



David Pilato
Developer | Evangelist
@dadoonet

And the beats go on!



Elastic Stack

User Interface



Kibana

Store, Index,
& Analyze



Elasticsearch



X-Pack

Security

Alerting

Monitoring

Graph

Ingest



Logstash



Beats



Elastic Cloud

Elastic Subscriptions: Product, Experience, & Support



Elastic Stack

Open Source

Elasticsearch

Kibana

Logstash

Beats

Plugins

Security (Shield)

Alerting (Watcher)

Monitoring (Marvel)

Elasticsearch as a Service (Found)



Expertise and Support

Development Production

Technical Guidance

- Architecture (hardware/software)
- Cluster management (tuning)
- Index / shard design
- Query optimization
- Integration with other products
- Backup and HA strategy
- Dev to production migration / upgrades
- Best practices

Troubleshooting & Support

- Dedicated, hands-on SLA-based support
- Analysis of internal logs
- Proactively monitoring of clusters
- Escalation to engineering team



beats

**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

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

Examples of operational data



Diagram illustrating three types of operational data and the associated Elastic Beats:

- wire data** (Teal circle) associated with **Packetbeat** (light gray rectangle).
- system stats** (Yellow circle) associated with **Metricbeat** (light gray rectangle).
- logs** (Pink circle) associated with **Filebeat** and **Winlogbeat** (light gray rectangle).

wire data

Packetbeat

system stats

Metricbeat

logs

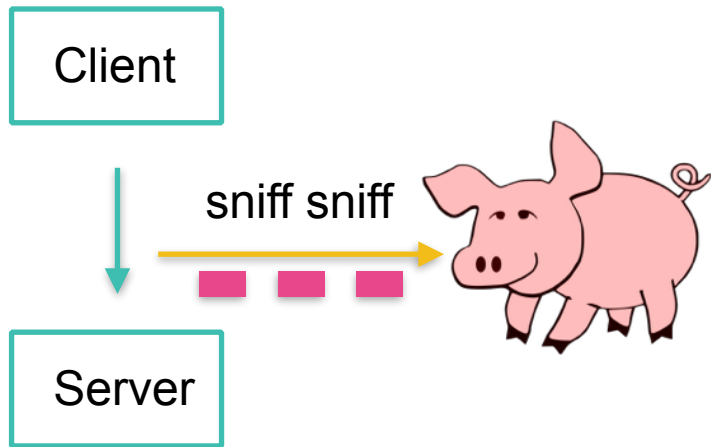
Filebeat
Winlogbeat

Packetbeat

Captures insights from
network packets

```
8:31.720908 IP 192.168.0.8.61563 > 52.91.152.165.443: Flags [P.], seq 200:4032, options [nop,
8:31.763428 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 3009:4427, ack 931, win 40
length 1418
8:31.763429 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 4427:4806, ack 931, win 4
length 379
8:31.812093 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], ack 4427, win 1644, options [n
8:31.812097 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], ack 4806, win 1642, options [n
8:31.968159 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 931:991, ack 4806, win 16
length 60
8:31.968204 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 991, win 4094, options [nop
8:31.971541 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 4806:5768, ack 991, win 4
length 962
8:31.971619 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 5768:5987, ack 991, win 4
length 219
8:32.021961 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], ack 5768, win 1646, options [n
8:32.021964 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], ack 5987, win 1645, options [n
8:32.070031 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 991:1068, ack 5987, win 1
length 77
8:32.070037 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1068:1246, ack 5987, win
length 178
8:32.070168 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 1068, win 4093, options [n
8:32.070268 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 1246, win 4090, options [n
8:32.070948 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1246:1444, ack 5987, win
length 198
8:32.070955 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1444:1490, ack 5987, win
length 46
8:32.071061 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 1444, win 4089, options [n
8:32.071061 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 1490, win 4088, options [n
8:32.072967 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], seq 5987:6033, ack 1490, win
length 46
8:32.120485 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], ack 6033, win 1653, options [n
8:32.183536 IP 192.168.0.8.61645 > 52.91.152.165.443: Flags [P.], seq 102:203, ack 266, win 4096
length 101
8:32.457241 IP 52.91.152.165.443 > 192.168.0.8.61645: Flags [P.], ack 203, win 122, options [nop,
8:32.457247 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1490:1540, ack 6033, win
length 50
8:32.457247 IP 64.233.166.189.443 > 192.168.0.8.61563: Flags [P.], seq 1540:1600, ack 6033, win
length 60
8:32.457385 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 1540, win 4094, options [n
8:32.457385 IP 192.168.0.8.61563 > 64.233.166.189.443: Flags [P.], ack 1600, win 4092, options [n
8:34.349331 IP 192.168.0.8.51759 > 52.22.148.39.443: Flags [P.], seq 1:38, ack 325, win 4096, op
n 37
8:34.518786 IP 52.22.148.39.443 > 192.168.0.8.51759: Flags [P.], ack 38, win 136, options [nop, no
8:34.812485 IP 52.91.152.165.443 > 192.168.0.8.61645: Flags [P.], seq 266:415, ack 203, win 122,
length 149
8:34.812533 IP 192.168.0.8.61645 > 52.91.152.165.443: Flags [P.], ack 415, win 4094, options [n
```

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

Packetbeat: Available decoders



HTTP



Thrift-RPC



DNS



MySQL



Memcache



AMQP



PostgreSQL



MongoDB



Add your own



Redis



ICMP

Processes: 367 total, 3 running, 6 stuck, 1338 sleeping, 1833 threads
 Load Avg: 2.79, 2.69, 2.66 CPU usage: 35.77% user, 8.25% sys, 55.96% idle
 SharedLibs: 161M resident, 22M data, 10M linkedit. MemRegions: 132196 total, 5764M resident, 90M private
 PhysMem: 13G used (3762M wired), 2581M unused.
 VM: 2084G vsize, 527M framework vsize, 56231732(0) swapins, 59291027(0) swapouts.
 Networks: packets: 66492613/48G in, 57364574/31G out. Disks: 6744547/369G read, 34720560/883G written.

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORTS	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS
64667	burn	89.3	04:00.32	5/1	0	15	580K	0B	0B	64667	63666	running	*0[1]
29105	java	21.0	92:05.82	74	0	186	369M	0B	65M	29105	98883	sleeping	*0[2]
15112	topbeat	18.7	02:56:57	12	0	61	9900K	0B	1108K	15111	15111	sleeping	*0[1]
325	iTerm	12.8	02:18:11	12	4	366	98M	4096B	40M	325	1	sleeping	*0[19746]
65237	top	7.4	00:08.34	1/1	0	24	7012K	0B	0B	65237	18734	running	*0[1]
0	kernel_task	4.6	23:39:40	220/4	0	2	1719M+	0B	0B	0	0	running	0[0]
61	mds	3.3	01:48:47	10	6	329+	14M	0B	34M	61	1	stuck	*0[1]
194	mds_stores	3.1	02:03:27	10	7	76	20M+	1184K	41M	194	1	stuck	*0[1]
186	WindowServer	1.8	14:39:27	5	2	1224	79M	5324K	267M	186	1	sleeping	*0[1]
637-	Dropbox	1.5	93:04.38	84	0	385	86M	0B	44M	637	1	sleeping	*0[53781]
97	hidd	1.4	93:44.84	6	2	98-	3300K-	0B	1400K	97	1	sleeping	*0[1]
53155	mdworker	1.1	01:05.39	4	0	66	14M	0B	1268K	53155	1	sleeping	*0[1]
53158	mdworker	1.0	01:06.41	4	0	62	12M	0B	1000K	53158	1	sleeping	*0[1]
3431	Slack	0.9	03:36:09	19	1	456	611M-	27M	331M	3431	1	sleeping	*0[22918]
75966	python2.7	0.7	28:17.99	3	1	33	8240K	0B	13M	75966	75959	stuck	*0[1]
53159	mdworker	0.7	01:04.20	4	0	62	17M	0B	1144K	53159	1	sleeping	*0[1]
53157	mdworker	0.6	01:06.09	4	0	62	15M	0B	948K	53157	1	sleeping	*0[1]
324-	zoom.us	0.5	13:07:07	13	0	44995	124M	0B	179M	324	1	sleeping	*48[18]
23794-	dbfsevents	0.4	01:45.43	1	0	7	4168K	0B	140K	637	23793	sleeping	*0[1]
75965	python2.7	0.4	22:28.90	2	0	15	6860K	0B	11M	75965	75959	sleeping	*0[1]
58424	Google Chrom	0.3	01:23.17	12	0	111	117M-	0B	37M	316	316	sleeping	*0[2]
46	fsevents	0.3	30:30.91	13	0	306	4796K	0B	4104K	46	1	sleeping	*0[1]
23795-	dbfsevents	0.2	01:07.97	1	0	7	32K	0B	152K	637	23794	sleeping	*0[1]
48135	Google Chrom	0.2	06:38.73	15	0	60	43M	0B	18M	316	316	sleeping	*0[1]
65300	screencaptur	0.2	00:00.14	2	0	52	2220K	20K	0B	336	336	sleeping	*0[1]
89	mDNSResponde	0.2	46:50.49	7	2	88	2096K	0B	1124K	89	1	sleeping	*0[1]
316	Google Chrom	0.1	20:08:22	48	1	1998	943M	736K	1245M	316	1	sleeping	*0[10133]
39271	inkscape-bin	0.1	03:05.45	7	0	39	7844K	0B	168M	39268	39268	sleeping	*0[1]
60240	Google Chrom	0.1	01:33.59	9	0	110	35M+	0B	34M	316	316	sleeping	*0[2]
212	symptomsd	0.1	16:32.42	4	2	102	2184K	0B	2112K	212	1	sleeping	0[66876]
23793-	dbfsevents	0.0	00:23.07	1	0	12	36K	0B	180K	637	637	sleeping	*0[1]
61533	Google Chrom	0.0	00:07.55	13	0	123	98M	0B	0B	316	316	sleeping	*0[2]
16578	Google Chrom	0.0	06:12.31	13	0	155	150M+	0B	50M	316	316	sleeping	*0[2]
639-	Google Photo	0.0	15:39.98	14	0	262	4648K	0B	34M	639	1	sleeping	*0[27266]
13660	Finder	0.0	16:11.28	9	2	423	115M	0B	239M	13660	1	sleeping	*0[2627]
58571	com.apple.ap	0.0	00:13.14	4	1	263	20M	512K	19M	58571	1	sleeping	*0[297]
54793	Google Chrom	0.0	00:18.73	13	0	121	80M	0B	16M	316	316	sleeping	*0[2]
54832	com.apple.ap	0.0	01:02.49	4	1	263	9248K	0B	26M	54832	1	sleeping	*0[1856]
62861	Google Chrom	0.0	00:01.85	12	0	112	76M	0B	0B	316	316	sleeping	*0[2]
3391	com.apple.ap	0.0	02:43.68	4	1	256	4992K	0B	16M	3391	1	sleeping	*0[13618]

Metricsbeat

Like the Unix **top** command but sends the output periodically to Elasticsearch. Also works on Windows.

Topbeat: Exported data

System wide

- system load
- total CPU usage
- CPU usage per core
- Swap, memory usage

Per process

- state
- name
- command line
- pid
- CPU usage
- memory usage

Disk usage

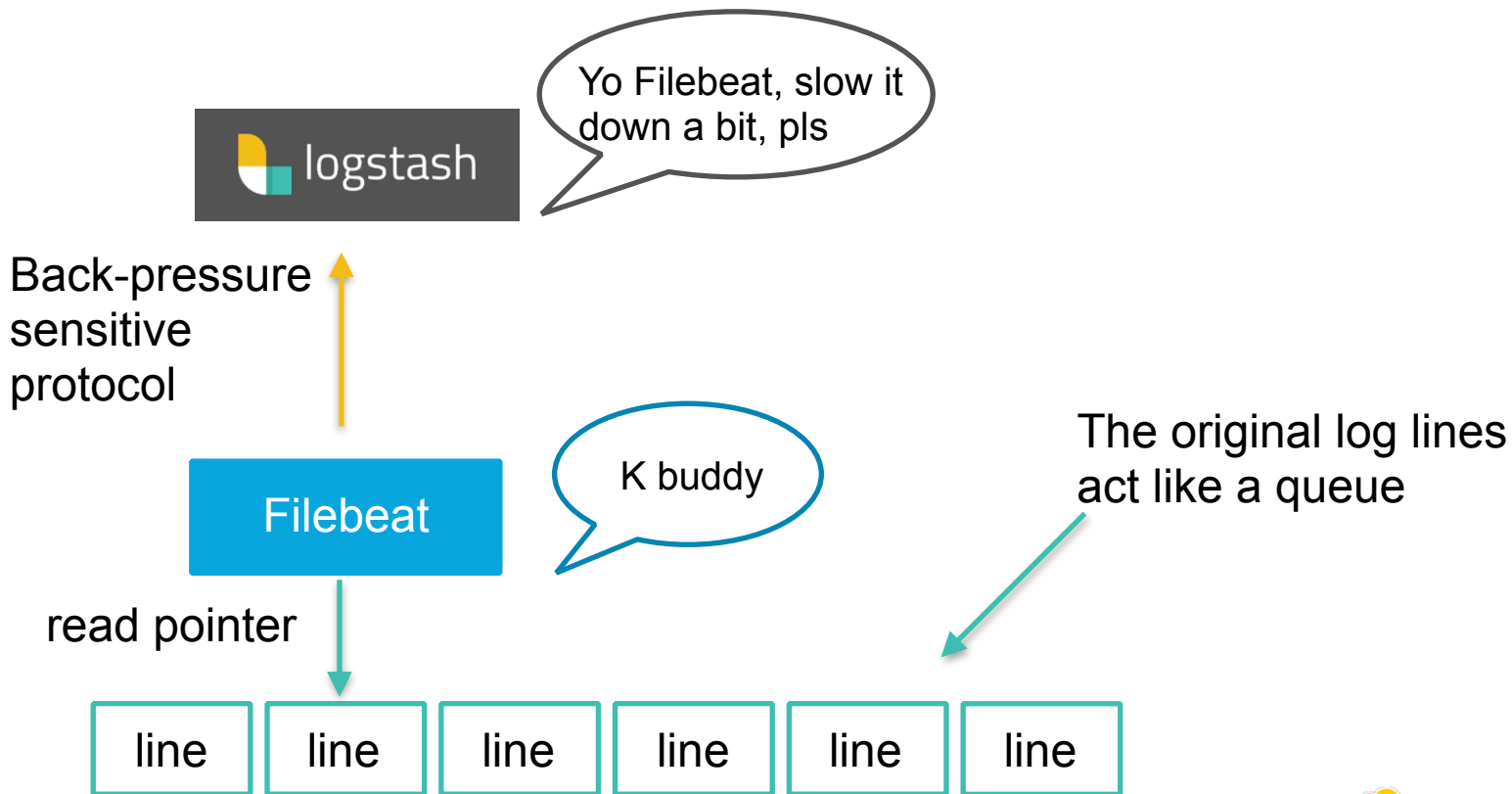
- available disks
- used, free space
- mounted points

Filebeat

Forwards log lines to
Elasticsearch

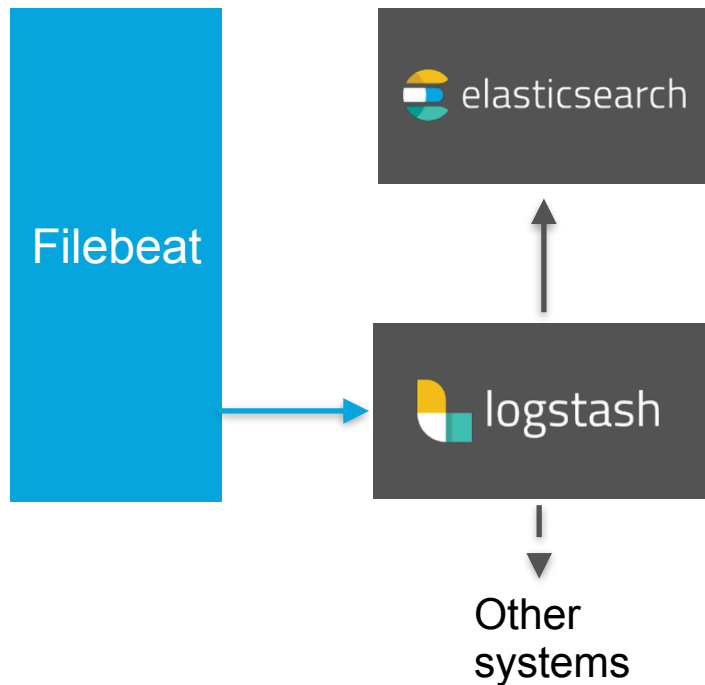
```
2016/02/09 21:20:42.414572 client.go:257: WARN Can not index event (status=400):
2 105 110 103 95 101 120 99 101 112 116 105 111 110 34 44 34 114 101 97 115 111 1
16 104 32 102 97 105 108 117 114 101 115 32 123 91 109 97 112 112 101 114 32 91 1
2 32 100 105 102 102 101 114 101 110 116 32 116 121 112 101 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
```

Filebeat: Never lose a log line



Filebeat: Parse logs with Logstash

Parse logs with Logstash



- Filebeat sends out unparsed log lines
- Use filters from Logstash to parse the log lines
- Flexible, with conditionals & custom filters
- Forward data to other systems using the Logstash output plugins

Filebeat: Parse logs with Ingest Node

Parse logs with Ingest node in Elasticsearch













- Filebeat sends out unparsed log lines directly to Elasticsearch
- Use Ingest Node processors (grok, geoip...) to parse the log lines
- Easier to setup

Winlogbeat

Forwards Windows Event logs
to Elasticsearch

System Number of events: 728

Level	Date and Time	Source	Event...	Task C
 Information	1/13/2015 9:26:35 AM	Service Control Manager	7036	None
 Information	1/13/2015 9:26:35 AM	Service Control Manager	7036	None
 Information	1/13/2015 9:26:35 AM	Service Control Manager	7036	None
 Information	1/13/2015 9:26:35 AM	Service Control Manager	7036	None
 Information	1/13/2015 9:26:35 AM	Service Control Manager	7036	None
 Information	1/13/2015 9:26:33 AM	Ntfs (Microsoft-Windows-N...	98	None
 Information	1/13/2015 9:26:33 AM	Kernel-Processor-Power (Mi...	55	(47)
 Information	1/13/2015 9:26:33 AM	Kernel-Processor-Power (Mi...	55	(47)
 Information	1/13/2015 9:26:32 AM	Kernel-Power	508	(159)
 Information	1/13/2015 9:26:32 AM	FilterManager	6	None

< III

Event 7036, Service Control Manager

General Details

The Plug and Play service entered the running state.

Log Name: System

Source: Service Control Manager

Event ID: 7036

Level: Information

User: N/A

OpCode: Info

Logged: 1/13/2015 9:26:35 AM

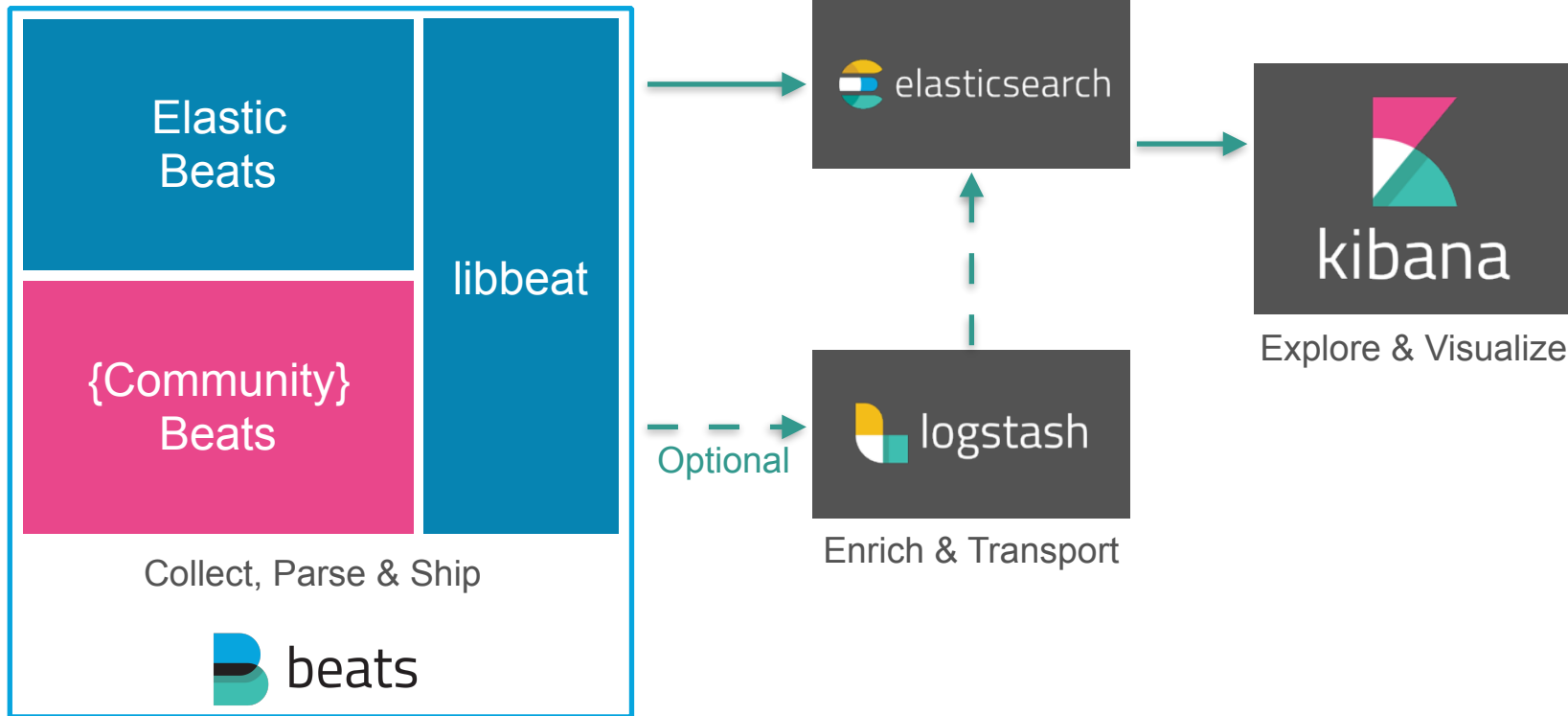
Task Category: None

Keywords: Classic

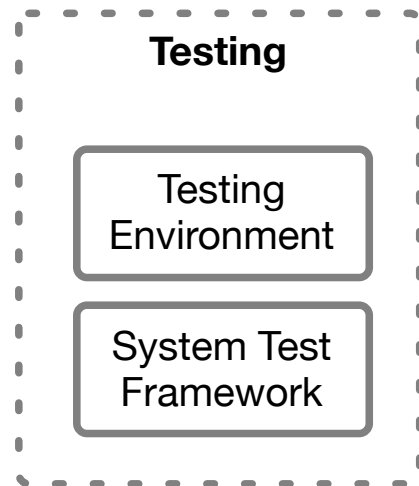
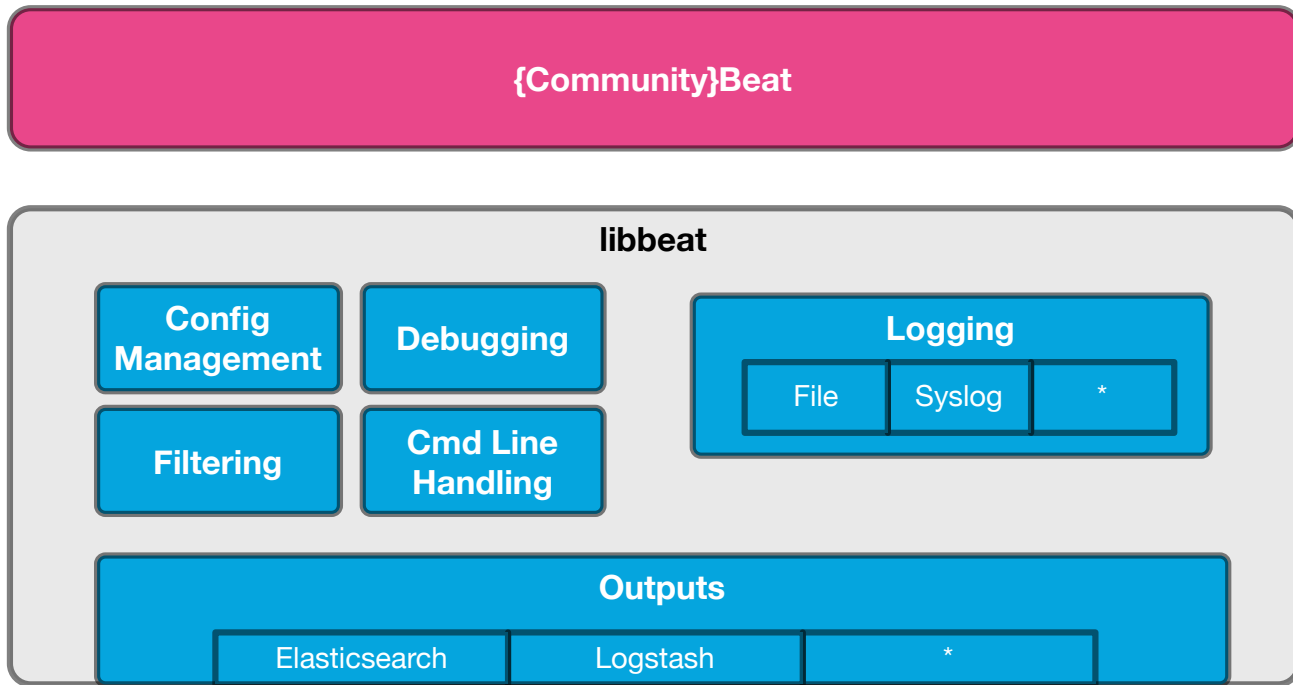
Computer: vagrant-2012-r2

More Information: [Event Log Online Help](#)

Beats Platform



Architecture Overview - libbeat



Beats Packer

Produces RPMs, DEBs, ...

<https://github.com/elastic/beats-packer>



Super-Sound-Single

45 rpm

**Disco-
Remix**

THE WHISPERS

RCA

And The Beat Goes On 7:30
Side 1

AF

Can You Do The Boogie 6:07
Side 2

FC 1895



Super-Sound-Single

45 rpm

**Disco-
Remix**

THE WHISPERS

RCA

And The Beat Goes On 7:30
Side 1

AF

Can You Do The Boogie 6:07
Side 2

FC 1895





elastic

metricbeat, packetbeat and soundbeat

<https://github.com/dadoonet/soundbeat>





thanks!

<https://github.com/dadoonet/soundbeat>

