ELK Stack for easy Log Management

Project Report

Submitted by
Sai Rakesh Ghanta (BE/25039/12)

Bachelor of Engineering
in
Computer Science

Birla Institute of Technology, Mesra

November 5, 2015

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **ELK Stack for easy Log Management**submitted by studentsGhantaSaiRakesh(BE/25039/12), Piyush Choudhary(BE/25005/12) andMudit Agarwal(BE/25016/12) in partial fulfillment of the requirements for the award of the **Degree Bachelor of Engineering**in Computer ScienceEngineering is a bonafide record of the work carried out under my(our) guidance and supervision at Birla Institute of Technology, Mesra-Jaipur campus.

Mentor:
Mrs. Anju Sharma
Asst. Professor
Computer Science Department

Coordinator:
Dr. ShripalVijayvargiya
Associate Professor
Computer Science Department

Head of the Department:
Dr. (Mrs.) Madhavi Sinha
Associate Professor
Department of Computer Science

This project report was evaluated by us on _____

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of our professors and mentor. We would like to extend our sincere thanks to all of them.

We are highly indebted to Mrs. Anju Sharma for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project.

We would like to express our gratitude towards our parents for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also goes to our professorswho helped us in developing the project and people who have willingly helped us out with their abilities.

Table of Contents

1. Introduction

1.1. Purpose

Log management (LM) comprises an approach to dealing with large volumes of computer generated log messages (such as event-logs). In an enterprise, there are a huge number of machines such as Linux Machines, Routers, Apache Servers etc. which generate hundreds of logs every minute.

Log management is often considered a complex exercise. Understanding log management tends to be a slow and evolutionary process. In response to issues and problems, new SysAdmins use a combination of cat, tail and grep (and often sed, awk or Perl too) become the tools of choice to diagnose and identify problems in log and event data. But these are not efficient and the complexity involved in the procedure is high. So, a simple process of log management needs to be evolved.

1.2. Scope

By combining the open source technologies Elasticsearch, Logstash and Kibana, we can create an end-to-end stack that delivers actionable insights in real time from almost any type of structured and unstructured data source. Built and supported by the engineers behind each of these open source products, the Elasticsearch ELK stack makes searching and analyzing data easier than ever before.

It is an integrated framework for log collection, centralization, parsing, storage and search which has a wide range of input and output mechanisms.We can integrate Logstash with metrics engines, alerting tools, graphing suites, storage destinations or easily build your own integration to destinations in your environment.

1.3. Definitions, Abbreviations, Acronyms

Term	**Definition**
ELK stack	Elastic Search, Logstash, Kibana
SysAdmin	System Administrator/ Engineer
User	SysAdmin managing the logs

JSON	Java Script Object Notation
Log Details	Log message converted into JSON object with Machine id, Host Name, Path, Log Type and message.
Remote Nodes	All the remote agents connected in a network.
Time Frame	Time period (real or absolute) to be set to retrieve the logs and for the real-time analytics to be done.

1.4. Major constraints

One assumption is that this product is not worthy enough as the logs are already being stored in the server hosting the network. But, in that case real-time analytics on the logs cannot be performed. The main role of this product is to retrieve and manage the logs in the most effective way possible.

1.5. Overview

The remainder of this document includes five chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. Further, the chapter also mentions the system constraints and assumptions about the product. The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences. The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen. The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it.

2.1. Product perspective

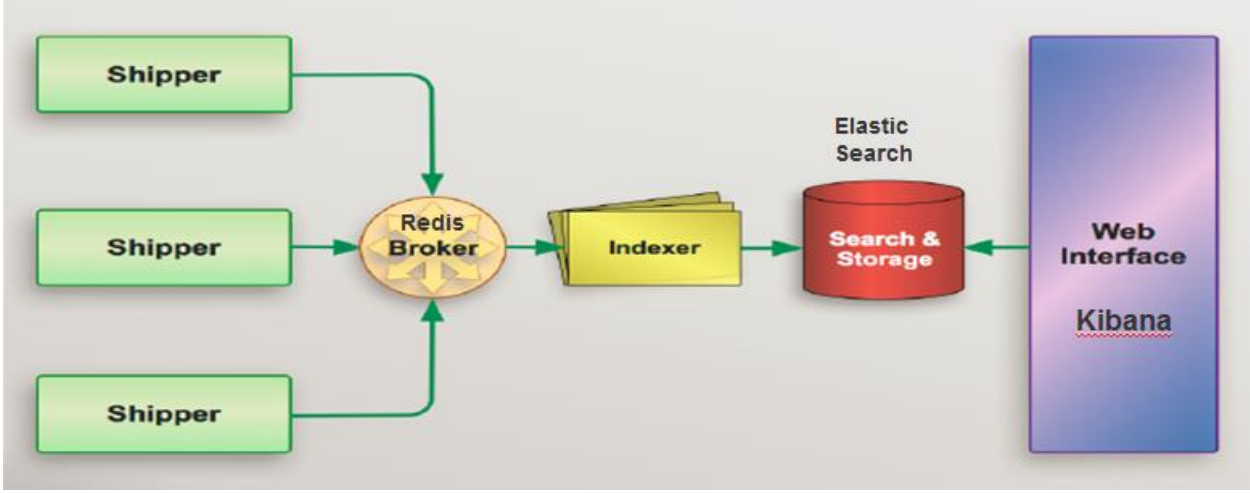
In the ELK stack ecosystem there are four components:

• Shipper: Sends events to Logstash. Your remote agents will generally only run this component.

• Broker and Indexer: Receives and indexes the events.

• Search and Storage: Allows you to search and store events.

• Web Interface: A Web-based interface to Logstash called Kibana.



Logstash servers run one or more of these components independently, which allows us to separate components and scale Logstash.

In most cases there will be two broad classes of Logstash host you will probably be running:

- Hosts running the Logstash agent as an event "shipper" that send your application, service and host logs to a central Logstash server. These hosts will only need the Logstash agent.

- Central Logstash hosts running some combination of the Broker, Indexer, Search and Storage and Kibana Web Interface which receive, process and store your logs.

2.2. Product functions

- Wide variety of input mechanisms - it can take inputs from TCP/UDP, files, Syslog, Microsoft Windows Event Logs, STDIN and a variety of other sources.
- Outputting data - Wide range of destinations, including TCP/UDP, email, files, HTTP and a wide variety of network and online services.
- We can integrate Log stash with metrics engines, alerting tools, graphing suites, storage destinations or easily build your own integration to destinations in your environment.
- Events flow in & log servers that hold the data.
- Integrating a light weight framework into remote nodes to ship logs without affecting their existing functionalities.
- Collection, centralization, parsing, storage and search of the logs of all the remote nodes connected in a network.

2.3. User Characteristics

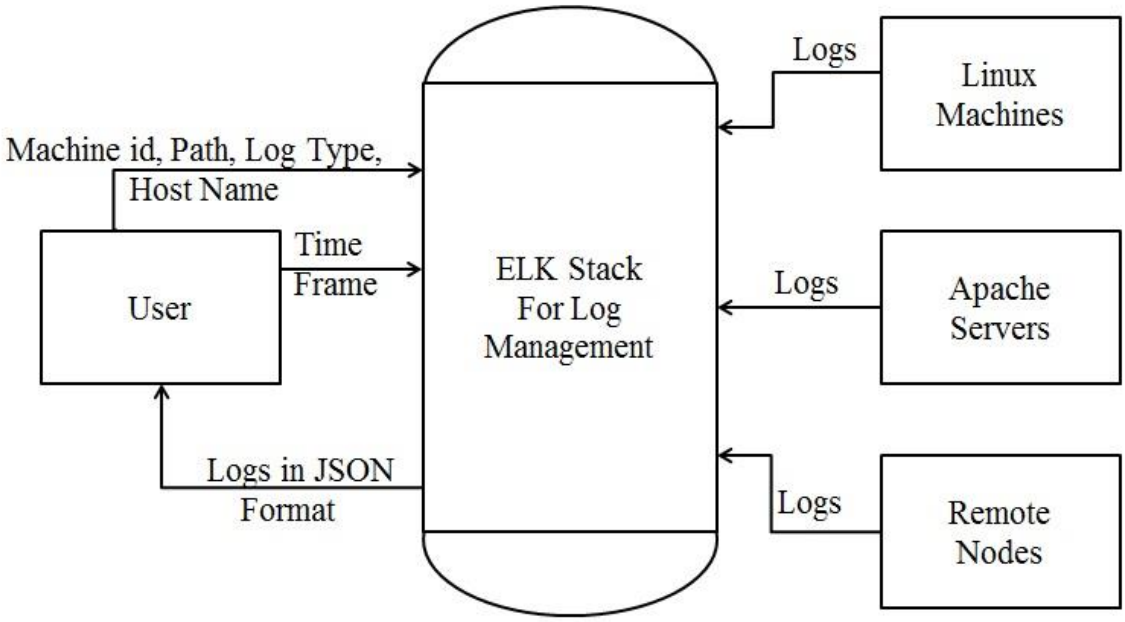
The type of users who use this product will be the SysAdmins (System Administrators/ Engineers). In response to issues and problems faced in the management of the logs, new SysAdmins use a combination of cat, tail and grep (and often sed, awk or perl too) become the tools of choice to diagnose and identify problems in log and event data. But these are not efficient and the complexity involved in the procedure is high. So, ELK stack simplifies the process of log management.

We may have tens, hundreds or even thousands of hosts. We run numerous, inter-connected applications and services across multiple locations and fabrics, both physically, virtually and in the cloud. In this world it quickly becomes apparent that logs from any one application, service or host are not enough to diagnose complex multi-tier issues.

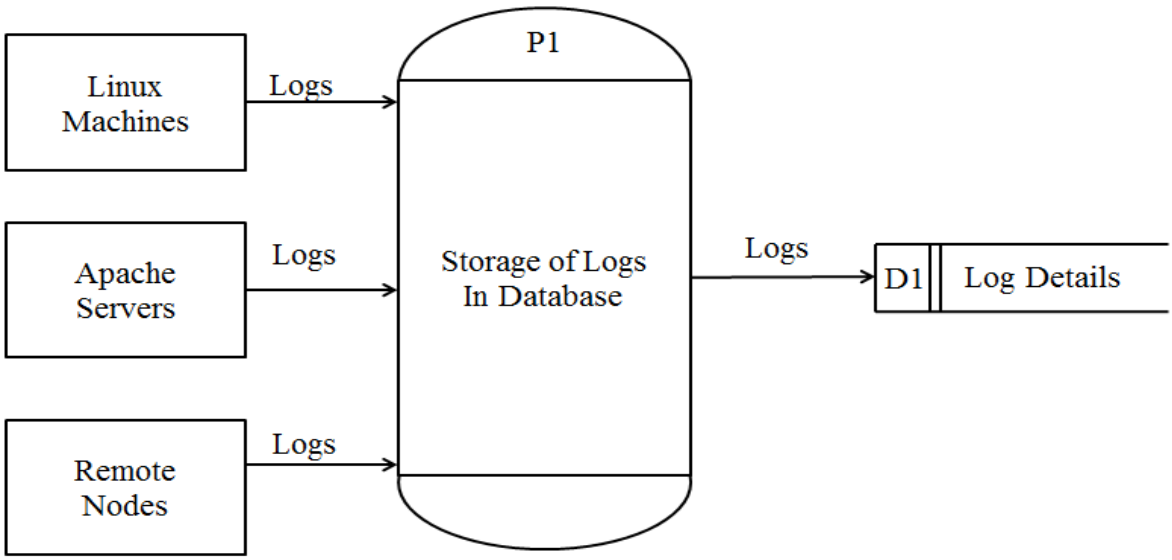
To address this gap your log environment must evolve to become centralized. The tools of choice expand to include configuring applications to centrally log and services like rsyslogand syslog-ngto centrally deliver Syslog output. Events start flowing in and log servers to hold this data are built, consuming larger and larger amounts of storage.

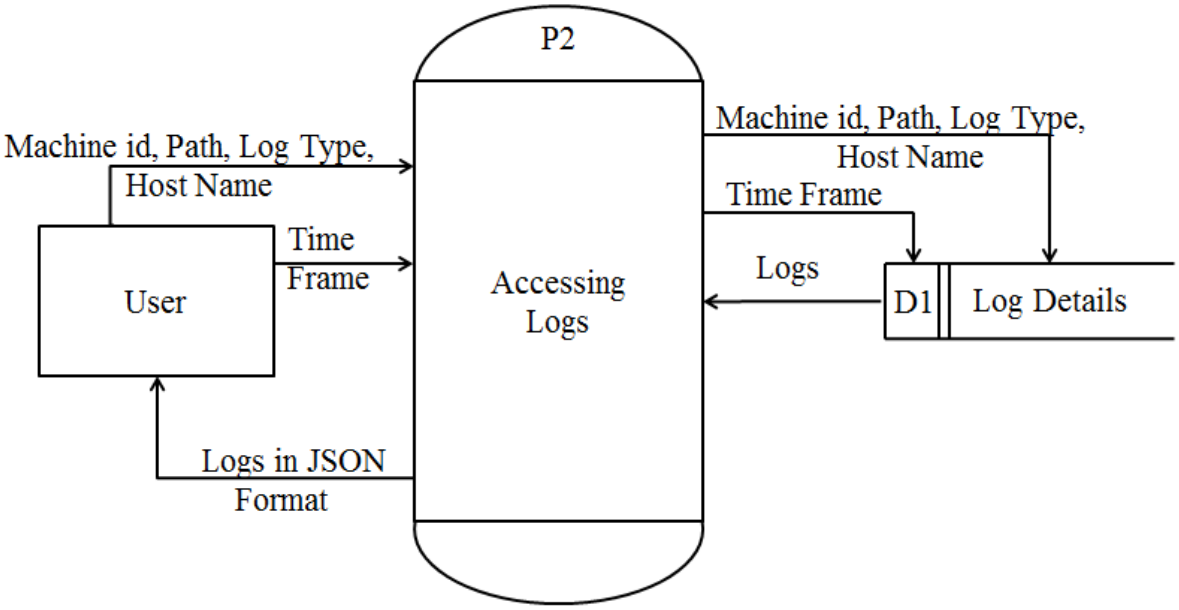
2.4. Data Flow Diagrams

Level 0



Level 1





2.5. Assumptions and Dependencies

One assumption is that this product is not worthy enough as the logs are already being stored in the server hosting the network. But, in that case real-time analytics on the logs cannot be performed. The main role of this product is to retrieve and manage the logs in the most effective way possible.

2.6 Apportioning of requirements

In the case that the project is delayed, there are some requirements that could be transferred to the next version of the product.

3. Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

3.1 External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1. User interface

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. We use Kibana to search, view, and interact with data stored in Elasticsearch indices. We can easily perform advanced data analysis and visualize your data in a variety of charts, tables, and maps.

Kibana makes it easy to understand large volumes of data. It's simple, browser-based interface enables you to quickly create and share dynamic dashboards that display changes to Elasticsearch queries in real time.From Kibana's Discover page, we can submit search queries, filter the results, and examine the data in the returned documents.We can construct visualizations of your search results from the Visualization page. Each visualization is associated with a search.We can save and share visualizations and combine them into dashboards to make it easy to correlate related information.

3.1.2. Software interfaces

Remote node software (shipper):

- Logstash Agent
  - ✓ Shipper is a Java based framework.
- Non-Logstash Agent
  - ✓ Beaver - It is a Python based light weight framework.
  - ✓ HiRedis - It is a C-client library.

Broker (Redis):

- It essentially acts as a "buffer" between our Logstash agents and our central server. It is useful for two reasons:

1. It is a way to enhance the performance of our Logstash environment by providing a caching buffer for log events.
2. It provides some resiliency in our Logstash environment. If our logstash indexing fails then our events will be queued in Redisrather than potentially lost.

❖ We could choose a variety of possible brokers like AMQP and 0MQ, but we're going with Redis because:

- It's very simple and very fast to set up.
- It's performant.
- It's well tested and widely used in the Logstash community.

Search & Storage (Elastic search):

- Elasticsearch is a distributed, open source search and analytics engine, designed for horizontal scalability, reliability, and easy management.
- Distributed, scalable, and highly available.
- Real-time search and analytics capabilities.
- Sophisticated RESTful API.

3.2. Functional Requirements

3.2.1. Remote Node

- On the remote node, the Logstash runs as a service which creates the pipeline (such as a simple server-client connection) to the central server for the logs to be shipped.
- Integrating a light weight Shipper consisting of I/O plugins which has paths of the logs to be monitored & details of the central Host into remote nodes to ship logs to a central log server without affecting their existing functionalities.

3.2.2. Central Server

- In the central log server, the logs shipped by the shipper are received and indexed on a millisecond (time stamp) basis.
- After indexing the logs, they are stored in the Data Base for the Real-Time Analytics to be done on the logs through the web-interface, i.e.; the front end.

4. Data Design

Algorithms and Details on Data Structures:

1) Conjunctions - implemented by Sort by cost & leap frog(modified version of Verlet Algorithm) algorithms

2) Cardinality Aggregation - A single-value metrics aggregation that calculates an approximate count of distinct values. Values can be extracted either from specific fields in the document or generated by a script.

3) Regexp Query - The regexp query allows you to use regular expression term queries. See Regular expression syntax for details of the supported regular expression language. The "term queries" in that first sentence means that Elasticsearch will apply the regexp to the terms produced by the tokenizer for that field, and not to the original text of the field.

4) Numeric doc values compression.

5. Architectural and component-level design

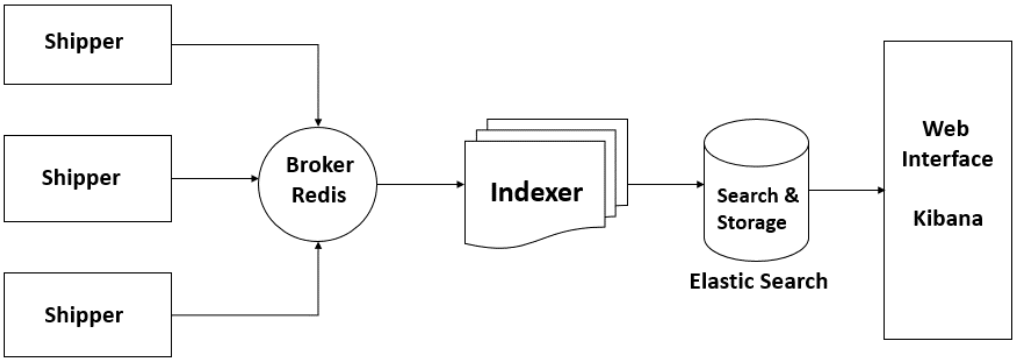
This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it.

5.1 Program Structure

In the ELK stack ecosystem there are four components:

• Shipper: Sends events to Logstash. Your remote agents will generally only run this component.
• Broker and Indexer: Receives and indexes the events.
• Search and Storage: Allows you to search and store events.
• Web Interface: A Web-based interface to Logstash called Kibana.

5.1.1 Basic Architecture diagram



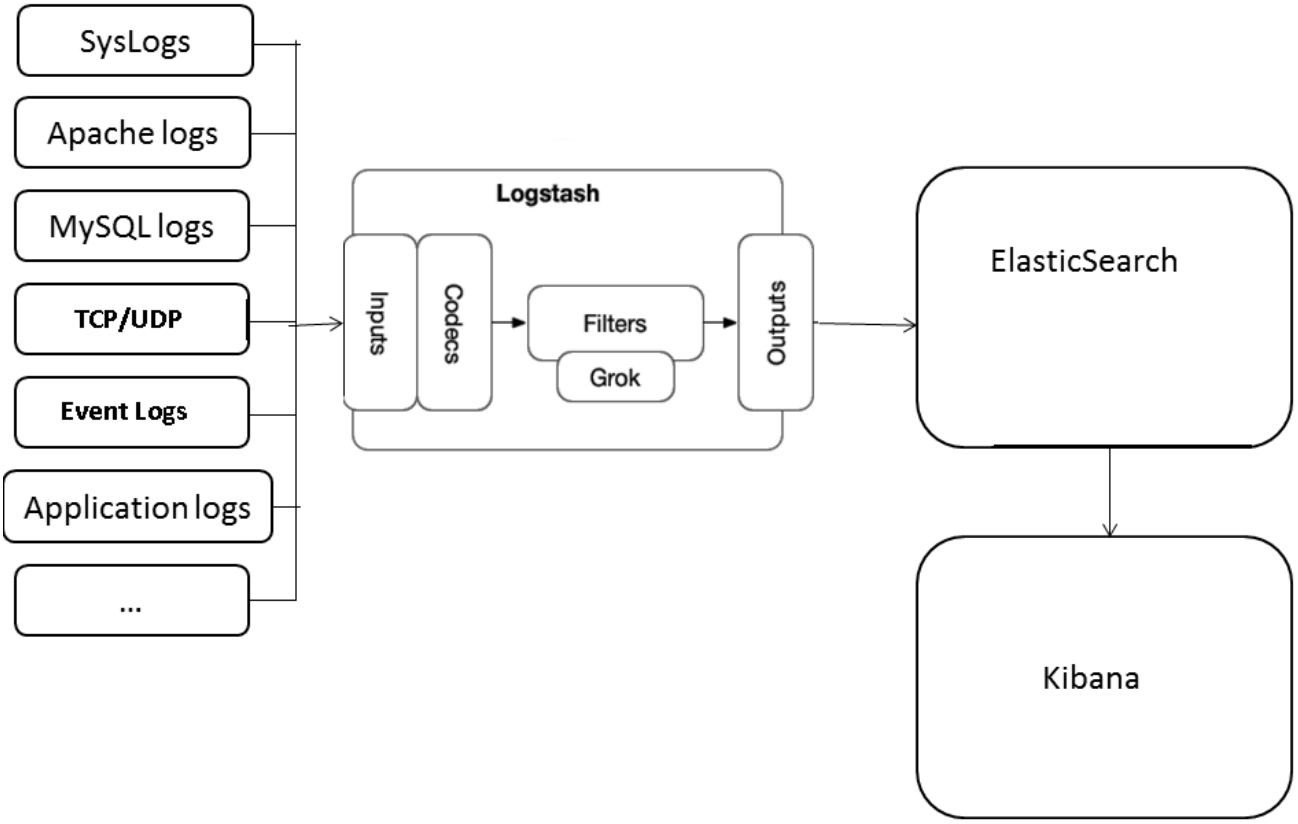
Logstash servers run one or more of these components independently, which allows us to separate components and scale Logstash.

In most cases there will be two broad classes of Logstash host you will probably be running:

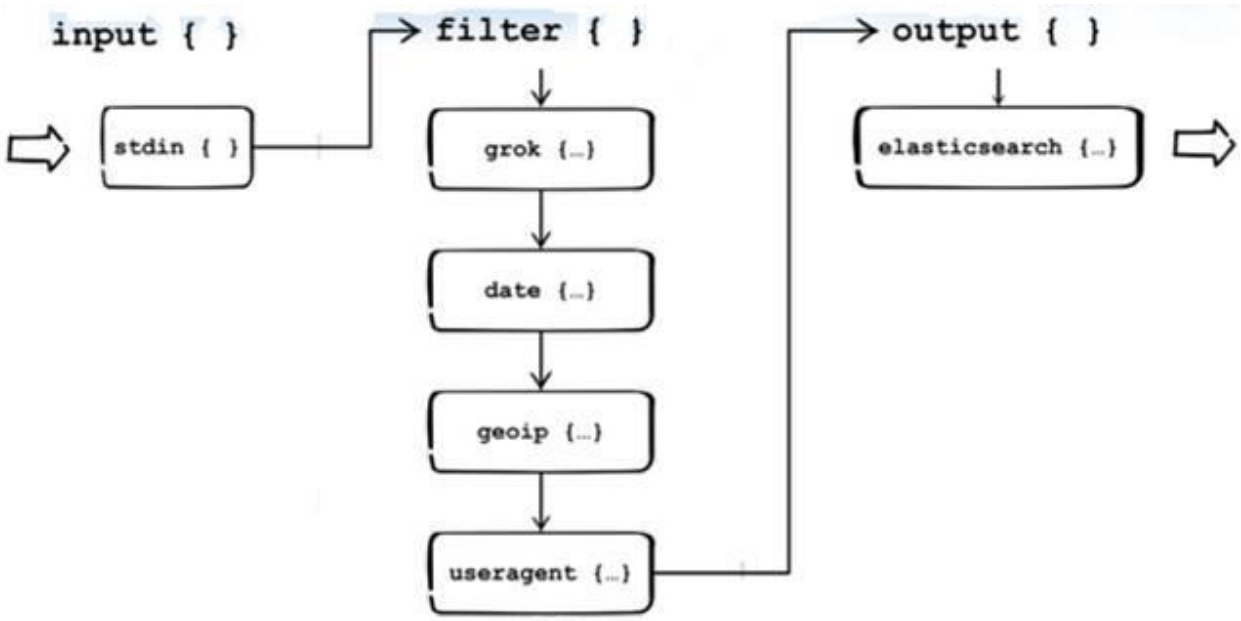
- Hosts running the Logstash agent as an event "shipper" that send your application, service and host logs to a central Logstash server. These hosts will only need the Logstash agent.

- Central Logstash hosts running some combination of the Broker, Indexer, Search and Storage and Kibana Web Interface which receive, process and store your logs.

5.1.2 Detailed Architecture Design

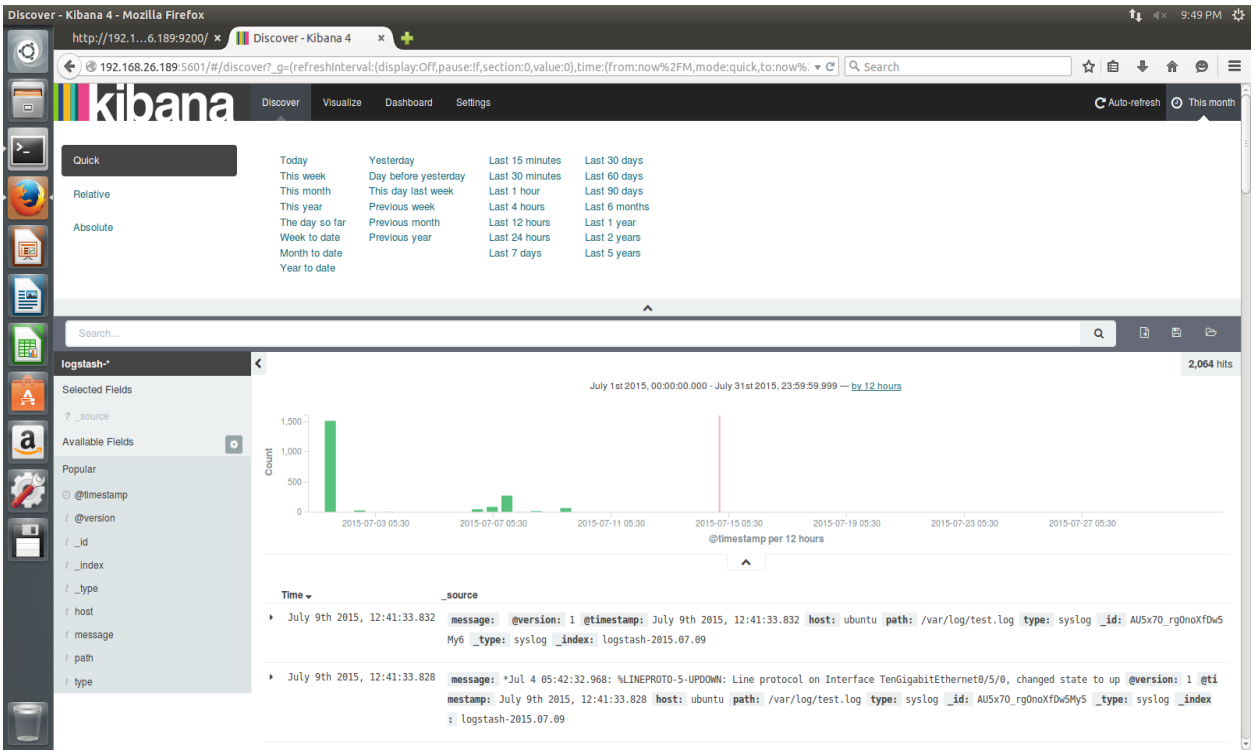


5.1.3 Plug-in Architecture Design



6. User interface design

- Kibana is an open source data visualization platform that allows you to interact with your data through stunning, powerful graphics that can be combined into custom dashboards that help you share insights from your data far and wide.
- Flexible analytics and visualization platform.
- Real-time summary and charting of streaming data.
- Intuitive interface for a variety of users.
- Instant sharing and embedding of dashboards.
- The Kibana web interface is a customizable dashboard that you can extend and modify to suit your environment. It allows the querying of events, creation of tables and graphs as well as sophisticated visualizations.



7. Implementation

Here, in our solution the implementation is done through the integration of three different open source technologies namely Elasticsearch, Logstash, Kibana. We write different configuration files using Linux Shell Scripting and Codec plugins. Using these configuration files, we achieve the integration of the database(Elasticsearch), Indexer(Logstash) and Front-end(Kibana). By this integration we form the ELK Stack framework through which Log Management can be made easy.

Here, we managed to attach some of the configuration files & plugins we designed in our implementation process.

7.1. Configuration Files

7.1.1. Shipper.conf

```
input {
file {
type => "syslog"
path => ["/var/log/dmesg", "/var/log/kern.log","/var/log/test.log"]
exclude => ["*.gz"]
}
}
output {
stdout { }
redis {
host => "72.163.178.147"
data_type => "list"
key => "logstash"
}
}
```

The function of the shipper is to capture the log instances and generate a stream of log data for the Logstash indexer. It is a plugin based Java configuration file, and is a Logstash agent. However, java-based configuration files require a greater amount of memory. The other alternatives are non-Logstash agents such as HiRedis (C-Client library) and Beaver (Python based light-weight framework).

7.2.2. Central.conf

```
input {
redis {
host => "127.0.0.1"
type => "redis-input"
data_type => "list"
key => "logstash"
message_format => "json_event"
}
}


output {
stdout{ }
elasticsearch {
host => "127.0.0.1"
cluster => "logstash"
}
}
```

8. Recent Developments and Further Steps

We implemented the Server Side of the proposed framework, coded the required configuration scripts and plug-ins, configured and set up the Central Server where the collection, parsing and analytics of logs is performed. Central Server is the heart of the project and forms a major part of the proposed solution.

We will be infusing the concepts of Grokfilters(Pattern Recognition) with Data Analytics (ELK Stack) to form an efficient, cheap and scalable log management framework that can solve today's enterprise issues.We are working towards contribution to the open source Elasticsearch. We are hopeful and confident that we can achieve this by the end of the next semester.

9. Conclusion

The presented solution integrates some of the newest and popular open-source technologies, to tackle the problems that many enterprises are facing in Log Management. To sum up:

- It improves the efficiency of the system by using real-time analytics from concurrently operating virtual machines in the cloud. This speeds up the log management process.
- This system makes the data in the logs indexed, structured and easily searchable.
- It also improves accessibility, as logs can even be accessed from remote locations.
- Using Kibana, it provides an easy-to-use web-based interface to retrieve the logs from the server.

While all the technologies used in this solution are open-source and thus highly customizable, a few alternatives to the software used are present:

- Splunk can be used as an alternative to ELK Stack, but Splunk is a license based software. The choice of application can be decided based on merits and cost of each alternative.
- Containers are popular in recent times but, Docker, Virtual Appliances and other virtualization techniques can also be used as alternatives.

10. Bibliography

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[2] James Turnbull, "The Logstash Book – Log management made easy", version v1.5.1(e1ce5d6), 2015.

[3] ELK Stack: http://blog.qbox.io/welcome-to-the-elk-stack-elasticsearch-logstash-kibana

[4]ELK Stack Installation: https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-4-on-ubuntu-14-04