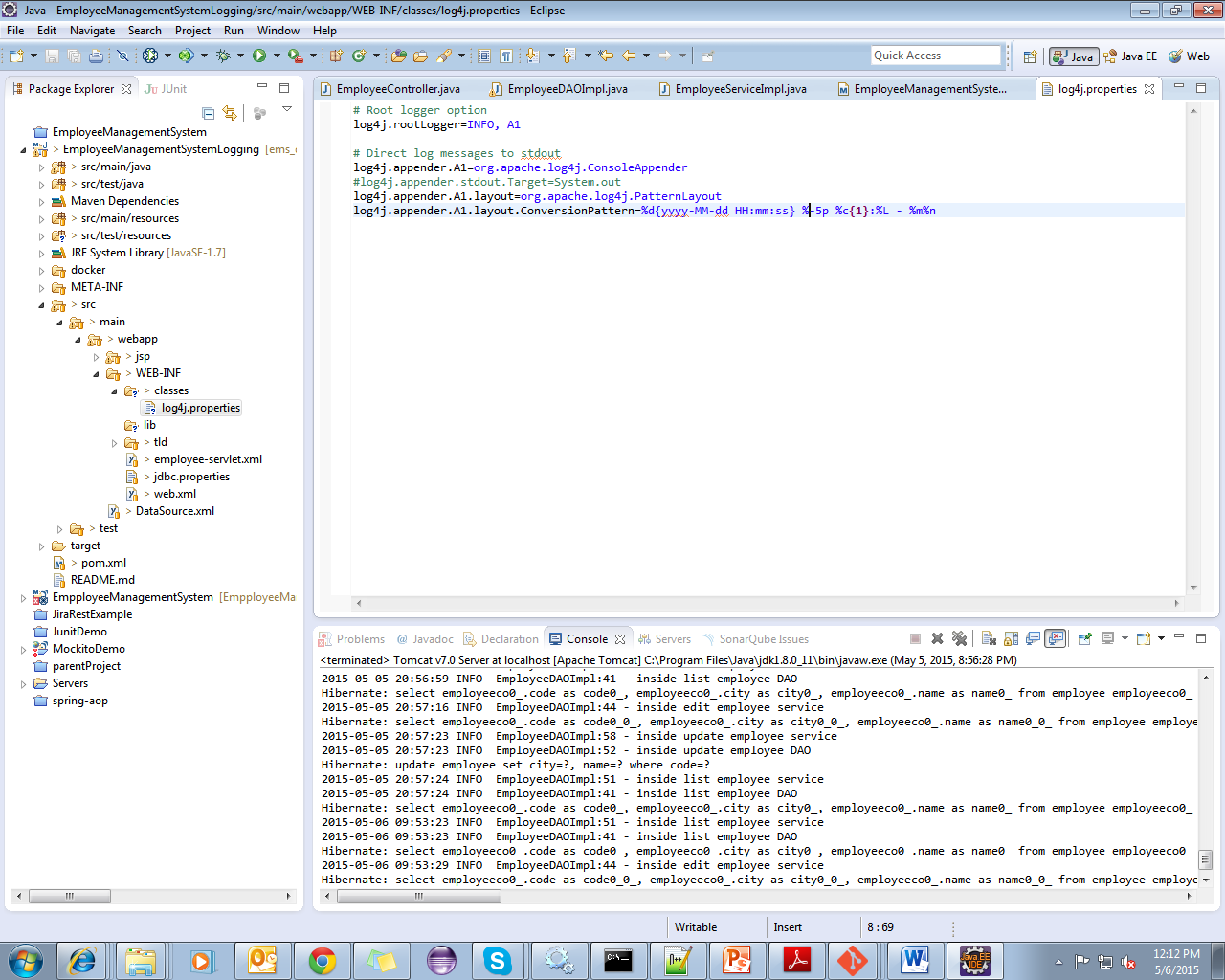
Thus, dashboards can be customized by adding and configuring rows and by adding panels to rows.

Once dashboard is created, save it.

# Additional information



Head plugin in elastic search



Log4j properties file for displaying logs on console