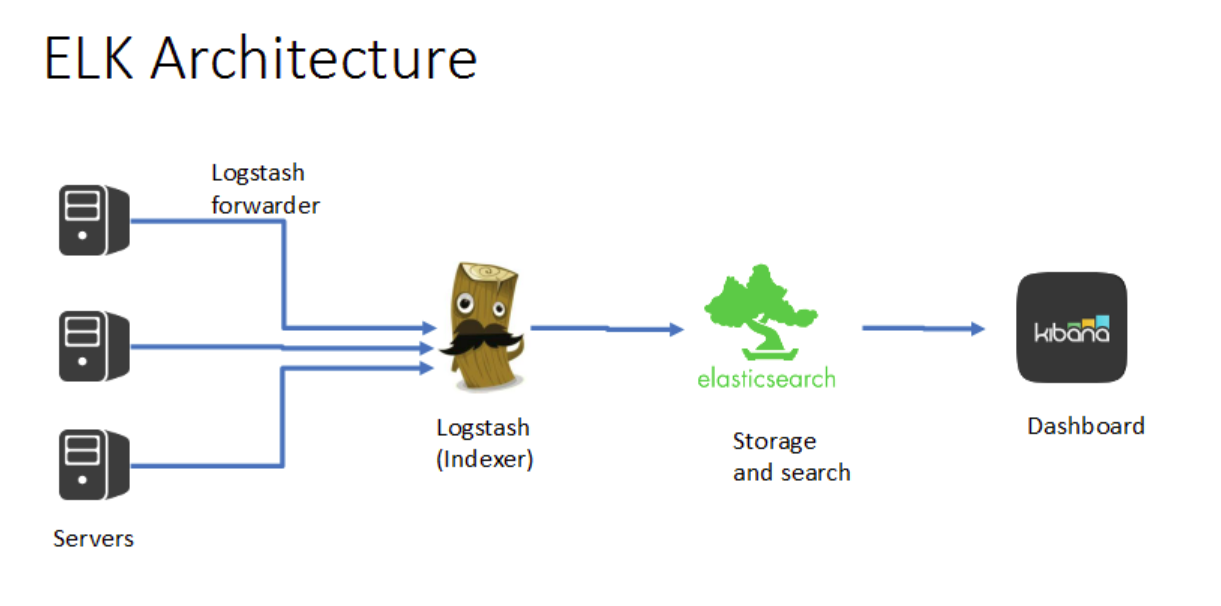
**Cloud Monitoring Service**

**POC Using ELK Stack**

* **The general implementation flow for creating logging architecture is:**

1. Create log files for all instances in microservices in defined formats.
2. Install filebeat on logging instance.
3. Install Elasticsearch.
4. Install Kibana.
5. Install logstash on a comman instance.
6. Edit configuration file, (filebeat.yml for filebeat and logstash\_sample.config for logstash)
7. Write filter in logstash to convert log data to meaningful json objects and configure logstash to push data to elasticsearch.
8. Visualize the data pushed to elasticsearch using kibana.

* **The general architecture for monitoring logs is :**



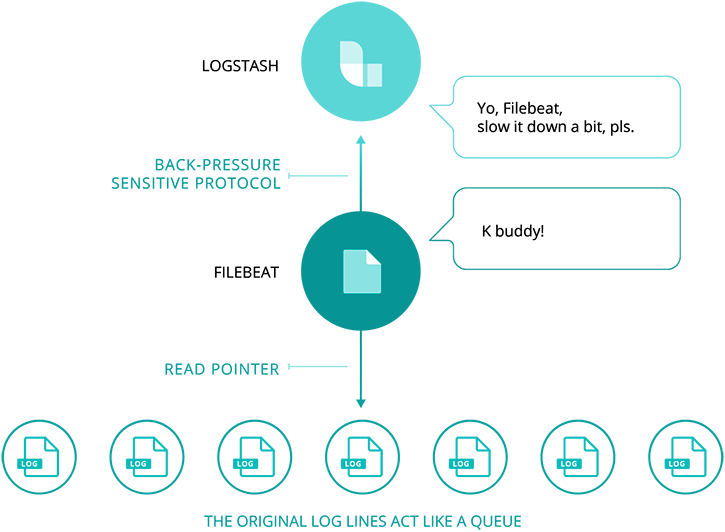
#### **Create log files for all instances in microservices in defined formats:**

* Log File: Records the events that occured in the operating system, software runs and the communication between the users.
* Log= timestamp+data
* The log files created by the services are the basic content generators in the logging infrastructure.
* The format of this logging files should be fixed and they should contain all the necessary information required for visualization in kibana.
* The general format for log files is defined as :
  + LOGLEVEL : ERROR | WARNING | INFO
  + Service name
  + IP address : For unique identification of instance
  + Timestamp: Sequence of characters identified when a event is occured
  + API : To identify which API is logging this log. (for technical troubleshooting)
  + API KEY : To identify which hub is the data stream related to. (for technical troubleshooting)
  + User identifier: Username / email address (for business analysis )
  + User activity: eg. Login, Logout etc. ( for business analysis )
  + CPU RAM and DISK usage : To be printed after specific interval
  + Description : Any other text with respect to the log line.
* The above mentioned fields are general guidelines as to what content must go in the log files for better visualization. Any other fields can be added/removed according to requirement.
* For sample application, we are using a sample.log file
* A sample log file should look like this :



#### **Installing Filebeat on logging instance:**

* Filebeat acts as a data shipper which ships the logged data to logstash.
* It runs as an agent on the logging instance and tails the log files to send new data to logstash.
* Filebeat uses a back pressure-sensitive protocol when sending data to Logstash or Elasticsearch to account for higher volumes of data.
* If Logstash is busy crunching data, it lets Filebeat know to slow down its read. Once the congestion is resolved, Filebeat will build back up to its original pace and keep on shipping.



* To install filebeat:
  + <https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.0.1-amd64.deb>
  + sudo dpkg -i filebeat-6.0.1-amd64.deb
* Configure filebeat to send data to logstash:
  + Configuration file for filebeat is at /etc/filebeat/filebeat.yml.
  + This file should look like :

filebeat.prospectors:

- type: log

paths:

* home/khushbu/Desktop/dattus/data.log

output.logstash:

hosts: ["localhost:9600”]

* To run filebeat :

sudo filebeat -e -c filebeat.yml -d "Publish"

#### **Install Elasticsearch:**

* Elasticsearch starts working after logstash.
* Elasticsearch takes input from the logstash and sends output to the kibana for visualization.
* Elasticsearch stores all the data from logstash in json under different indices.
* This is the actual database where data is stored for visualization.
* Build on top of **lucene**
* Lucene: The Apache lucene is a open source library for information retrieval from structured data.
* Elasticsearch is document-oriented: It stores real world complex entities as structured JSON documents and indexes all fields by default, with a higher performance result.
* To install elasticsearch:
  + wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.0.1.zip
  + unzip elasticsearch-6.0.1.zip
  + cd elasticsearch-6.0.1/
* Run elasticsearch:
  + ~/elasticsearch-6.0.1/bin/elasticsearch

#### **Install Logstash:**

* Logstash is an open source data processing Pipeline.
* Logstash take input from Beats.
* Logstash is a server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to Elasticsearch.
* Download from <https://artifacts.elastic.co/downloads/logstash/logstash-6.0.1.zip>
* Unpack the downloaded package.
* Logstash uses configuration file from command line with the -f option.
* A sample configuration file for logstash will look like :

input {

beats {

port => "9600"

}

}

filter {

grok {

patterns\_dir => ["/home/khushbu/logstash-6.0.0/patterns"]

match => { "message" => "%{DATESTAMP:time} %{IP:ipaddress} %{LOGLEVEL:loglevel} %{SERVICE:service} %{OPERATION:operation} %{APIKEY:apikey} %{GREEDYDATA:description}"}

}

}

output {

# stdout { codec => rubydebug }

elasticsearch {

hosts => [ "localhost:9200" ]

index => "dataproducer-%{+YYYY.MM.dd}"

}

}

* To run logstash:

logstash6.0.1/bin/logstash -f second-pipeline.conf --config.reload.automatic

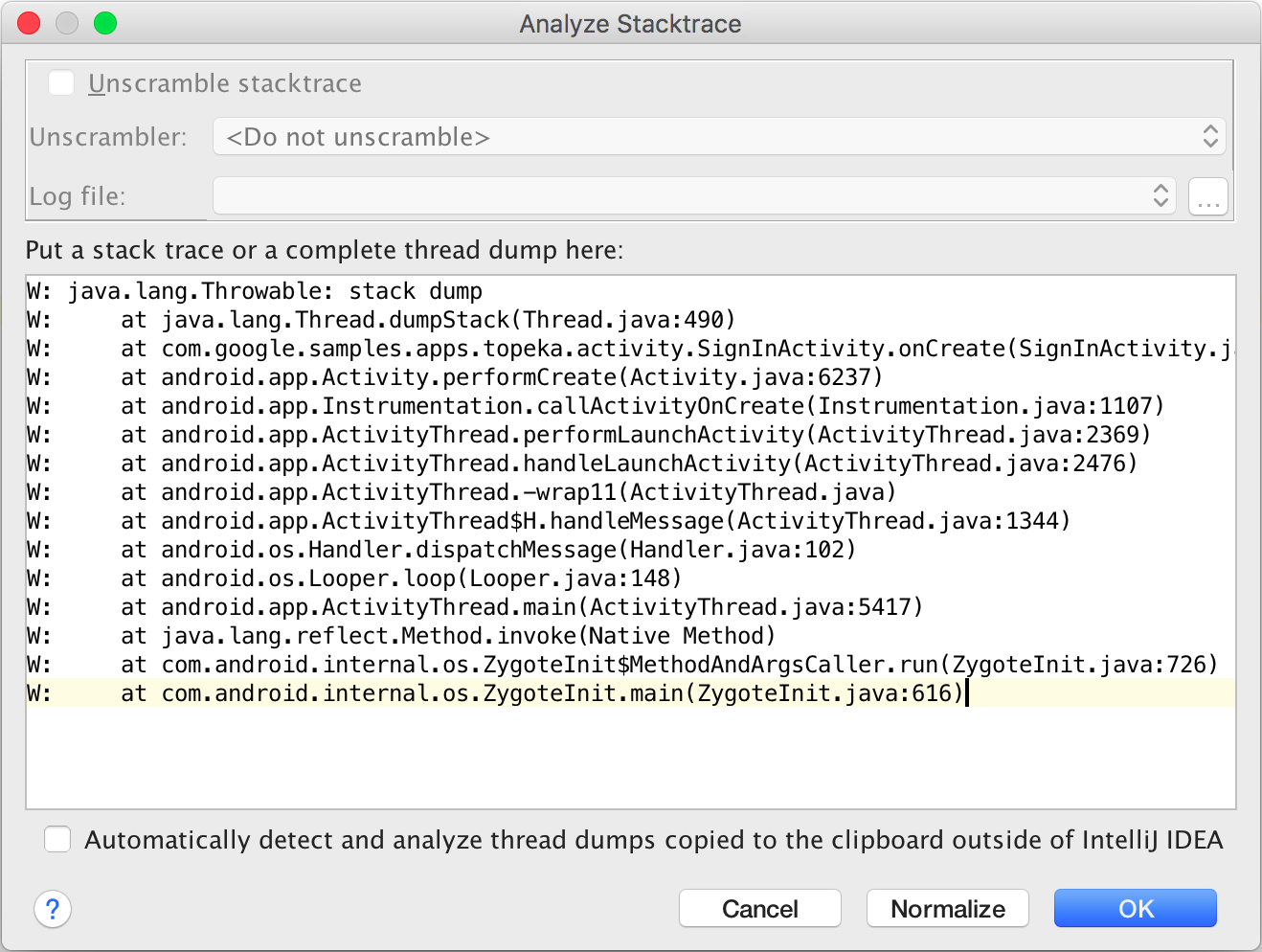
* Run this when you have already started elasticsearch.

#### Writing filter for log data:

* A filter plugin performs intermediary processing on an event.
* Filters are often applied conditionally depending on the characteristics of the event.
* Writing filters helps us to parse the log files into meaningful JSONs with relevant fields.
* One such filter provided by logstash is the grok filter.
* It helps us find patterns in log files and assign fields to them.
* **Grok** is a great way to parse unstructured log data into something structured and queryable.
* A detailed document on grok is [here](https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html).
* We have used grok for parsing log files.

#### Handling stack trace:

* **Stack Trace** is a list of method calls from the point when the application was started to the point where the exception was thrown.



* Stack trace of logs can be handled by adding multiline support in filebeat.
* Add the following lines to /etc/filebeat/filebeat.yml

multiline.pattern: '^\['

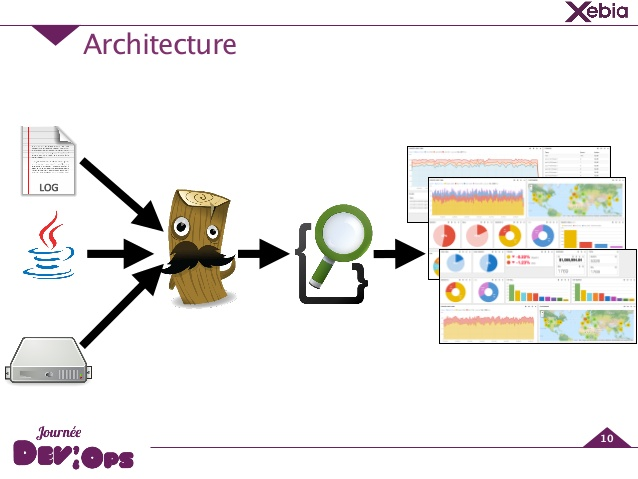
multiline.negate: true

multiline.match: after

* The configuration treats the bracket ‘[‘ as a new line and does not end log stream on new line.
* Care must be taken that every new logger statement starts with ‘[‘.
* Details are available [here](https://www.elastic.co/guide/en/beats/filebeat/master/multiline-examples.html).

#### **Install Kibana:**

* Kibana works as a dashboard to visualize data present on elasticsearch.
* We use kibana UI for data visualization.
* Visualisation can be done on Kibana.
* Kibana takes results from Elasticsearch.
* The output can be visualised diagrammatically on kibana.
* Create a new index pattern for the data using management tool option in kibana.
* Management tool option helps to edit the pattern.
* Adjust timeline to view data on kibana.
* Add filters to get relevant data.
* Use graphical representations according to requirement and place them on dashboard for better understanding of data.



* To install kibana:
  + wget https://artifacts.elastic.co/downloads/kibana/kibana-6.0.1-linux-x86\_64.tar.gz
  + tar -xzf kibana-6.0.1-linux-x86\_64.tar.gz
  + cd kibana-6.0.1-linux-x86\_64/
* To run kibana:
  + ~/kibana-6.0.1/bin/kibana
* Go to localhost: 9600
* Work on the UI for creating relevant visualizations.
* Look into [official document for reference.](https://www.elastic.co/guide/en/kibana/current/getting-started.html) :
* **Official documentations:**
* [Filebeat](https://www.elastic.co/products/beats/filebeat)
* [Logstash](https://www.elastic.co/guide/en/logstash/current/index.html)
* [Elasticsearch](https://www.elastic.co/guide/en/elasticsearch/reference/6.0/index.html)
* [Kibana](https://www.elastic.co/guide/en/kibana/current/introduction.html)

#### **Installation on test setup:**

* Filebeat config file : /etc/filebeat/filebeat.yml
* Logstash config file : /opt/logstash-6.0.1/bin/first-pipeline.conf
* Elasticsearch installed: ~/
* Kibana installed: ~/
* Logstash installed: /opt/
* Grok pattern list: /opt/logstash-6.0.1/pattern/