

Building a Cloud-based Log Processing System

Lingxiao Xia



1.

Modern Systems Are Complicated

Really, really complicated

- ◆ Tech team
 - ◆ Frontend javascript masters
 - ◆ Restful API masters
 - ◆ Website designers
 - ◆ Devops
- ◆ But what about other departments of the company?

Really, really complicated

- ◆ Imagine having to write a server for every department of the company
 - ◆ Company as a service
 - ◆ Sales
 - ◆ Marketing
 - ◆ Accounting
 - ◆ Customer Service
- ◆ Quite a horrible image I am getting...

Really, really complicated

- ◆ Luckily there are other companies who serve those departments professionally
- ◆ Zoho, Intercom, Stripe, Xero...
- ◆ Now we just need to connect all these services together

How?

Here comes the log



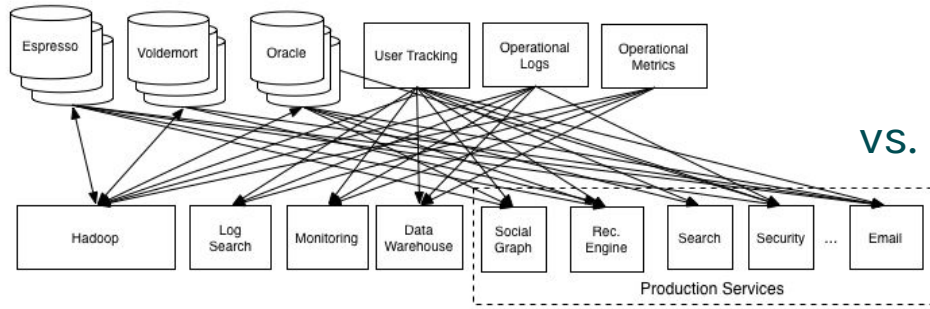
2.

Log-centric Approach

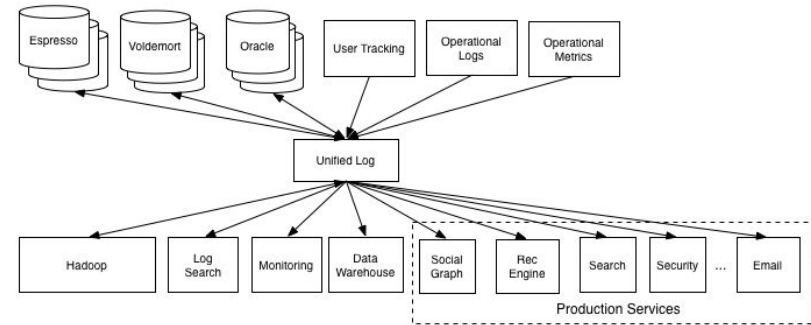
We call it log, but what is log

- ◆ It is really just data
- ◆ Oh but better, data with timestamps
- ◆ The potential is endless...

Log centric architecture



VS.



Source: [The Log: What every software engineer should know about real-time data's unifying abstraction]

(<https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying>)

Apache Kafka

Fast, cheap and scalable

The central goal of the project

Reliable

Production level reliability, if you configure it correctly

Reliable buffer

Only need to consume at a speed larger than the average input speed

That is not the end of it, though

Need to find a way to deploy it
reliably, on the cloud, again
and again



Open source community comes to the rescue

Ansible

IT automation tool. Agentless...

Docker

Containerize all servers for enhanced reliability

Kafka

Log, log, log, buffered log

Elasticsearch

Persistent data store

Logstash

The actual worker

Kibana

UI

The background consists of several overlapping, semi-transparent geometric shapes in various shades of green and teal. A large, dark teal shape is at the top, with a lighter green shape below it. A large, medium teal shape is in the center, and a dark teal shape is at the bottom. The overall effect is a layered, abstract landscape.

3.

The Actual System

DockReg

- ◆ Ansible Orchestrated Docker Registry 2.0
- ◆ Git: <https://github.com/xialingxiao/dockreg.git>

AnDoKaElk

- ◆ Ansible Orchestrated Kafka and Elasticsearch Cluster
- ◆ Git: <https://github.com/xialingxiao/andokaelk.git>