elastic

Introduction to Beats and extending Beats

Medcl/曾勇
@medcl

小明：老师，今天我们要聊耳机么？

老师：出去!

Who am I?

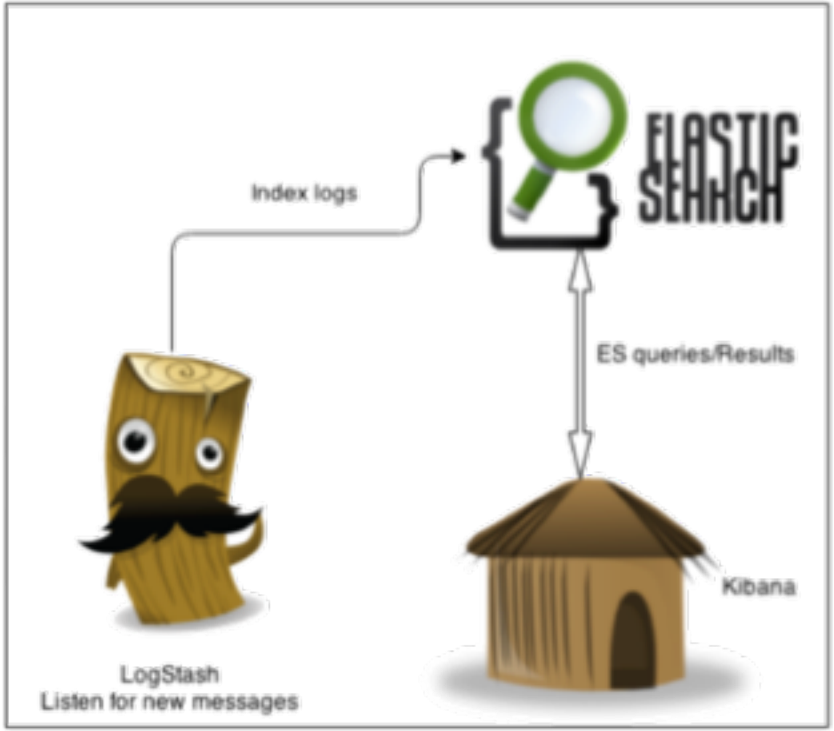
Medcl，曾勇（Zeng Yong）
Developer @ Elastic
- Follow Elasticsearch since v0.5, 2010
- Joined Elastic since September, 2015
- Now in Beats team

@medcl
medcl@elastic.co
http://github.com/medcl
Based in Changsha, Hunan, China

What’s Elastic?

- A distributed startup company，since 2012
  - –HQ: Mountain View, CA AND Amsterdam, Netherlands
  - –With employees in 27 countries (and counting), spread across 18 time zones, speaking over 30 languages
- We are working on Open Source projects!
  - –(Luckily some of them are popular, eg:elasticsearch)
- Offering Support Subscription，X-pack，Cloud and Trainings
- Find us on: https://github.com/elastic and https://www.elastic.co

Have you heard of ELK?

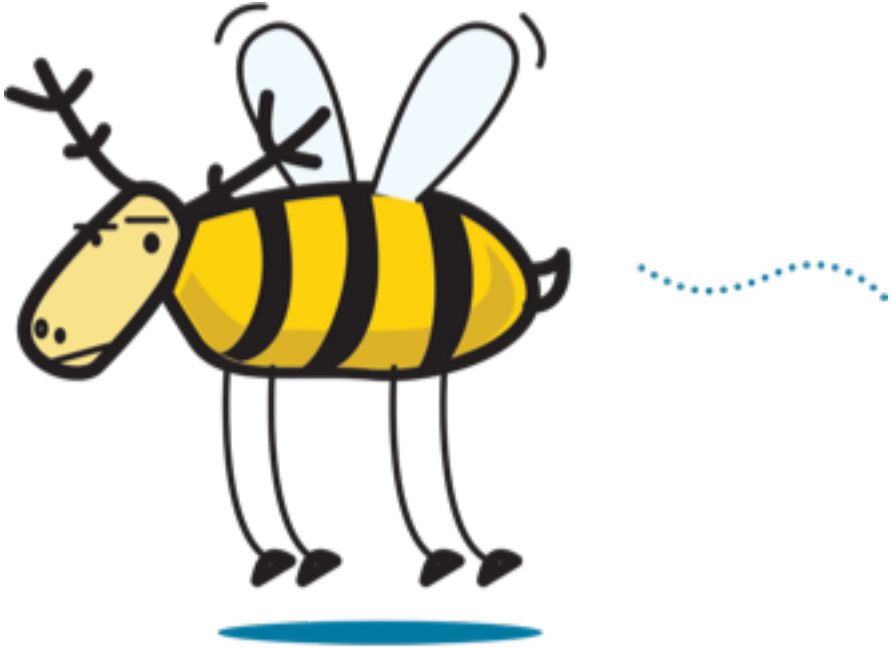


“ELK” is gone!
Beats! the terminator!

Well, just the name!

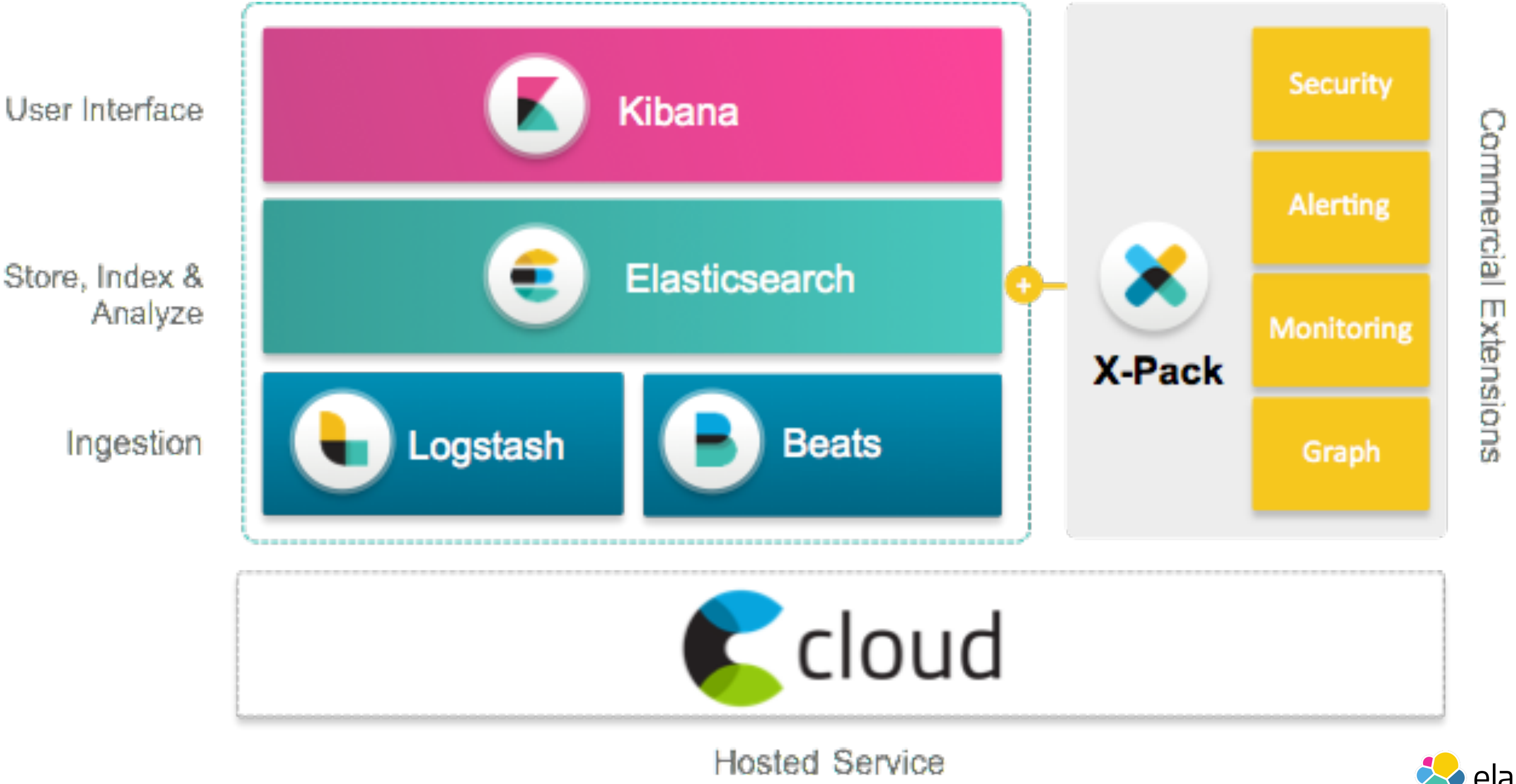
ELK?

Beats & Packetbeat

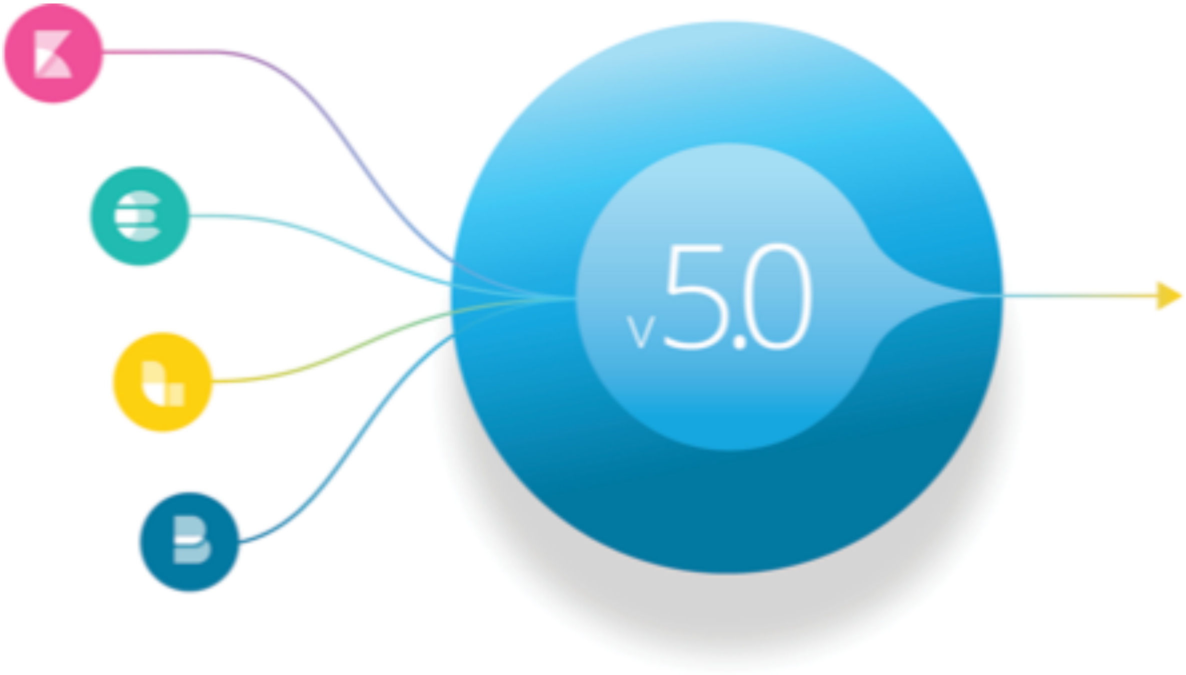


ELKB? BELK? LKBE? BKEL?

No “ELK” but “Elastic Stack”



Also, from V5.0, we release together



Beats are lightweight shippers that collect and ship all kinds of operational data to Elasticsearch

Beats are lightweight shippers that collect and ship all kinds of operational data to Elasticsearch

Lightweight shipper

- Small application
- Install as agent on your servers
- Written in Golang
- No runtime dependencies
- Single purpose

https://www.flickr.com/photos/8barbikes/17256970434/

http://github.com/elastic/beats

elastic

Beats are lightweight shippers that collect and ship all kinds of operational data to Elasticsearch

Examples of operational data

wire data

Packetbeat

system stats

Metricbeat

logs

Filebeat
Winlogbeat

14+

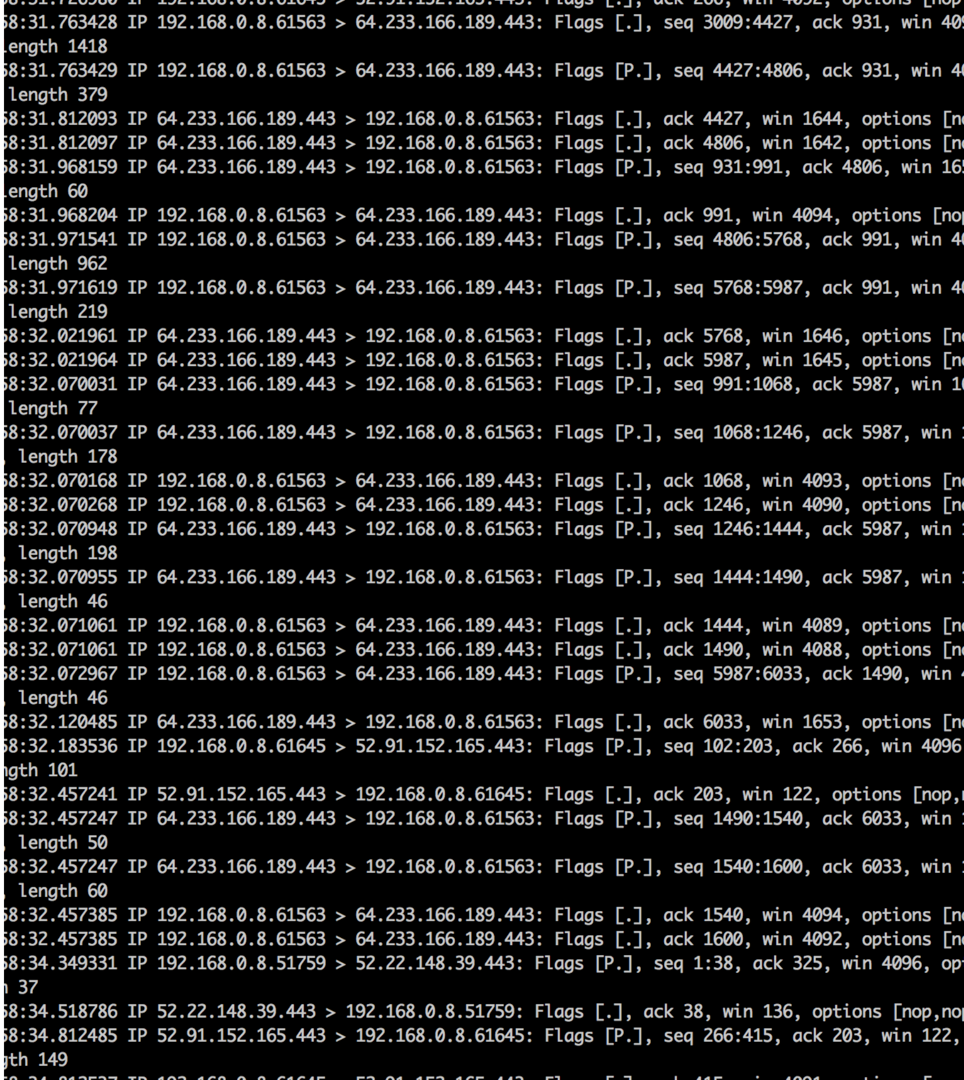
COMMUNITY BEATS

Sending all sorts of data to Logstash and Elasticsearch

1. Apachebeat
2. Dockerbeat
3. Elasticbeat
4. Execbeat
5. Factbeat
6. Hsbeat
7. Httpbeat
8. Nagioscheckbeat
9. Nginxbeat
10. Phpfpmbeat
11. Pingbeat
12. Redisbeat
13. Unifiedbeat
14. Uwsgibeat

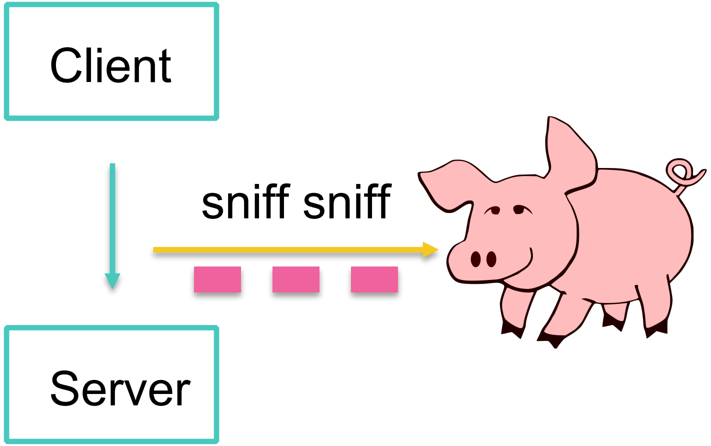
Packetbeat

Captures insights from network packets



```
58:31.763428 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], seq 3009:4427, ack 931, win 40
ength 1418
58:31.763429 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 4427:4806, ack 931, win 4
 length 379
58:31.812093 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [.], ack 4427, win 1644, options [n
58:31.812097 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [.], ack 4806, win 1642, options [n
58:31.968159 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 931:991, ack 4806, win 16
ength 60
58:31.968204 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 991, win 4094, options [nop
58:31.971541 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 4806:5768, ack 991, win 4
 length 962
58:31.971619 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 5768:5987, ack 991, win 4
 length 219
58:32.021961 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [.], ack 5768, win 1646, options [n
58:32.021964 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [.], ack 5987, win 1645, options [n
58:32.070031 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 991:1068, ack 5987, win 1
 length 77
58:32.070037 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1068:1246, ack 5987, win
, length 178
58:32.070168 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 1068, win 4093, options [n
58:32.070268 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 1246, win 4090, options [n
58:32.070948 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1246:1444, ack 5987, win
, length 198
58:32.070955 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1444:1490, ack 5987, win
, length 46
58:32.071061 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 1444, win 4089, options [n
58:32.071061 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 1490, win 4088, options [n
58:32.072967 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 5987:6033, ack 1490, win
, length 46
58:32.120485 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [.], ack 6033, win 1653, options [n
58:32.183536 IP 192.168.0.8.61645 > 52.91.152.165.443: Flags [P.], seq 102:203, ack 266, win 4096
ngth 101
58:32.457241 IP 52.91.152.165.443 > 192.168.0.8.61645: Flags [.], ack 203, win 122, options [nop,
58:32.457247 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1490:1540, ack 6033, win
, length 50
58:32.457247 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1540:1600, ack 6033, win
, length 60
58:32.457385 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 1540, win 4094, options [n
58:32.457385 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [.], ack 1600, win 4092, options [n
58:34.349331 IP 192.168.0.8.51759 > 52.22.148.39.443: Flags [P.], seq 1:38, ack 325, win 4096, op
h 37
58:34.518786 IP 52.22.148.39.443 > 192.168.0.8.51759: Flags [.], ack 38, win 136, options [nop,no
58:34.812485 IP 52.91.152.165.443 > 192.168.0.8.61645: Flags [P.], seq 266:415, ack 203, win 122,
gth 149
```

Sniffing the network traffic



- Copy traffic at OS or hardware level
- Is completely passive
- ZERO latency overhead
- Not in the request/response path, cannot break your application

Sniffing use cases

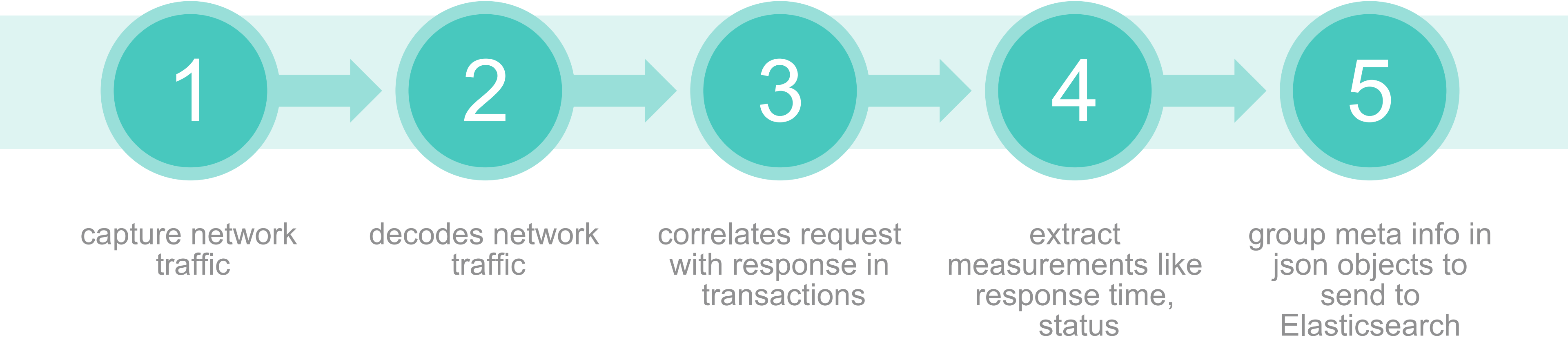
- Security
  - Intrusion Detection Systems
- Troubleshooting network issues
- Troubleshooting applications
- Performance analysis

WIRESHARK

SNORT®

TCPDUMP

Packetbeat: Real-time application monitoring



1 capture network traffic

2 decodes network traffic

3 correlates request with response in transactions

4 extract measurements like response time, status

5 group meta info in json objects to send to Elasticsearch

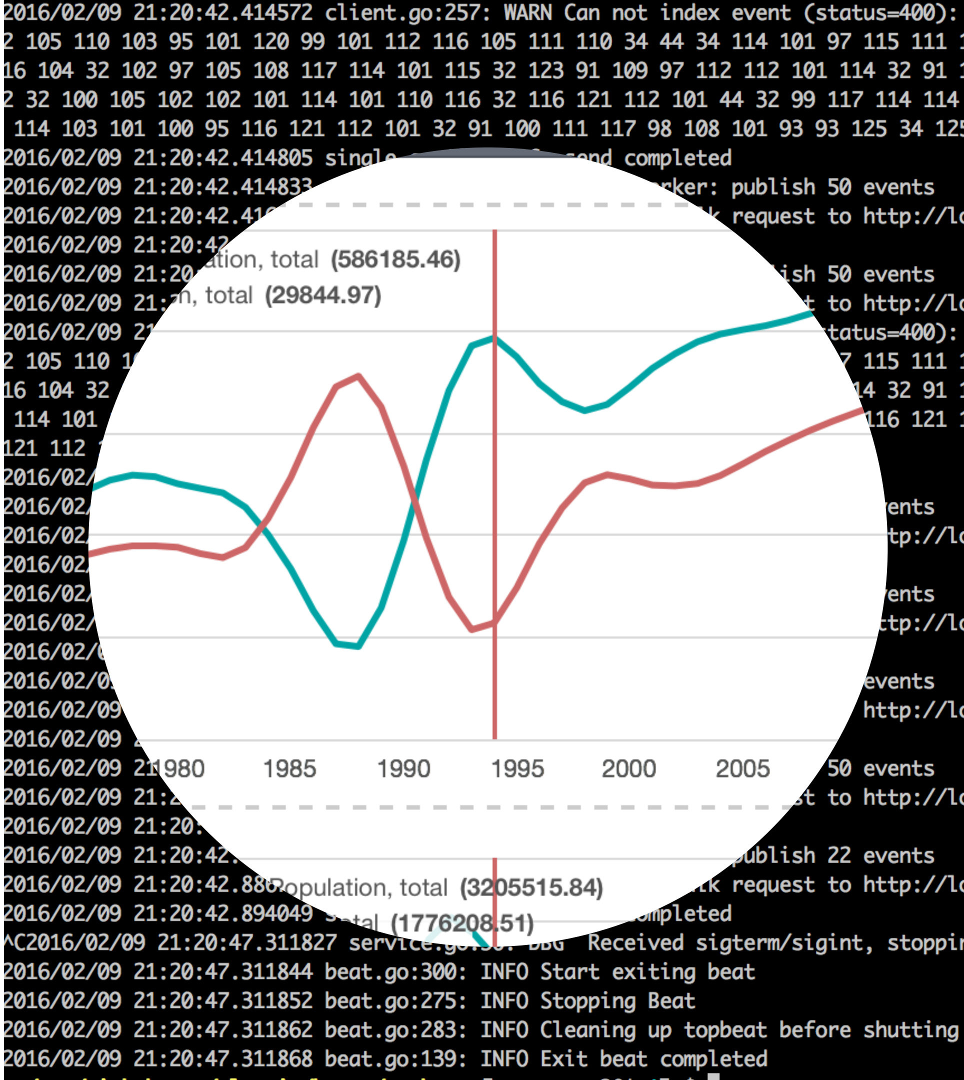
It does all of these in real-time directly on the target servers.

Packetbeat: Available decoders

- HTTP
- MySQL
- PostgreSQL
- Redis
- Thrift-RPC
- Memcache
- MongoDB
- ICMP
- DNS
- AMQP
- + Add your own

Metricbeat

Provides a common infrastructure for all “metrics” related Beats. Upcoming in 5.0.0-alpha1.



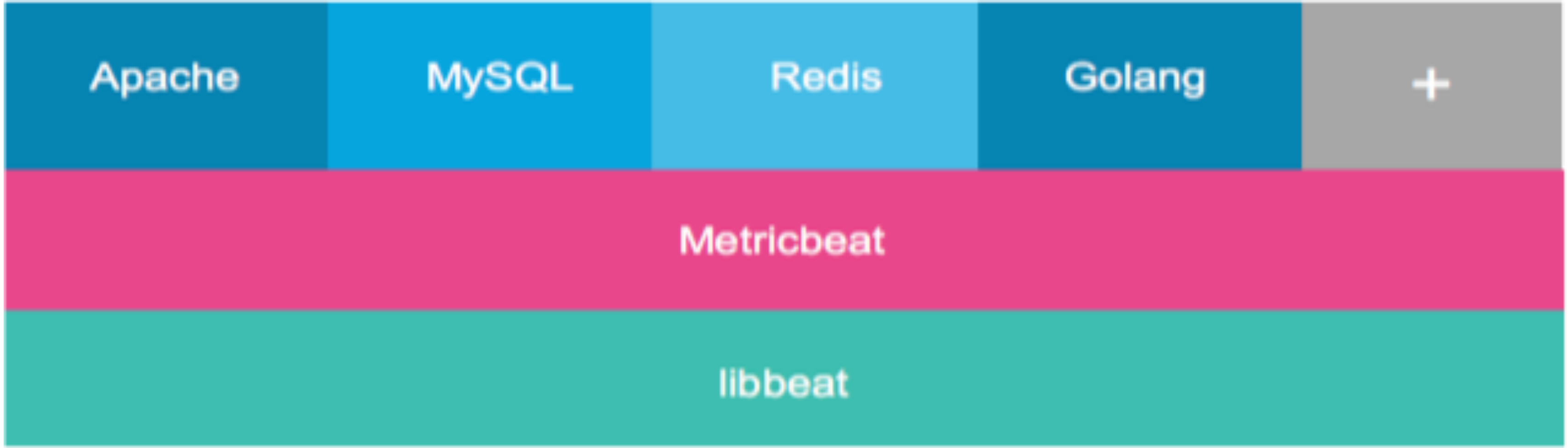
```
2016/02/09 21:20:42.414572 client.go:257: WARN Can not index event (status=400):
2 105 110 103 95 101 120 99 101 112 116 105 111 110 34 44 34 114 101 97 115 111 1
16 104 32 102 97 105 108 117 114 101 115 32 123 91 109 97 112 112 101 114 32 91 1
2 32 100 105 102 102 101 114 101 110 116 32 116 121 112 101 44 32 99 117 114 114
 114 103 101 100 95 116 121 112 101 32 91 100 111 117 98 108 101 93 93 125 34 125
2016/02/09 21:20:42.414805 single                end completed
2016/02/09 21:20:42.414833                     rker: publish 50 events
2016/02/09 21:20:42.41                         k request to http://lo
2016/02/09 21:20:42
2016/02/09 21:20                                  lish 50 events
2016/02/09 21:                                    t to http://lo
2016/02/09 2                                      tatus=400):
2 105 110 1                                       7 115 111 1
16 104 32                                         14 32 91 1
 114 101                                          116 121 1
121 112 1
2016/02/
2016/02/                                          ents
2016/02/                                          tp://lo
2016/02/
2016/02/                                          ents
2016/02/                                          ttp://lo
2016/02/0
2016/02/0                                         events
2016/02/09                                        http://lo
2016/02/09 2
2016/02/09 21                                     50 events
2016/02/09 21:2                                   t to http://lo
2016/02/09 21:20:
2016/02/09 21:20:42.                           publish 22 events
2016/02/09 21:20:42.886                        k request to http://lo
2016/02/09 21:20:42.894049                   ompleted
^C2016/02/09 21:20:47.311827 service.go:50: DBG  Received sigterm/sigint, stoppin
2016/02/09 21:20:47.311844 beat.go:300: INFO Start exiting beat
2016/02/09 21:20:47.311852 beat.go:275: INFO Stopping Beat
2016/02/09 21:20:47.311862 beat.go:283: INFO Cleaning up topbeat before shutting
2016/02/09 21:20:47.311868 beat.go:139: INFO Exit beat completed
```

Metricbeat: Collecting metrics from other systems

1. Periodically polls monitoring APIs of various services
2. Groups performance data into documents
3. Ships them to Logstash / Elasticsearch

Beats: Metricbeat

Listens to the internal “beat” of systems via APIs.



Apache MySQL Redis Golang +

Metricbeat

libbeat

Metricbeat module vs standalone Beat

Metricbeat module

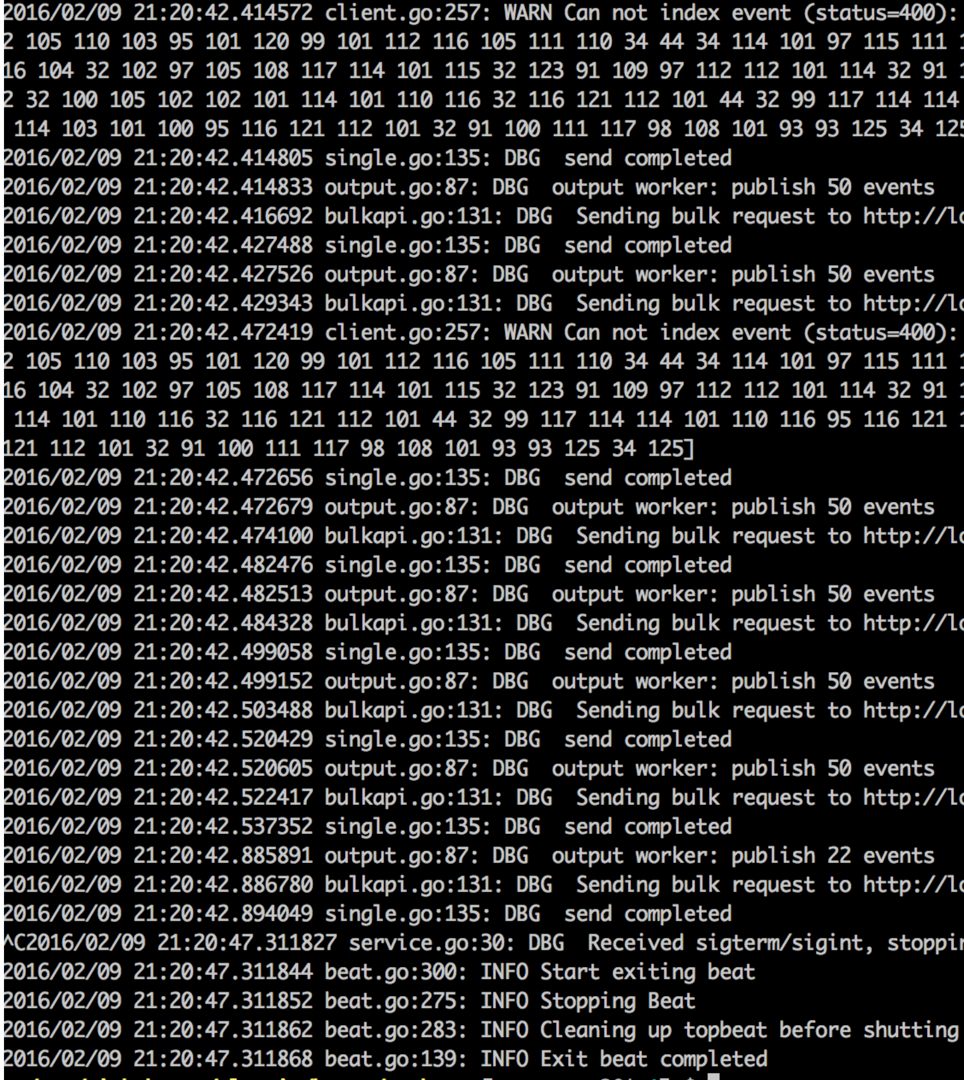
- Contributed via PR to the elastic/beats Github repository
- Officially supported
- Supports common systems
- Docker based integration tests

Standalone Beat

- In a separate Github repository
- Supported by the community
- Supports specialized systems
- Optional Docker based integration tests

Filebeat

Forwards log lines to Elasticsearch



```
2016/02/09 21:20:42.414572 client.go:257: WARN Can not index event (status=400):
2 105 110 103 95 101 120 99 101 112 116 105 111 110 34 44 34 114 101 97 115 111 1
16 104 32 102 97 105 108 117 114 101 115 32 123 91 109 97 112 112 101 114 32 91 1
2 32 100 105 102 102 101 114 101 110 116 32 116 121 112 101 44 32 99 117 114 114
 114 103 101 100 95 116 121 112 101 32 91 100 111 117 98 108 101 93 93 125 34 125
2016/02/09 21:20:42.414805 single.go:135: DBG  send completed
2016/02/09 21:20:42.414833 output.go:87: DBG  output worker: publish 50 events
2016/02/09 21:20:42.416692 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.427488 single.go:135: DBG  send completed
2016/02/09 21:20:42.427526 output.go:87: DBG  output worker: publish 50 events
2016/02/09 21:20:42.429343 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.472419 client.go:257: WARN Can not index event (status=400):
2 105 110 103 95 101 120 99 101 112 116 105 111 110 34 44 34 114 101 97 115 111 1
16 104 32 102 97 105 108 117 114 101 115 32 123 91 109 97 112 112 101 114 32 91 1
 114 101 110 116 32 116 121 112 101 44 32 99 117 114 114 101 110 116 95 116 121 1
121 112 101 32 91 100 111 117 98 108 101 93 93 125 34 125]
2016/02/09 21:20:42.472656 single.go:135: DBG  send completed
2016/02/09 21:20:42.472679 output.go:87: DBG  output worker: publish 50 events
2016/02/09 21:20:42.474100 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.482476 single.go:135: DBG  send completed
2016/02/09 21:20:42.482513 output.go:87: DBG  output worker: publish 50 events
2016/02/09 21:20:42.484328 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.499058 single.go:135: DBG  send completed
2016/02/09 21:20:42.499152 output.go:87: DBG  output worker: publish 50 events
2016/02/09 21:20:42.503488 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.520429 single.go:135: DBG  send completed
2016/02/09 21:20:42.520605 output.go:87: DBG  output worker: publish 50 events
2016/02/09 21:20:42.522417 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.537352 single.go:135: DBG  send completed
2016/02/09 21:20:42.885891 output.go:87: DBG  output worker: publish 22 events
2016/02/09 21:20:42.886780 bulkapi.go:131: DBG  Sending bulk request to http://lo
2016/02/09 21:20:42.894049 single.go:135: DBG  send completed
^C2016/02/09 21:20:47.311827 service.go:30: DBG  Received sigterm/sigint, stoppin
2016/02/09 21:20:47.311844 beat.go:300: INFO Start exiting beat
2016/02/09 21:20:47.311852 beat.go:275: INFO Stopping Beat
2016/02/09 21:20:47.311862 beat.go:283: INFO Cleaning up topbeat before shutting
2016/02/09 21:20:47.311868 beat.go:139: INFO Exit beat completed
```

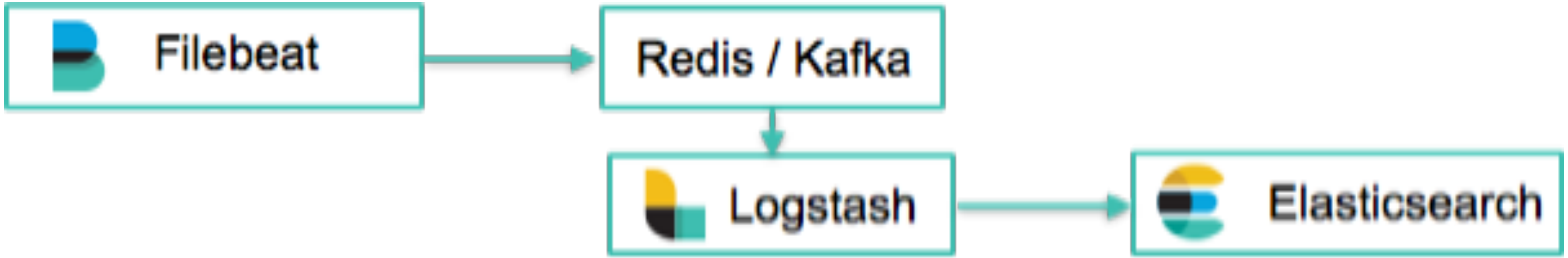
Beats: Filebeat

A more lightweight log shipper

- **Multiline**
- **Support Generic filtering**

  Flexibly reduce the amount of data sent of the wire and stored
- **Support Kafka/Redis**
- **Decode JSON from log lines**
- **Integration with IngestNode**

  Set “pipelineparameter” in the Elasticsearch output config

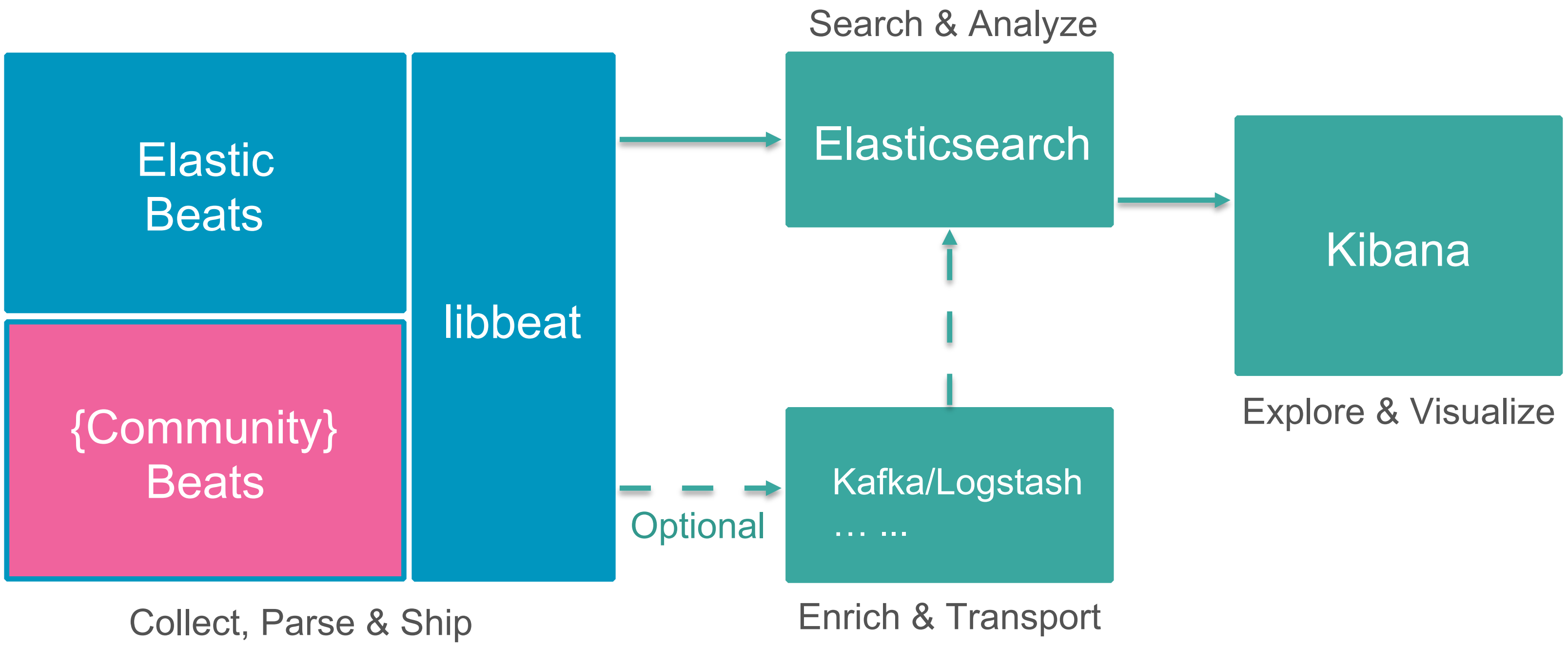


DEMO?

Extending Beats

General Beat
Packetbeat protocol
MetricBeat metricset

How beats works?



Search & Analyze

Elastic Beats

{Community} Beats

libbeat

Elasticsearch

Kibana

Optional

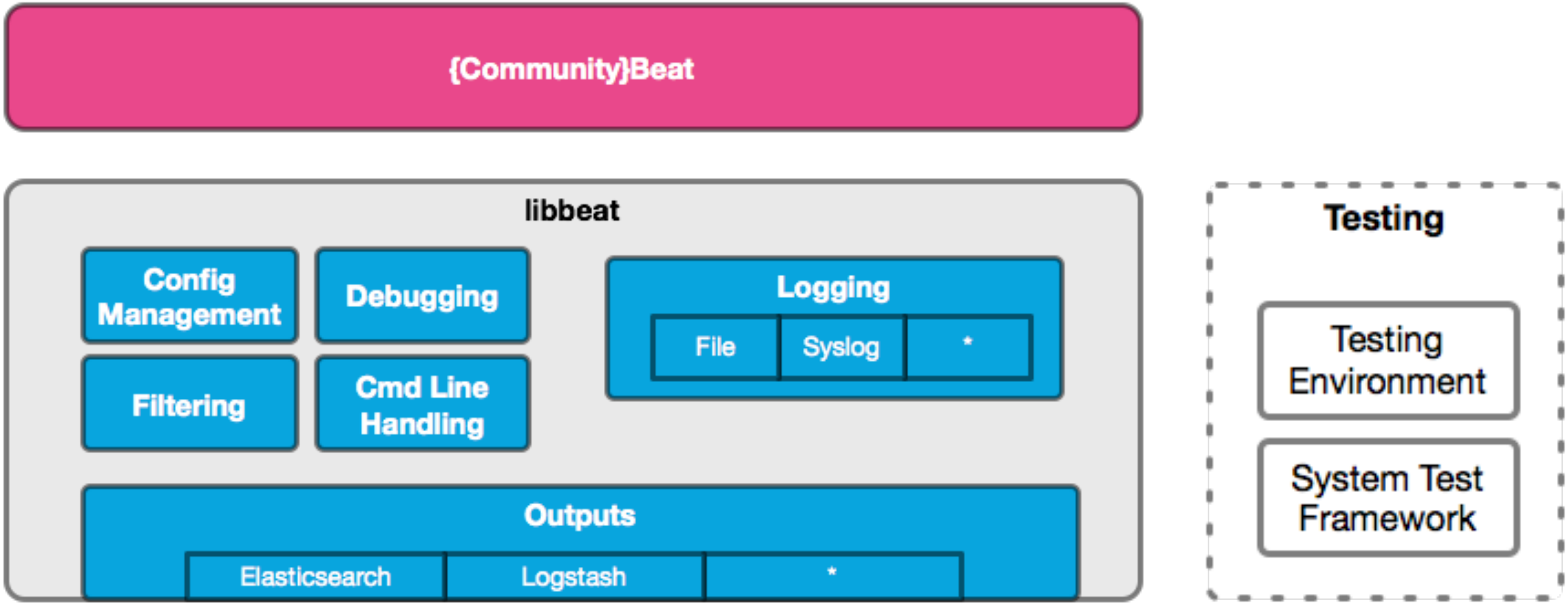
Kafka/Logstash … ...

Explore & Visualize

Collect, Parse & Ship

Enrich & Transport

Architecture Overview - libbeat



Beat generator

Quickly get started with the development of a new Beat

```
$ pip install cookiecutter

  ...
project_name [Examplebeat]: Mybeat
github_name [your-github-name]: tsg
beat [examplebeat]: mybeat
beat_path [github.com/your-github-name]: github.com/tsg
full_name [Firstname Lastname]: Tudor Golubenco
```

http://github.com/elastic/beats/generate

Extending Metricbeat

- Create you own metricbeat
  - Step 0
    - pip install cookiecutter
  - Step 1
    - git clone https://github.com/elastic/beats $GOPATH/src/github.com/elastic/beats
  - Step 2
    - cookiecutter $GOPATH/src/github.com/elastic/beats/generate/metricbeat/metricset
  - Step 3
    - make setup
- OR extending Metricbeat: cd **beats/metricbeat** && make create-metricset

How Metricbeat/General beats extending works?

- Metricbeat
  - Module(Nginx/System/…)
    - Fetch
      - Parse
        - Store
- General beats
  - Community Beat (Nginxbeat/Dockerbeat/ ….)
    - Fetch
      - Parse
        - Store

PULL the data

elastic

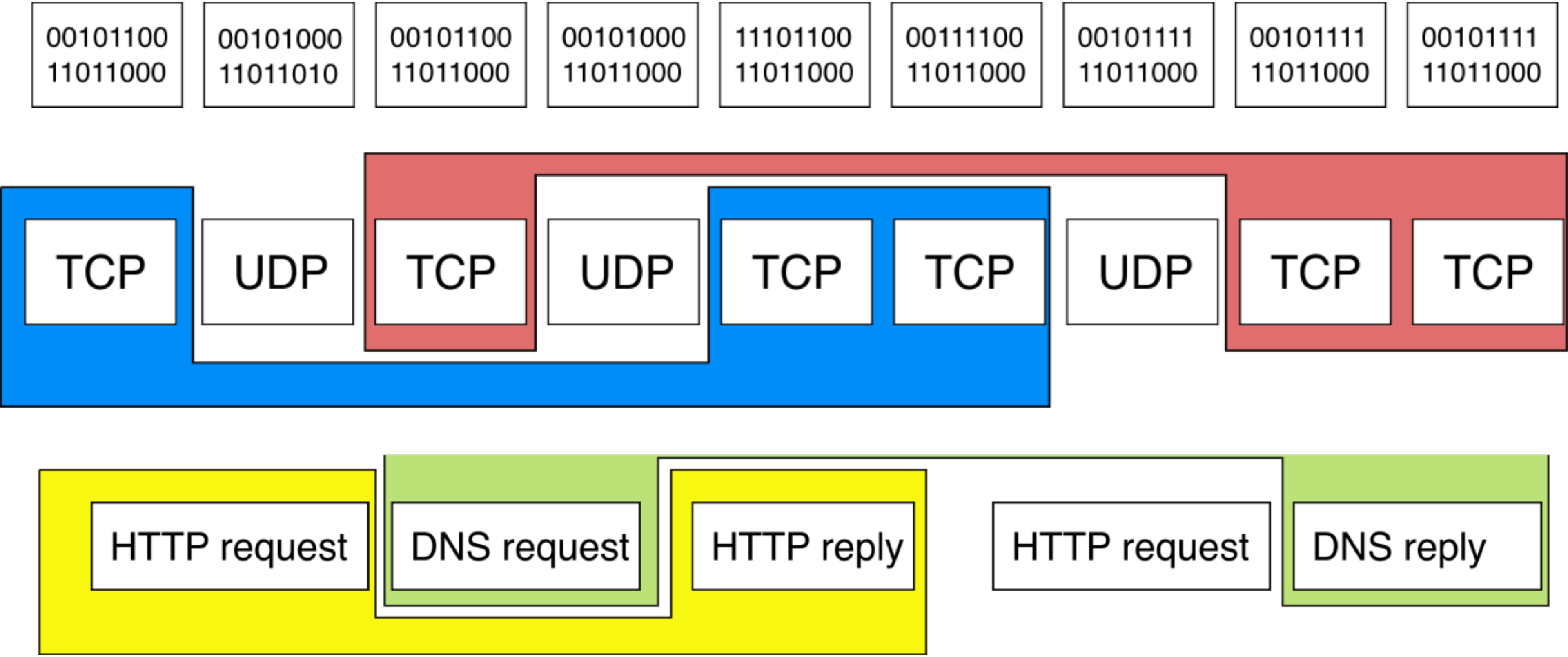
Packetbeat?

Listen to the data

说好不聊耳机，还来?

How to extend Packetbeat?

Decode, TCP is hard!



00101100 11011000

00101000 11011010

00101100 11011000

00101000 11011000

11101100 11011000

00111100 11011000

00101111 11011000

00101111 11011000

00101111 11011000

TCP

UDP

TCP

UDP

TCP

TCP

UDP

TCP

TCP

HTTP request

DNS request

HTTP reply

HTTP request

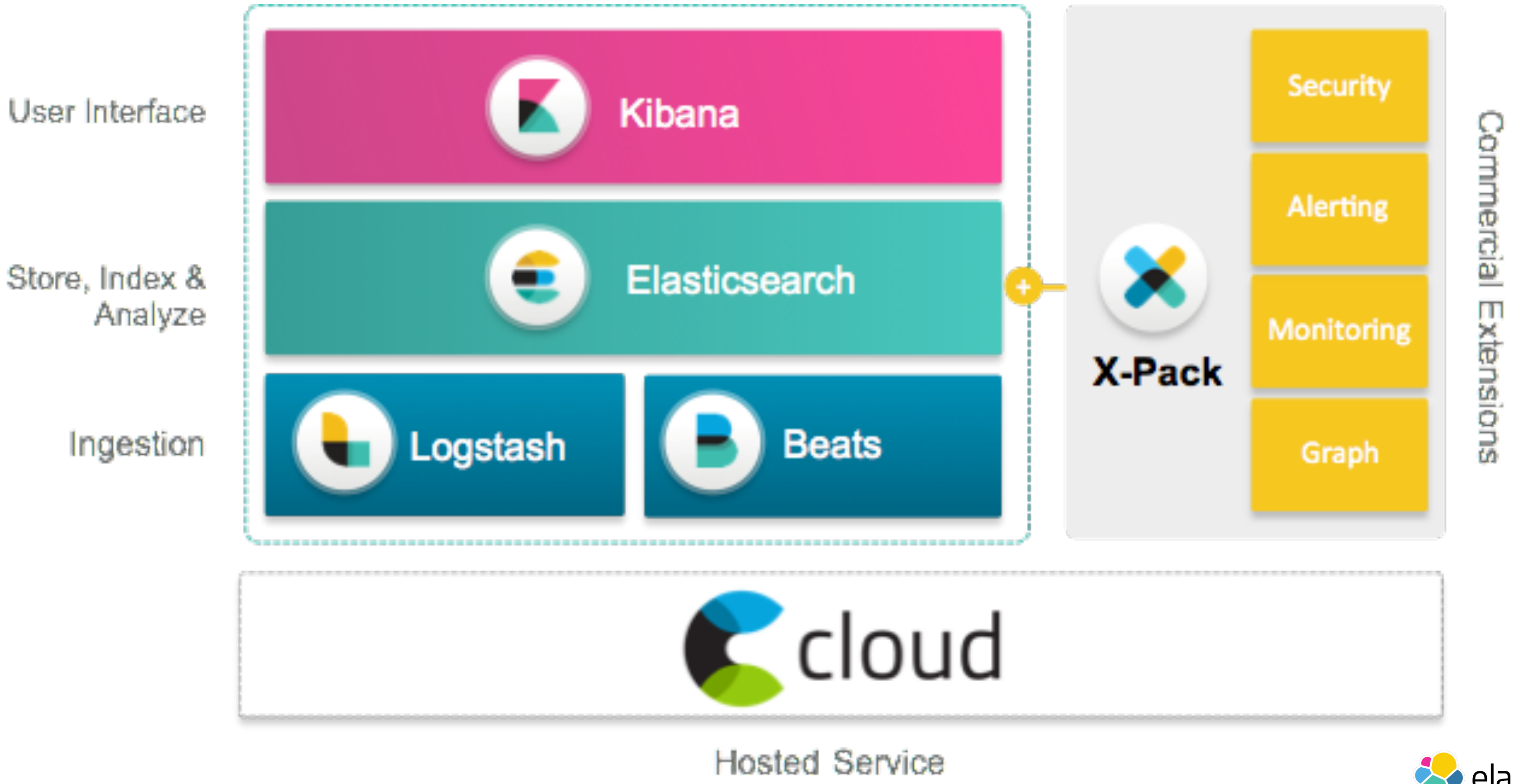
DNS reply

Let’s DIY a Cassandra protocol

Common issue

- **packetbeat.interfaces.device:** any
  - CRIT Exiting: Initializing sniffer failed: Error creating decoder: Unsupported link type: UnknownLinkType(12)

The “Elastic Stack”



User Interface

Kibana

Store, Index & Analyze

Elasticsearch

Ingestion

Logstash

Beats

X-Pack

Security

Alerting

Monitoring

Graph

Commercial Extensions

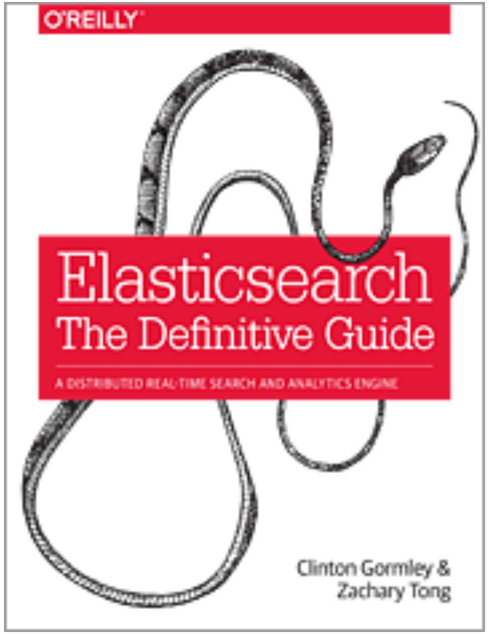
cloud

Hosted Service

Community

- 源码 & Issue: http://github.com/elastic/
- 中文社区: http://elasticsearch.cn
- 官方 QQ 群: 190605846

ES权威指南翻译中，欢迎志愿者加入！
https://github.com/elasticsearch-cn/elasticsearch-definitive-guide


O'REILLY
Elasticsearch
The Definitive Guide
A DISTRIBUTED REAL-TIME SEARCH AND ANALYTICS ENGINE
Clinton Gormley & Zachary Tong

Thanks