elastic Meetup Night Class: Centralized Logging

Motivation

- Aggregate log events
  - across applications
  - across processes
  - across machines
- Combine and visualize data

Use Cases

- Business
  - amount and duration of transactions
  - history of operations
  - correlation between applications
- Performance and health
  - amount and duration of requests, request times
  - errors
  - correlation between systems, data bases, applications
- Security
  - fraud detection, intrusion detection
  - tracing

Overview

Beats

- Lightweight agent that pushes data to another destination
- Common for centralized logging: Filebeat
- Implemented in Go
- Other beats:
  - Metricbeat
  - Winlogbeat
  - Packetbeat

Logstash

- Kind of ETL system for transforming event data
- Often used for logs
- Written in JRuby
- Configured in Ruby Syntax
  - Input
  - Filter
  - Output

Elasticsearch

- Search engine
- Can store big amounts of data
- HTTP interface for storing and querying data
- Query DSL for accessing the data

Elasticsearch

- Stores the log events
- Often time based indexing
  - filebeat-2017-03-23
  - Index pattern filebeat-*

Kibana

- Web interface for analyzing data
- Explore data
- Generates requests to Elasticsearch
- Visualizes results: diagrams, maps
- Written in JavaScript

Reading files: Filebeat

- Installation as service or standalone
- Configuration in filebeat.yml
- Prospectors define where to read data from
- Output determines where to send data to

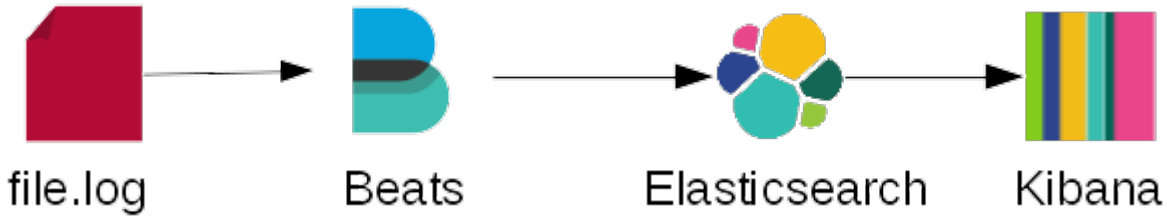
Filebeat: Prospector configuration

```
filebeat.prospectors:

- input_type: log

  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*
```

- Includes, excludes
- Multiline configuration

Filebeat: Setup options

- Using Ingest Node to process data



- Using Logstash to process data

Filebeat: Output configuration

- Sending data to Elasticsearch
  - for storing the message as a whole
  - or for processing with Ingest pipeline

```
output.elasticsearch:
  hosts: ["localhost:9200"]

  # Optional protocol and basic auth credentials.
  # username: "elastic"
  # password: "changeme"
```

Filebeat: Output configuration

- Sending data to Logstash
  - for processing and redistribution

```
output.logstash:
  # The Logstash hosts
  hosts: ["localhost:5044"]
```

Reading files: Logstash

- 3 major sections
  - Input: Specifies where to read data from
  - Filter: Different ways to massage the data
  - Output: Data sinks
- Countless plugins for all

Logstash: Input

- Server process for reading beat data

```
input {
  beats {
    port => 5044
  }
}
```

- or reading from file

```
input {
    file {
        path => "/var/log/apache2/access.log"
    }
}
```

Logstash: Grok filter

- Common filter: Grok
  - Regular expressions with predefined patterns

```
filter {
    grok {
        match => { message => "%{COMBINEDAPACHELOG}" }
    }
}
```

Logstash: Output

- Sending data to Elasticsearch

```
output {
    elasticsearch {
        host => "localhost"
        protocol => "http"
    }
}
```

- For debugging

```
    stdout {
      codec => rubydebug
    }
```

Grok

- Online tools for creating grok expressions
  - grokdebug https://grokdebug.herokuapp.com/
  - Grok Constructor http://grokconstructor.appspot.com/
- Careful with expensive regular expressions

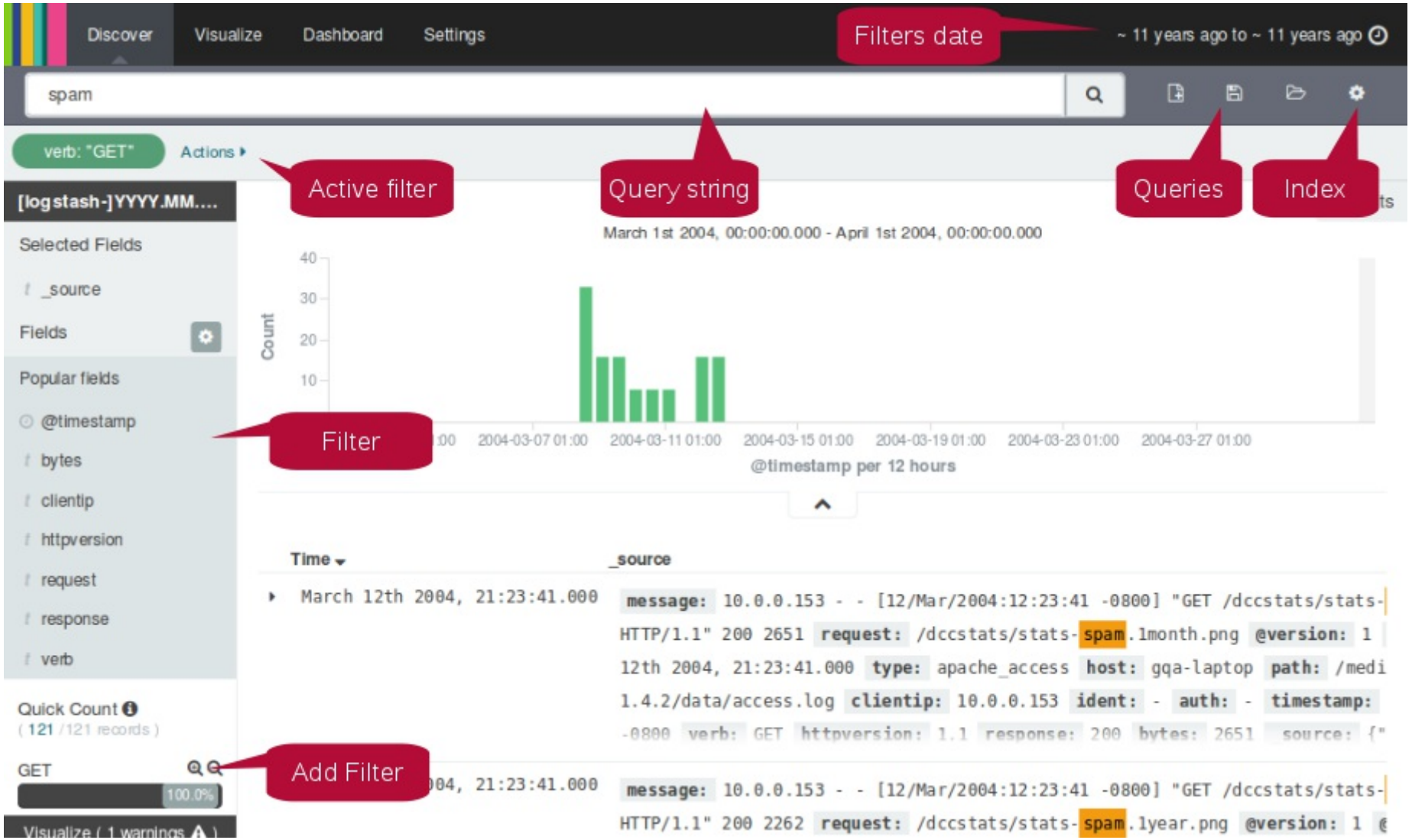
Logstash-Alternative: Ingest Node

- Implementation of some of the Logstash filters
- Pipeline defines processing order
- management via HTTP API
- Pipeline can be defined in Filebeat config

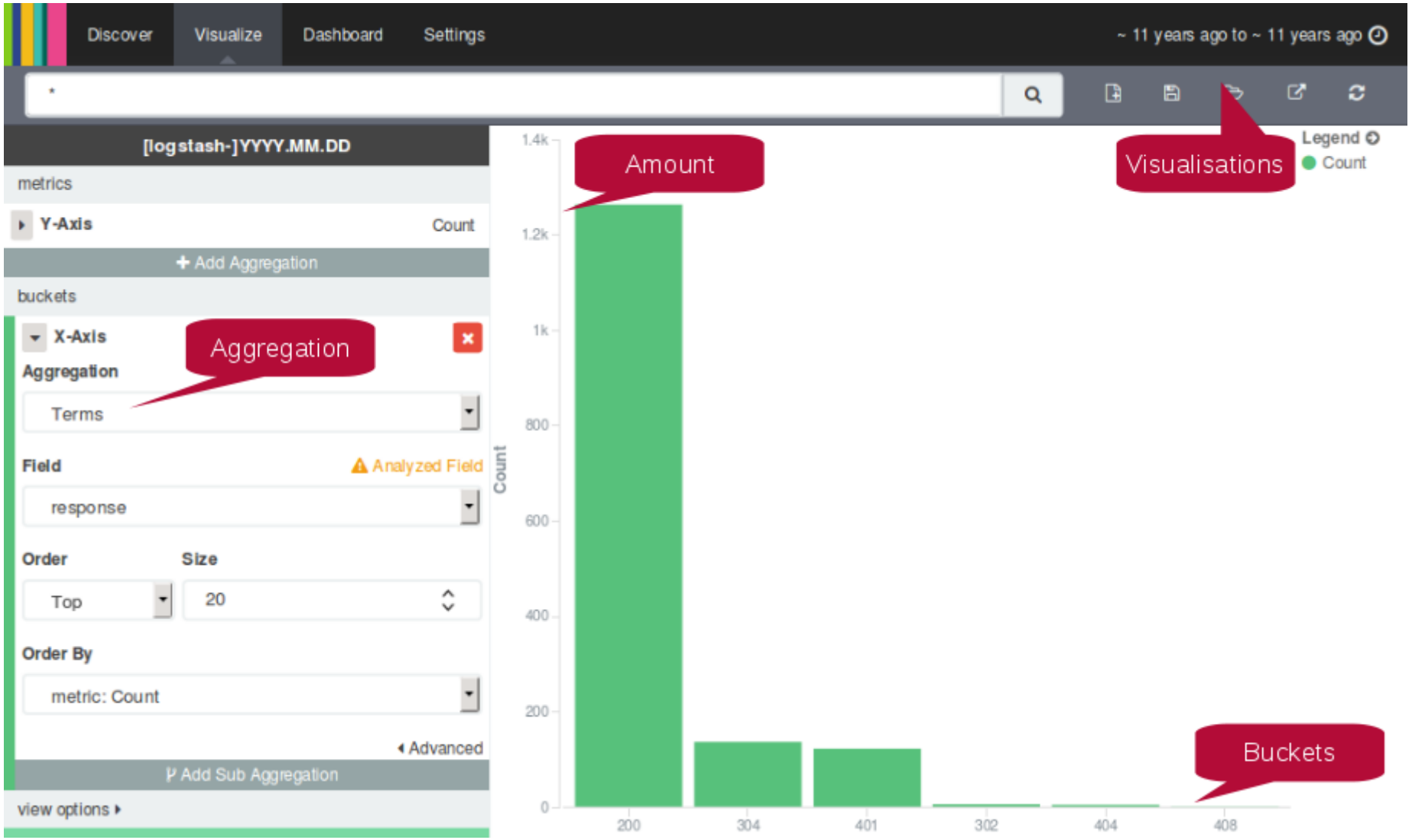
Kibana

- Search in log events
- Visualize data

Kibana



Kibana



Scaling options

- Backpressure sensitive when sending to Logstash and Elasticsearch
- Persistent Queues in Logstash
- Buffer for load spikes and for scaling Logstash
  - Redis
  - Kafka