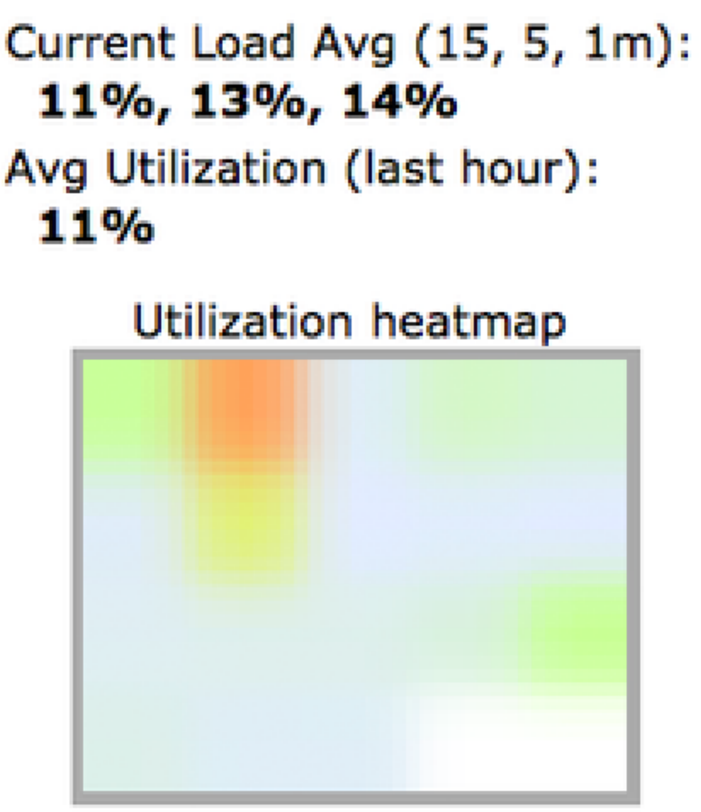
Get Started With Beats

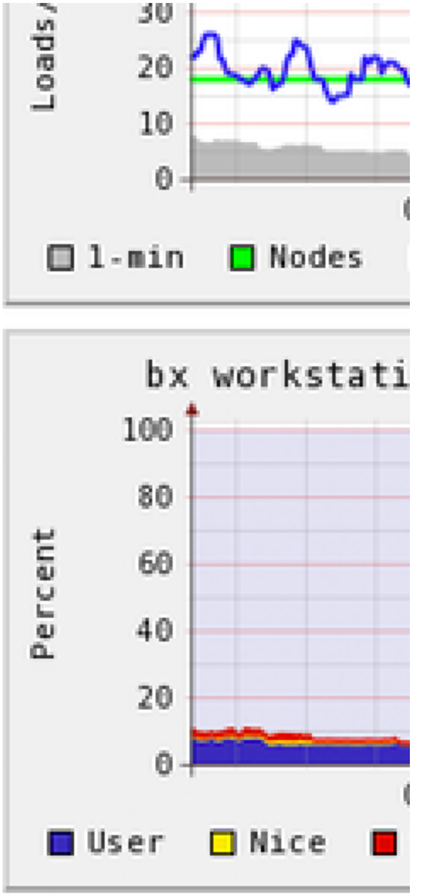
Medcl

elastic

In today's program

- 无数监控图表,需要借助搜索
- 每项指标都有N个维度
- 使用平均值来进行监控
- 系统架构/组件越来复杂
- 分布式环境异常复杂





About me

@medcl

http://github.com/medcl

I work for

elastic

the company behind

Elasticsearch

Logstash

Kibana

also known as

Photo credit: https://www.flickr.com/photos/lsmith2010/8215026548

Open source culture

- GitHub
  - Pull Requests
  - Issues
- 会议
  - Elastic{ON}
- 社区
  - discuss.elastic.co
  - elasticsearch.cn
- 博客

Image credit: https://www.flickr.com/photos/tappnel/5798812875

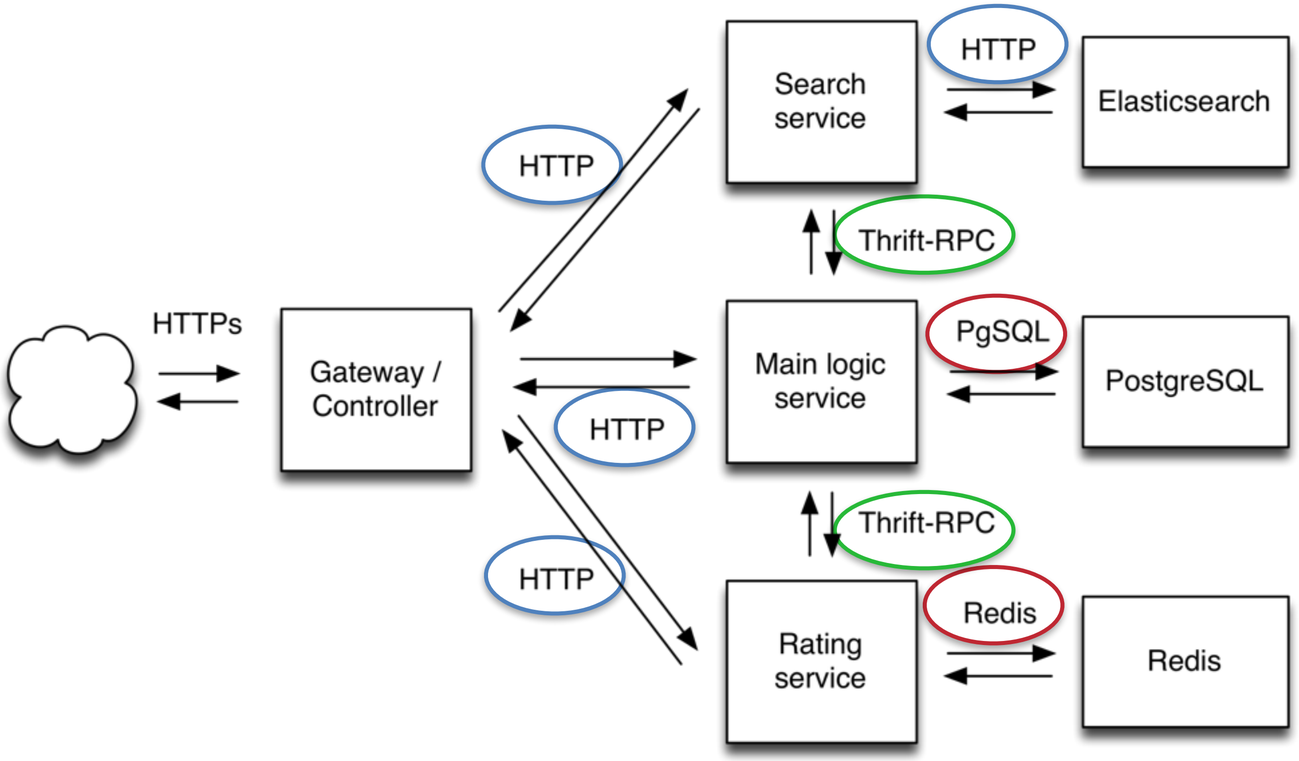
Open source project

Started by **Monica Sarbu** 5.2014
Tudor Golubenco joined full time 11.2014
Open Source, Written in Golang
Now part of Elastic family

Beats 能干嘛?

对复杂应用及基础设施进行监控和诊断

Idea



系统内部通信的一个例子

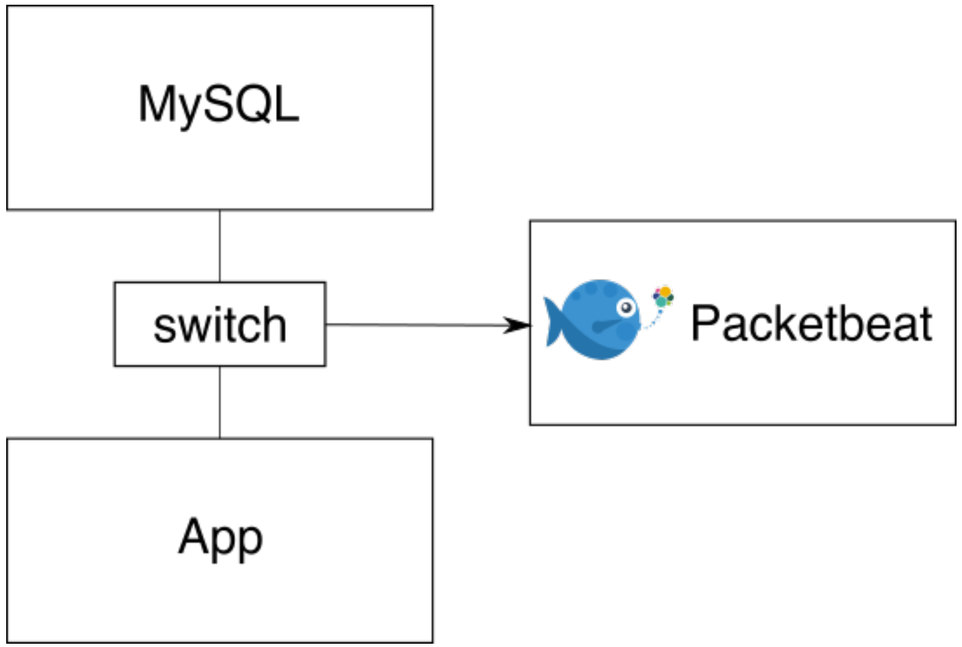
Capture network packets

- 监听基础设施不麻烦
- 被动监听网络流量请求

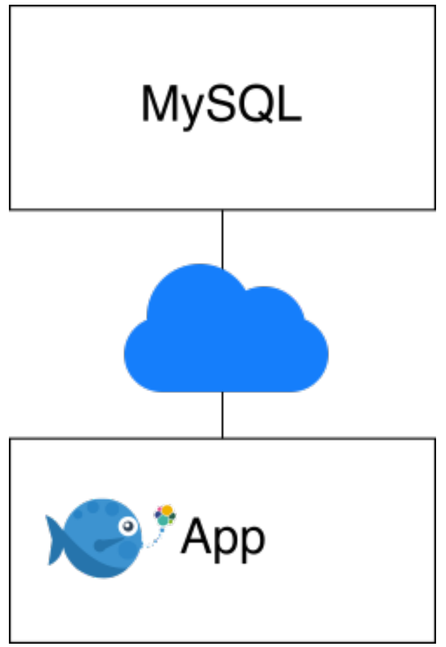
- 不会添加网络延时
- 不会弄坏你的程序

Image credit: https://www.flickr.com/photos/bigdrumthump/3223280727

Packet capturing



1. Using port mirroring



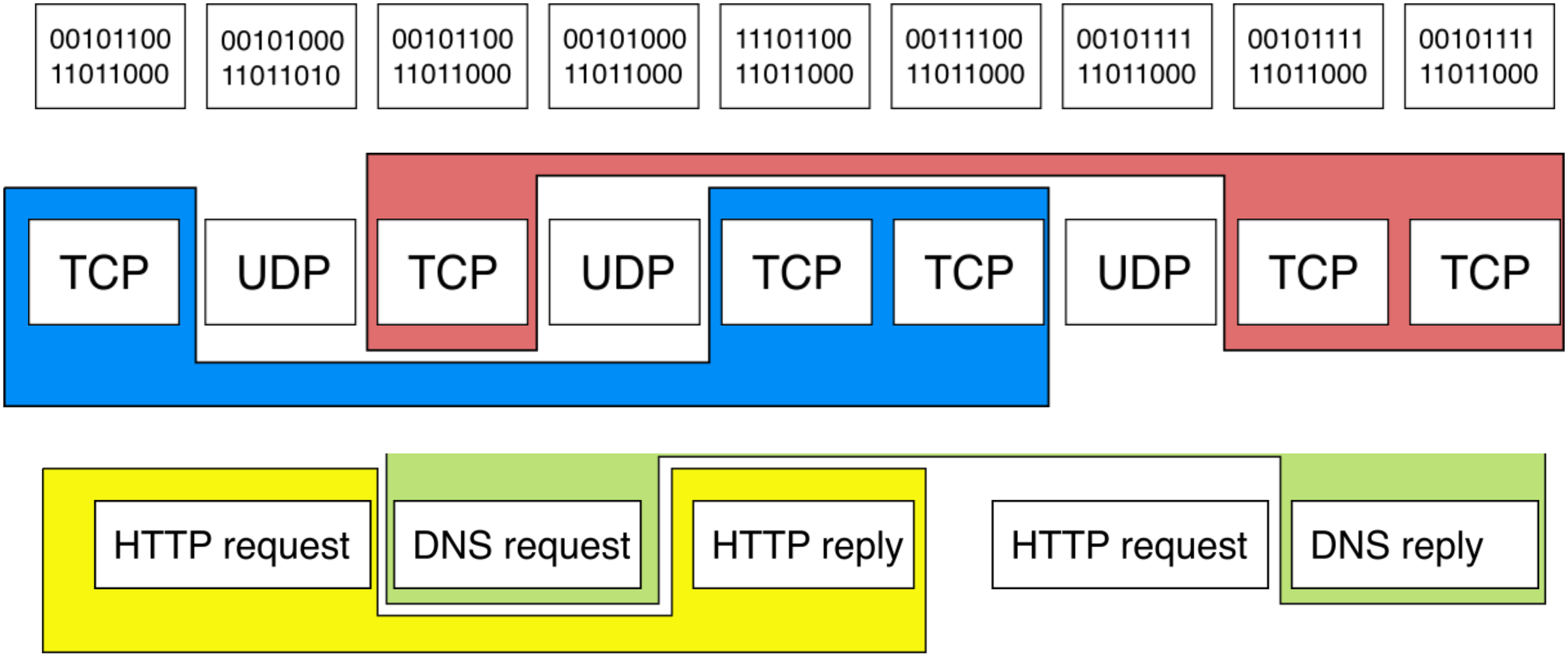
2. As an “agent”

Sniffing from a technical PoV

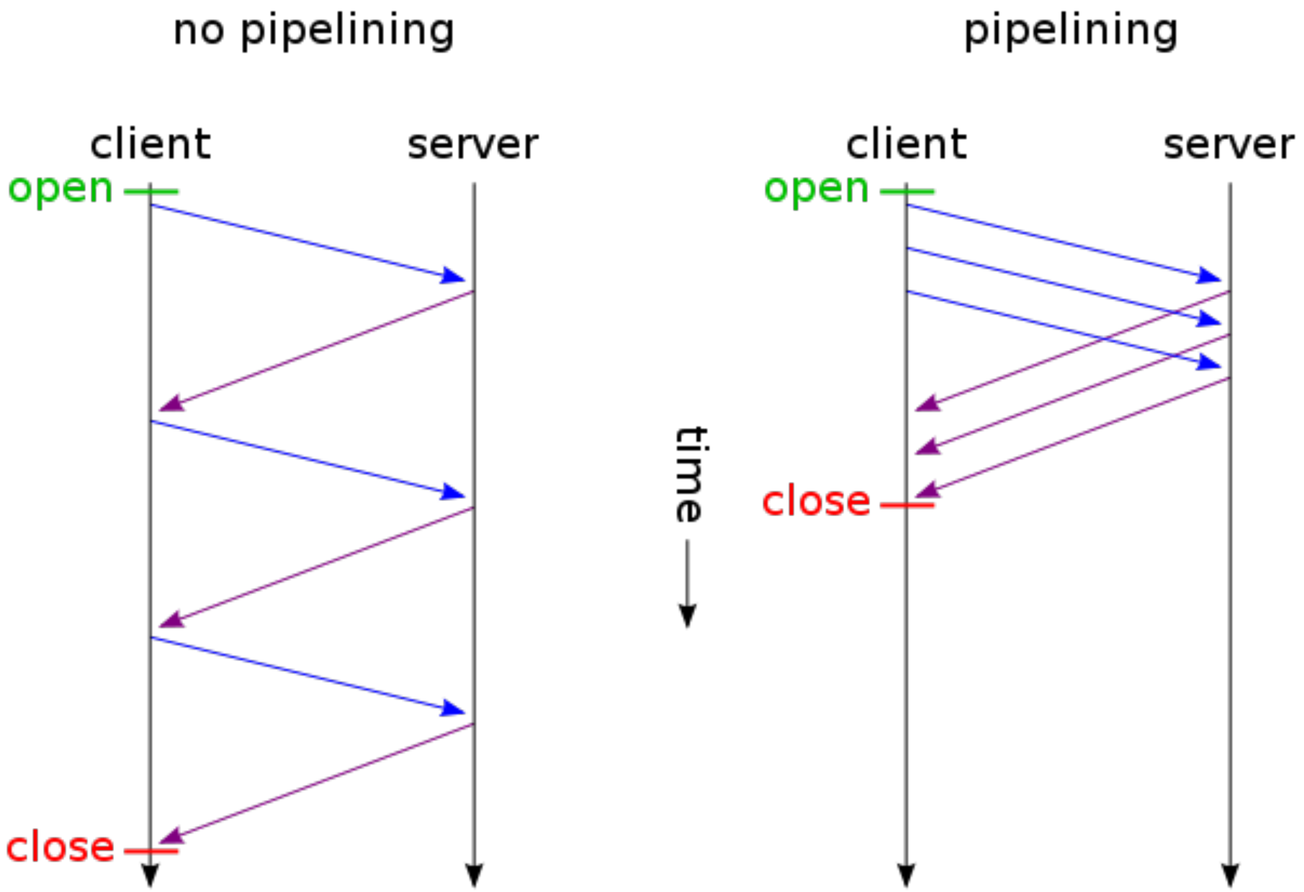
- **libpcap** (tcpdump), supports all Unix like systems
- **Winpcap**, supports Windows
- For Go, **gopacket** provides bindings and more
- High speed API for packet capturing on Linux: **af_packet**

Image credit: https://www.flickr.com/photos/57881779@N04/7930362242/

Decoding


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

Matching requests and responses



- Pipelining complicates matching the requests with the responses.

Create a JSON object for each request-response pair

Response code

GET method

Response time

HTTP transaction

SQL example


```
{
  "bytes_in": 63,
  "bytes_out": 147,
  "client_ip": "127.0.0.1",
  "client_port": 37881,
  "count": 1,
  "ip": "127.0.0.1",
  "method": "INSERT",
  "pgsql": {
    "error_code": "23505",
    "error_message": "duplicate key value violates unique constraint \"unique_a\"",
    "error_severity": "ERROR",
    "iserror": true,
    "num_fields": 0,
    "num_rows": 0
  },
  "port": 5432,
  "query": "insert into test(a, b, c) values('mea5', 'meb5', 'mec5')",
  "responsetime": 0,
  "shipper": "Tudors-MBP.localdomain",
  "status": "Error",
  "timestamp": "2015-01-10T20:11:10.469Z",
  "type": "pgsql"
}
```


Bandwidth information

SQL method

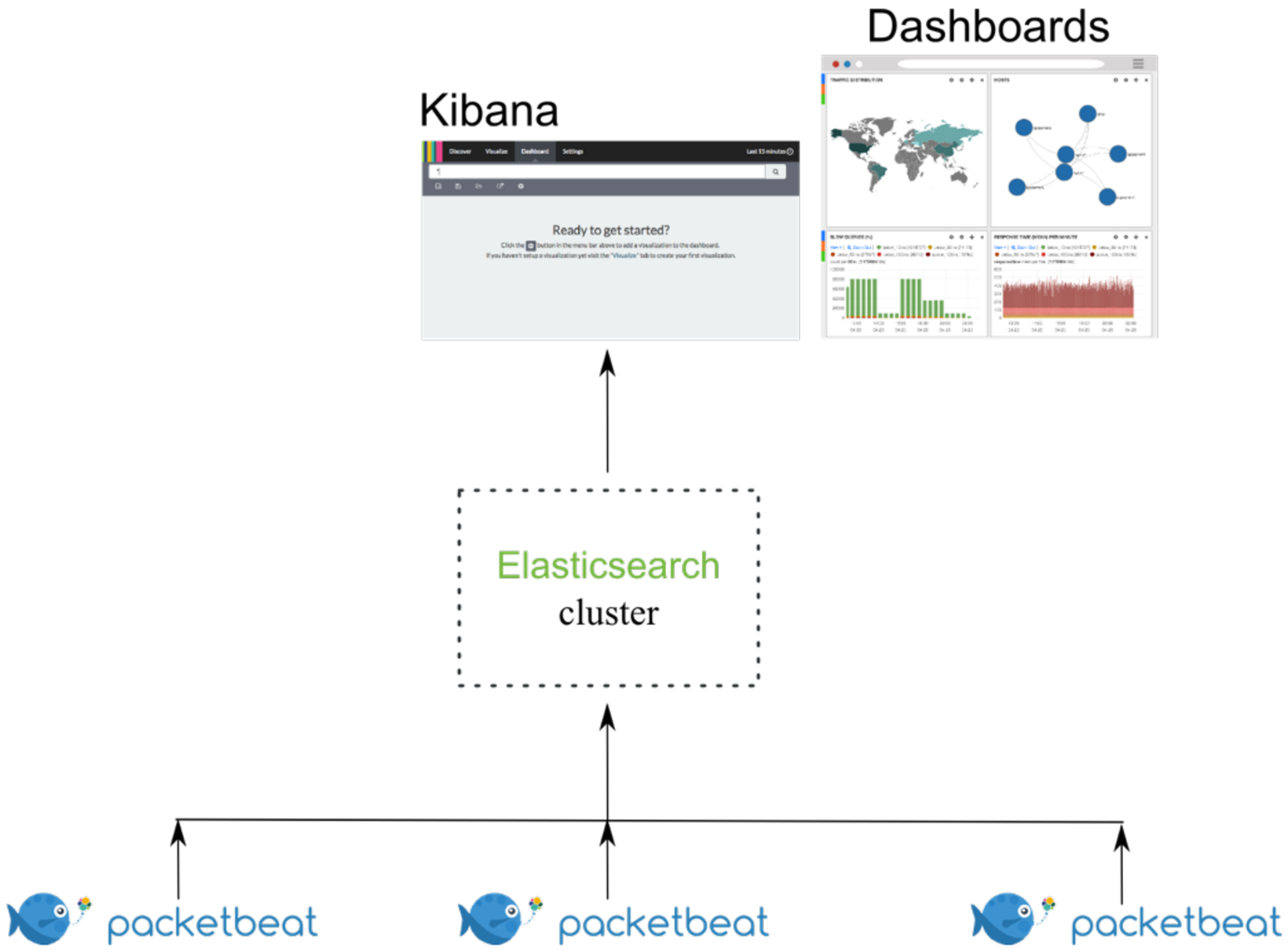
Error message

Query

DNS example


Query type
Domain name
Response time
{
"bytes_in": 28,
"bytes_out": 284,
"client_ip": "192.168.238.68",
"client_port": 60893,
"count": 1,
"dns": {
"additionals_count": 0,
"answers_count": 16,
"authorities_count": 0,
"flags": {
"authoritative": false,
"recursion_allowed": true,
"recursion_desired": true,
"truncated_response": false
},
"id": 18439,
"op_code": "QUERY",
"question": {
"class": "IN",
"name": "google.com",
"type": "A"
},
"response_code": "NOERROR"
},
"ip": "192.168.238.1",
"method": "QUERY",
"port": 53,
"query": "class IN, type A, google.com",
"resource": "google.com",
"responsetime": 61,
"shipper": "Tudors-MBP.localdomain",
"status": "OK",
"timestamp": "2015-08-27T08:00:55.638Z",
"transport": "udp",
"type": "dns"
}

Packetbeat: Overview



Packetbeat UP AND RUN

1.Download

```
curl -L -O https://download.elastic.co/beats/packetbeat/packetbeat-1.0.0-beta4-darwin.tgz
tar xzvf packetbeat-1.0.0-beta4-darwin.tgz
```

2.Run

```
sudo ./packetbeat -e -c packetbeat.yml
```

3.Test

```
./packetbeat
-c etc/packetbeat.yml
-N -d "publish"
-I /Users/medcl/Desktop/Beats/test_capture_data/smtp-25.pcap
-e
-d smtp
```

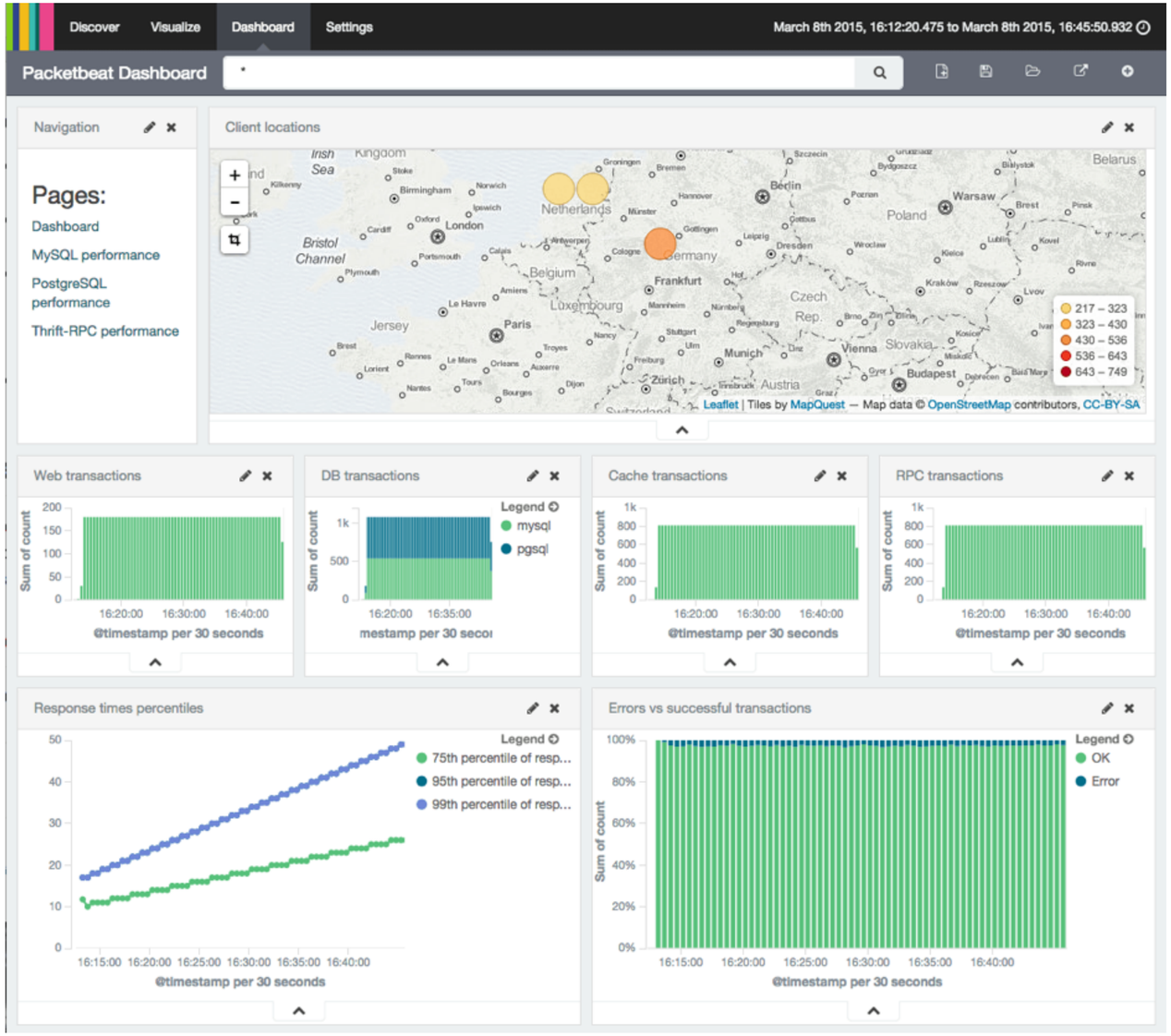
Command Help

```
Usage of ./packetbeat:
  -I="": file
  -N=false: Disable actual publishing for testing
  -O=false: Read packets one at a time (press Enter)
  -c="/etc/packetbeat/packetbeat.yml": Configuration file
  -cpuprofile="": Write cpu profile to file
  -d="": Enable certain debug selectors
  -devices=false: Print the list of devices and exit
  -dump="": Write all captured packets to this libpcap file
  -e=false: Output to stdout and disable syslog/file output
  -l=1: Loop file. 0 - loop forever
  -memprofile="": Write memory profile to this file
  -t=false: Read packets as fast as possible, without sleeping
  -test=false: Test configuration and exit.
  -v=false: Log at INFO level
  -version=false: Print version and exit
  -waitstop=0: Additional seconds to wait befor shutting down
```

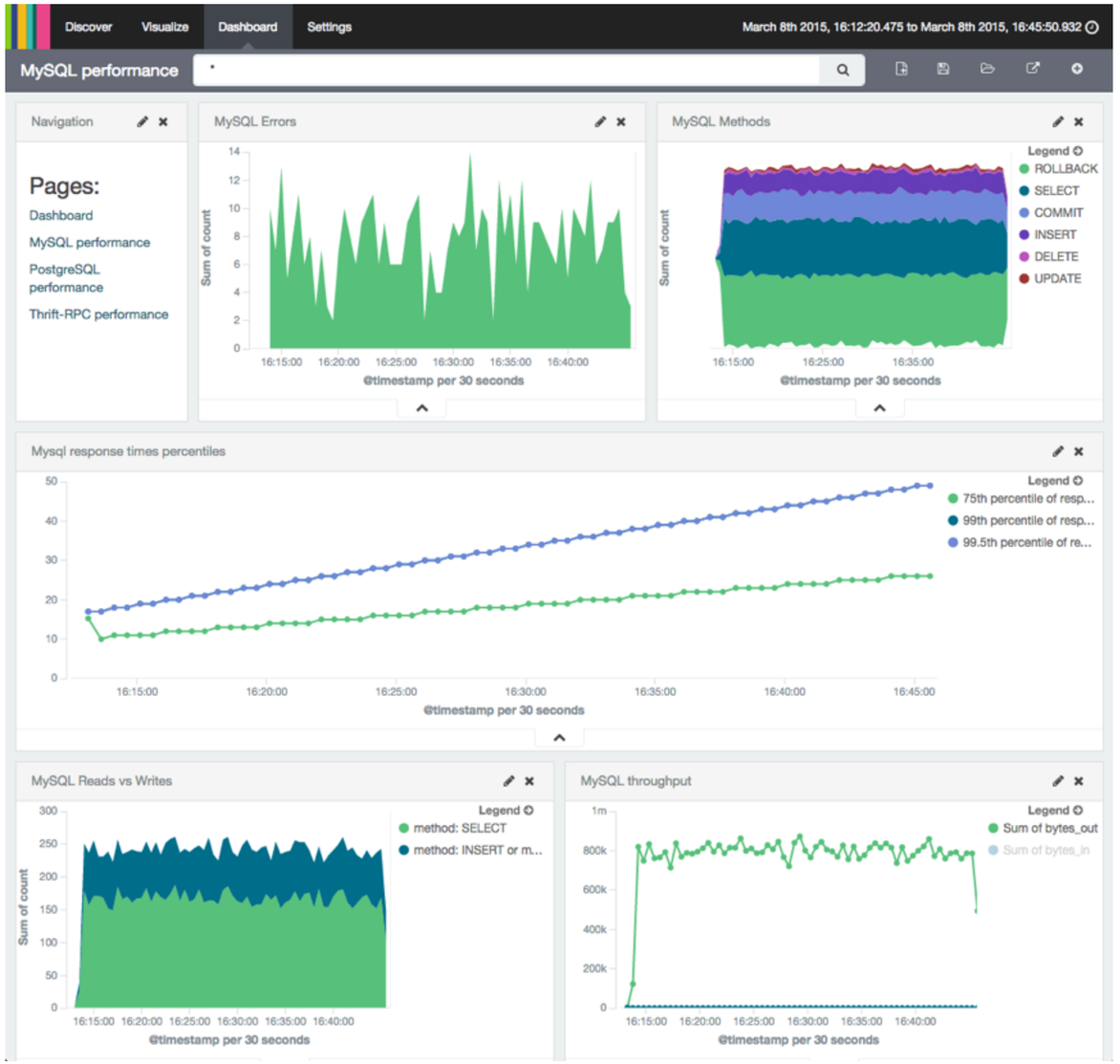
packetbeat.yml

```
interfaces:
  device: any

protocols:
  http:
    ports: [80, 8080, 8081, 5000, 8002]

  memcache:
    ports: [11211]

  mysql:
    ports: [3306]

  redis:
```

Visualize data with Kibana



Visualize data with Kibana



There’s more to apps than packets

Packetbeat

Listens to the “beat” of the network packets.

Topbeat

Listens to the “beat” of the operating system metrics.

Filebeat

Listens to the “beat” of logs.

Metricsbeat

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

Image credits:
https://www.flickr.com/photos/7147684@N03/921738874/
https://www.flickr.com/photos/bigdrumthump/3223280727
https://www.flickr.com/photos/jadeashleyphotography/6584949945/
https://www.flickr.com/photos/mitosettembremusica/2839965900/

Topbeat

```
top - 12:05:20 up 2 days,  6:39,  1 user,  load average: 0.06, 0.06, 0.05
Tasks:  77 total,   1 running,  76 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.0 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   2060912 total,  1570144 used,   490768 free,   244840 buffers
KiB Swap:   892924 total,        0 used,   892924 free,   653312 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 5130 elastics  20   0 1773m 415m  14m S   1.7 20.7 116:19.57 java
    1 root      20   0 10652  804  672 S   0.0  0.0   0:07.06 init
    2 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S   0.0  0.0   0:06.99 ksoftirqd/0
    5 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kworker/u:0
    6 root      rt   0     0    0    0 S   0.0  0.0   0:00.24 migration/0
    7 root      rt   0     0    0    0 S   0.0  0.0   0:06.10 watchdog/0
    8 root      rt   0     0    0    0 S   0.0  0.0   0:00.22 migration/1
   10 root      20   0     0    0    0 S   0.0  0.0   0:04.31 ksoftirqd/1
   12 root      rt   0     0    0    0 S   0.0  0.0   0:05.82 watchdog/1
   13 root      rt   0     0    0    0 S   0.0  0.0   0:00.25 migration/2
   15 root      20   0     0    0    0 S   0.0  0.0   0:04.56 ksoftirqd/2
   16 root      rt   0     0    0    0 S   0.0  0.0   0:05.57 watchdog/2
   17 root      rt   0     0    0    0 S   0.0  0.0   0:00.24 migration/3
   18 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kworker/3:0
   19 root      20   0     0    0    0 S   0.0  0.0   0:03.02 ksoftirqd/3
   20 root      rt   0     0    0    0 S   0.0  0.0   0:04.06 watchdog/3
```

- Like the Unix **top** command but sending the data periodically to Elasticsearch
- Works also on Windows

Topbeat system wide and per process stats


```
"count": 1,
"cpu": {
  "user": 42565,
  "user_p": 0,
  "nice": 0,
  "system": 22353,
  "system_p": 0.99,
  "idle": 12179145,
  "iowait": 1738,
  "irq": 1849,
  "softirq": 597,
  "steal": 0
},
"load": {
  "load1": 0,
  "load5": 0,
  "load15": 0
},
"mem": {
  "total": 250780,
  "used": 213688,
  "free": 37092,
  "used_p": 85.21,
  "actual_used": 54548,
  "actual_free": 196232
},
```


CPU stats

CPU “steal” time

Total / used / free memory

Per process stats


```
publish.go:68: DBG  Publish: {
  "count": 1,
  "proc.cpu": {
    "user": 0,
    "user_p": 0,
    "system": 0,
    "total": 0,
    "start_time": "Sep09"
  },
  "proc.mem": {
    "size": 5932,
    "rss": 572,
    "rss_p": 0.23,
    "share": 476
  },
  "proc.name": "getty",
  "proc.pid": 1211,
  "proc.ppid": 1,
  "proc.state": "sleeping",
  "shipper": "squeeze64",
  "timestamp": "2015-09-10T16:32:26.602Z",
  "type": "proc"
}
```

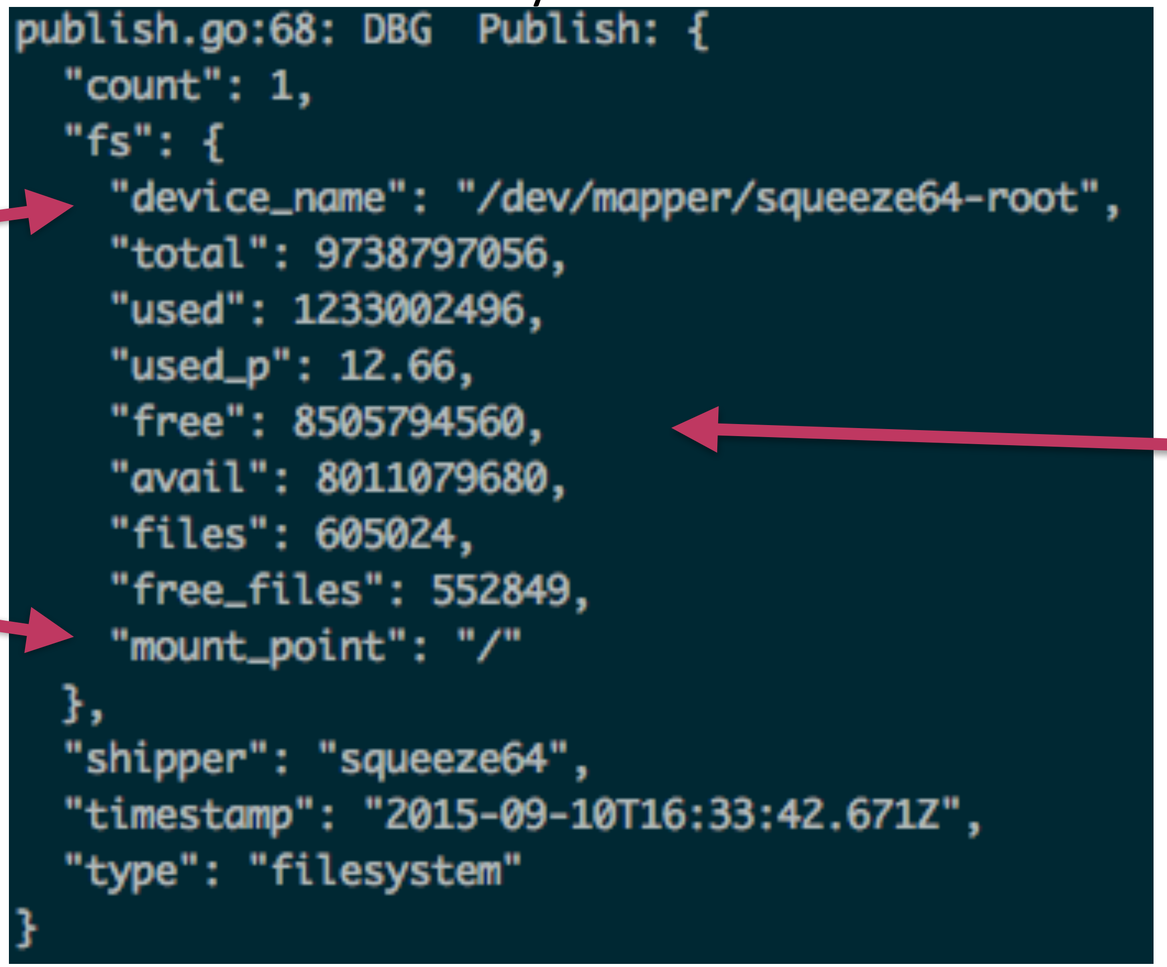

CPU time consumed

Memory used

Process pid, name, parent pid, etc.

Topbeat output objects

File system stats

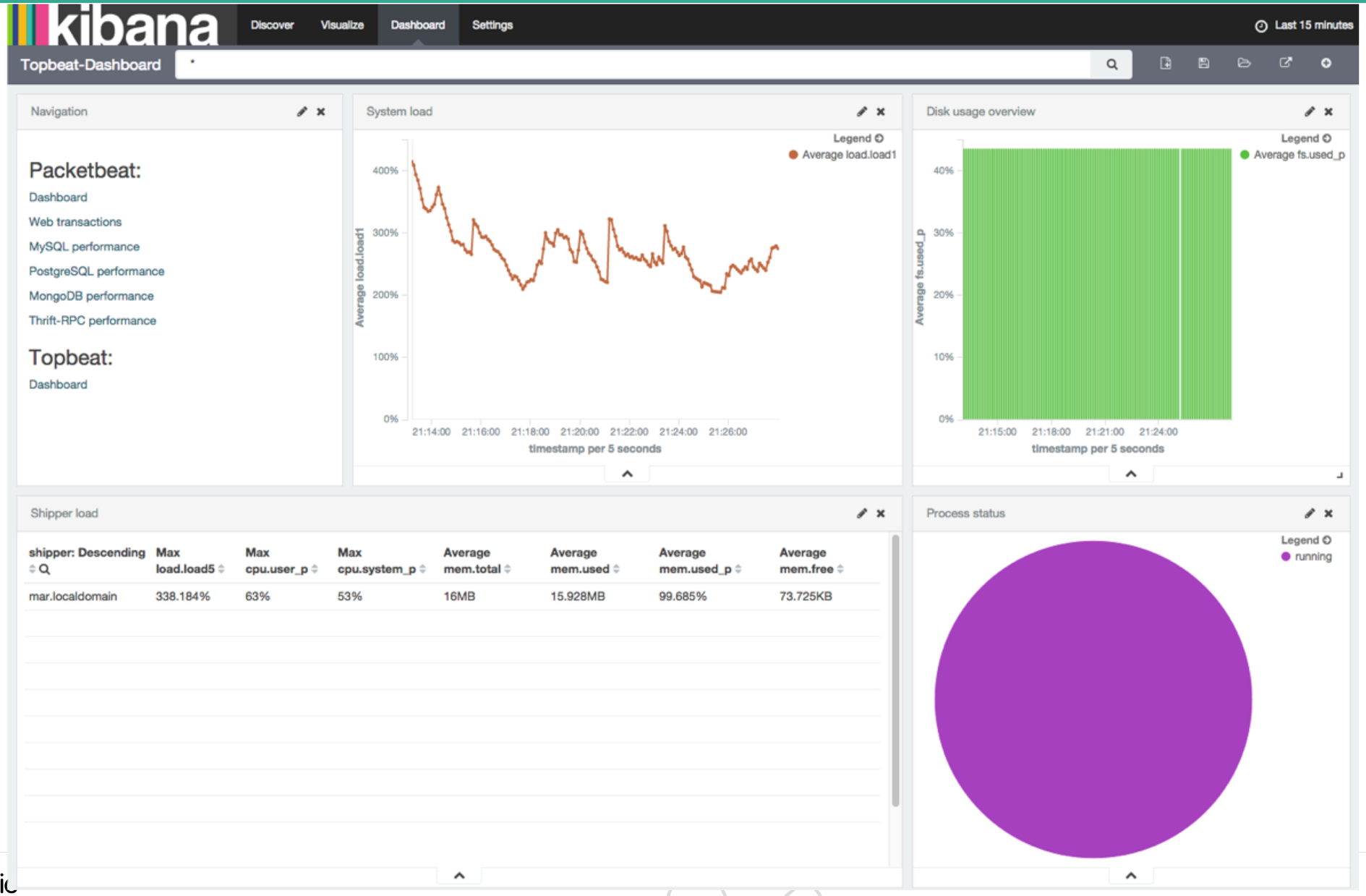


Device name

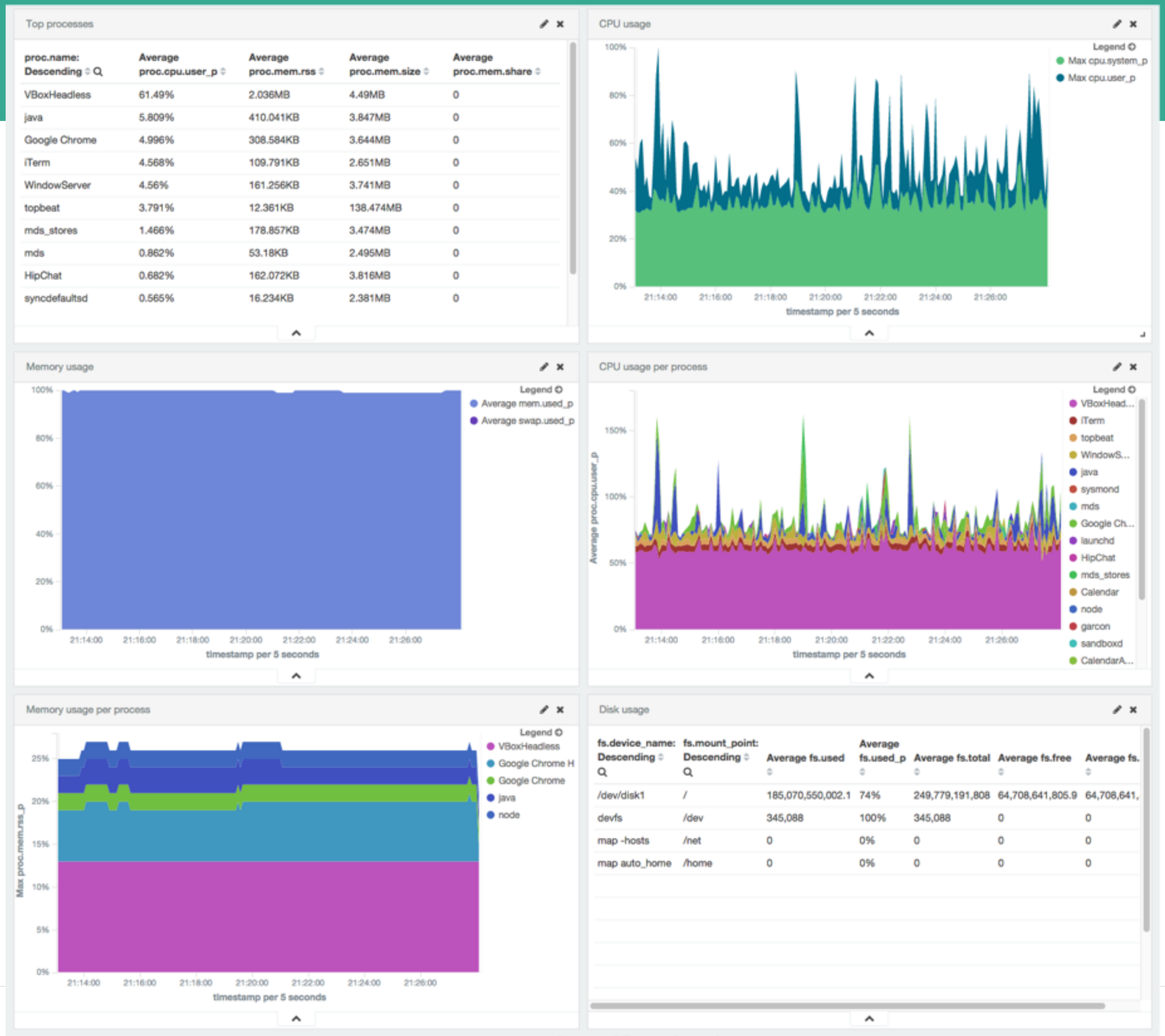
Mount point

Total, used, free disk space

Visualize data with Kibana

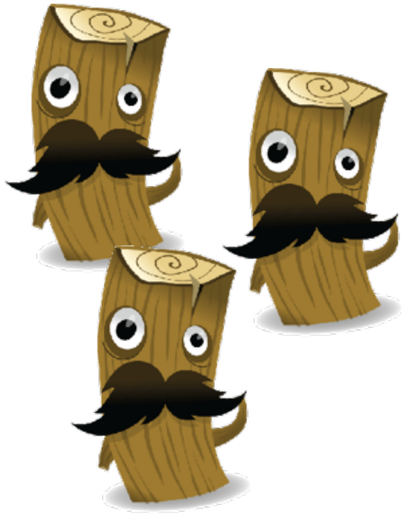


Visualize


Top processes
proc.name: Descending
Average proc.cpu.user_p
Average proc.mem.rss
Average proc.mem.size
Average proc.mem.share
VBoxHeadless 61.49% 2.036MB 4.49MB 0
java 5.809% 410.041KB 3.847MB 0
Google Chrome 4.996% 308.584KB 3.644MB 0
iTerm 4.568% 109.791KB 2.651MB 0
WindowServer 4.56% 161.256KB 3.741MB 0
topbeat 3.791% 12.361KB 138.474MB 0
mds_stores 1.466% 178.857KB 3.474MB 0
mds 0.862% 53.18KB 2.495MB 0
HipChat 0.682% 162.072KB 3.816MB 0
syncdefaultsd 0.565% 16.234KB 2.381MB 0
CPU usage
Legend
Max cpu.system_p
Max cpu.user_p
100%
80%
60%
40%
20%
0%
21:14:00
21:16:00
21:18:00
21:20:00
21:22:00
21:24:00
21:26:00
timestamp per 5 seconds
Memory usage
Legend
Average mem.used_p
Average swap.used_p
timestamp per 5 seconds
CPU usage per process
Average proc.cpu.user_p
150%
100%
50%
VBoxHead...
iTerm
topbeat
WindowS...
java
sysmond
mds
Google Ch...
launchd
HipChat
mds_stores
Calendar
node
garcon
sandboxd
CalendarA...
Memory usage per process
Max proc.mem.rss_p
25%
20%
15%
10%
5%
VBoxHeadless
Google Chrome H
Google Chrome
java
node
Disk usage
fs.device_name: Descending
fs.mount_point: Descending
Average fs.used
Average fs.used_p
Average fs.total
Average fs.free
Average fs.
/dev/disk1 / 185,070,550,002.1 74% 249,779,191,808 64,708,641,805.9 64,708,641,
devfs /dev 345,088 100% 345,088 0 0
map -hosts /net 0 0% 0 0 0
map auto_home /home 0 0% 0 0 0

Filebeat

- A “Beat” based on the Logstash-Forwarder source code
- Do one thing well:
  - Send log files to Logstash & Elasticsearch
- Light on consumed resources
- Easy to deploy on multiple platforms



Filebeat UP AND RUN

1.Download

curl -L -O https://download.elastic.co/beats/filebeat/filebeat-1.0.0-beta4-darwin.tgz

tar xzvf filebeat-1.0.0-beta4-darwin.tgz

2.Run

./filebeat -c filebeat.yml

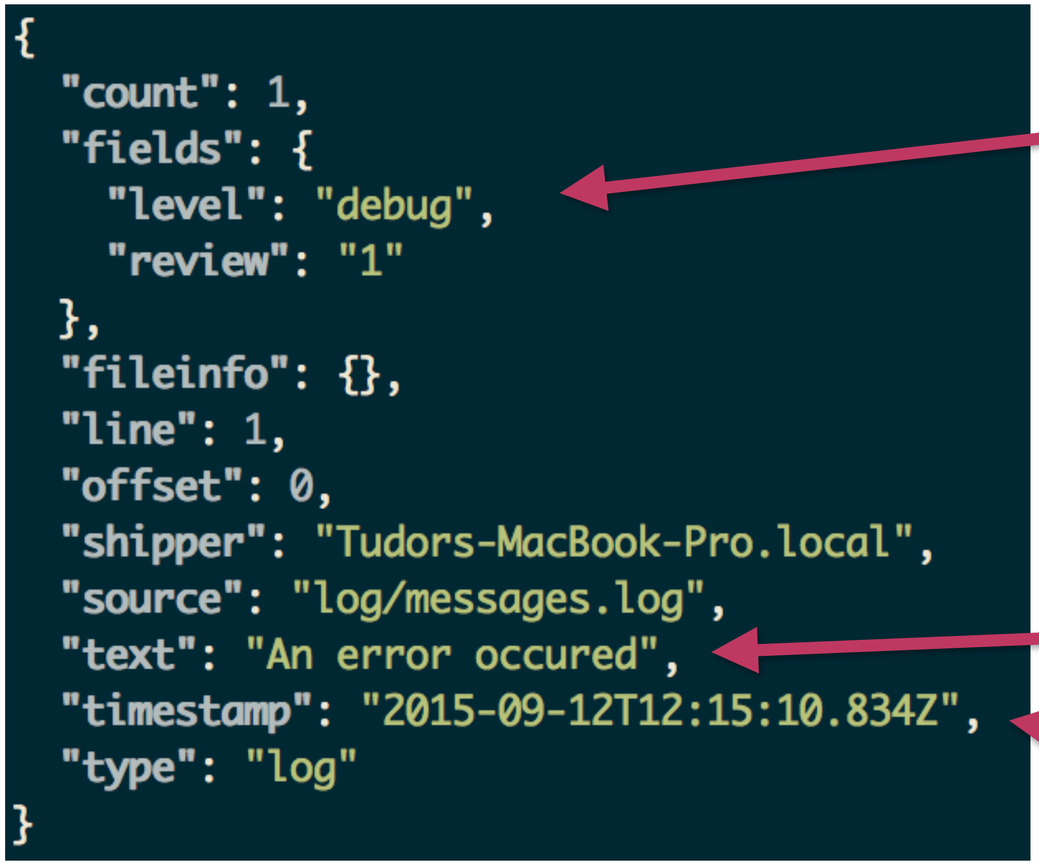
filebeat.yml

```
filebeat:
  prospectors:
    -
      paths:
        - /var/log/system.log
        - /var/log/wifi.log
    -
      paths:
        - /var/log/apache/*
```

Command Help

```
Usage of ./filebeat:
  -N Disable actual publishing for testing
  -c string
     Configuration file (default "/etc/filebeat/filebeat.yml")
  -cpuprofile string
     Write cpu profile to file
  -d string
     Enable certain debug selectors
  -e Output to stdout and disable syslog/file output
  -memprofile string
     Write memory profile to this file
  -test
     Test configuration and exit.
  -v Log at INFO level
  -version
     Print version and exit
```

Filebeat JSON output

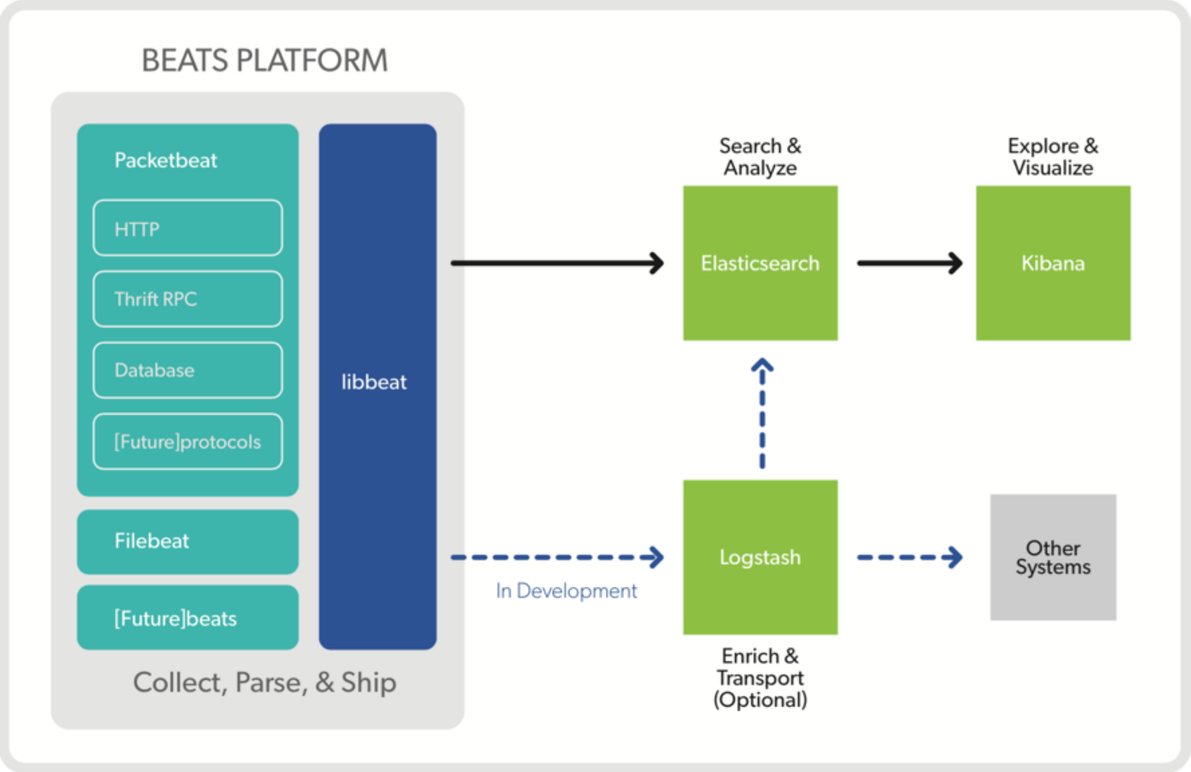


The log level

The log message

The timestamp

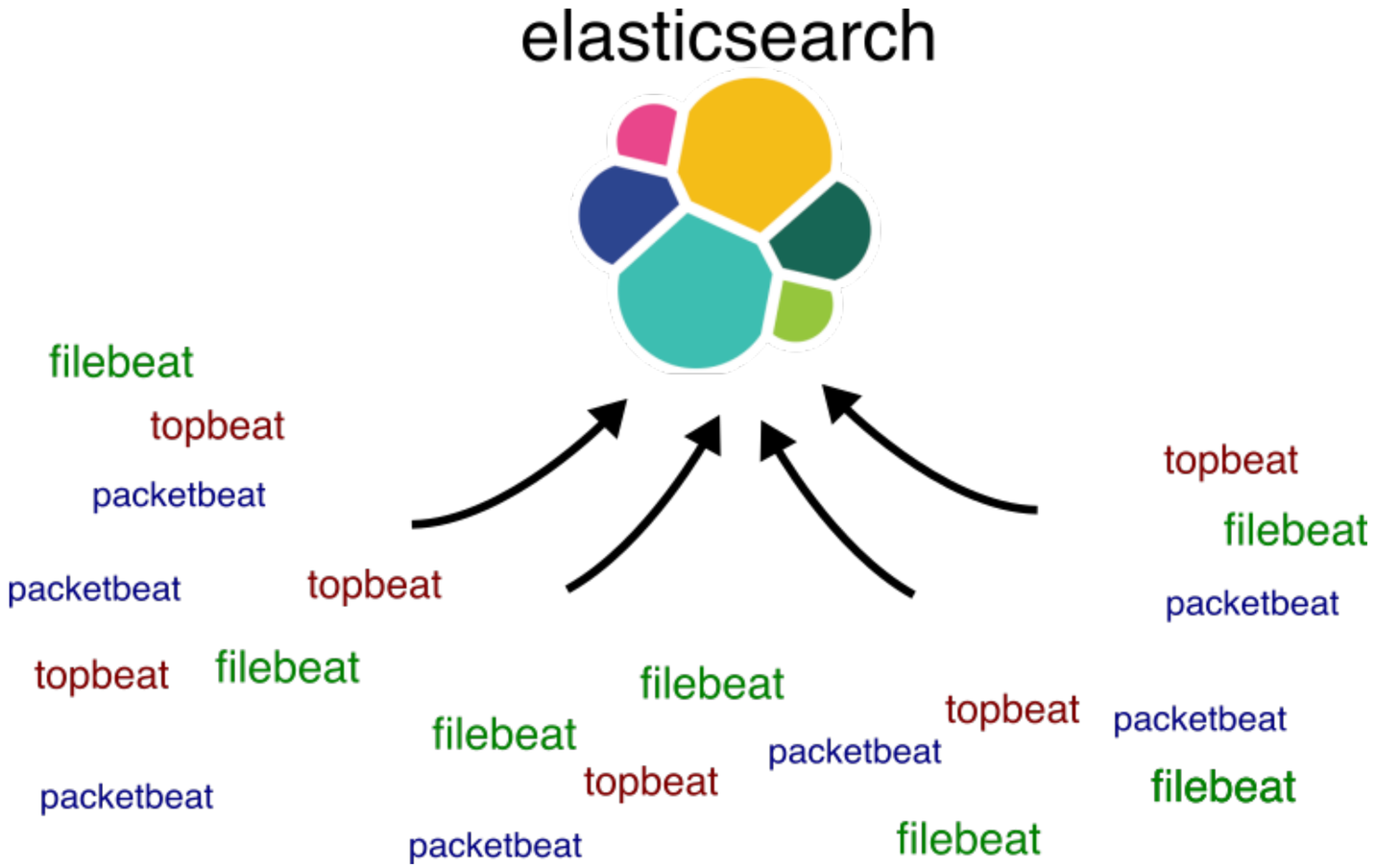
Beats have libbeat in common



- Go library
- Provides common things for all Beats:
  - logging, service handling, configuration file handling, CLI flags
  - Outputs and filters

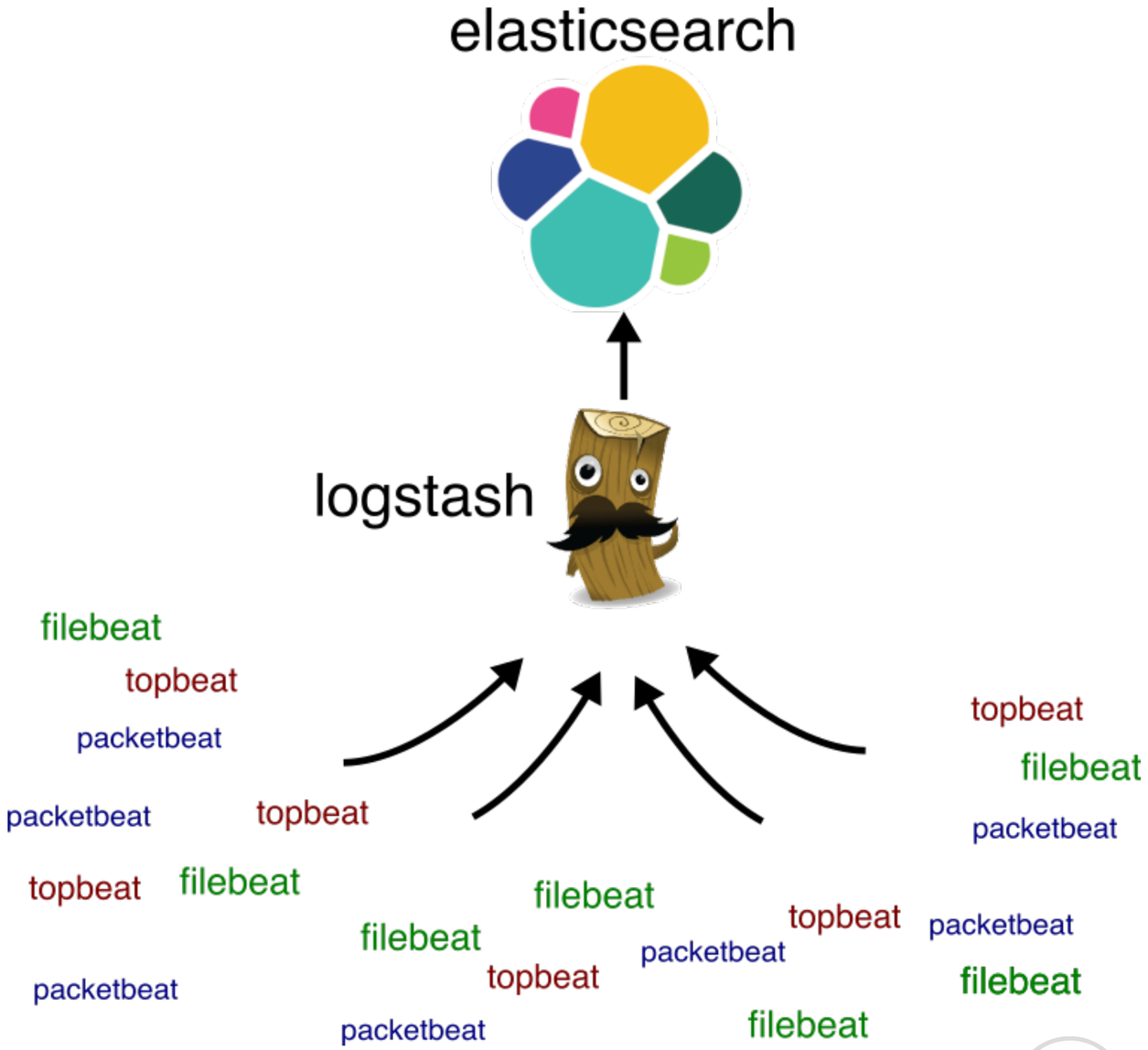
Dev guide for creating a new Beat: https://www.elastic.co/guide/en/beats/libbeat/current/index.html

Deployment: directly to ES



- Option 1: Insert directly into Elasticsearch via the bulk API
- Security can be provided via Shield and HTTPs

Deployment: Send to Logstash



- Option 2: Insert via Logstash
- Uses the Lumberjack protocol which offers security
- Gives the opportunity of enriching or modifying the data

Question

- What we can do with beats?
  - SMTP beats?
  - LVS beats?
  - … …

Thanks

- Live demo: http://demo.elastic.co/packetbeat/
- Thanks!

elastic

elastic