



---

# Analyze redis slowlog using elastic stack

rockybean@yourdream

2017-05-14



```
config set slowlog-log-slower-than 20000  
config set slowlog-max-len 100
```

```
slowlog get [len]  
slowlog reset
```

```
redis 127.0.0.1:6379> slowlog get 2  
1) 1) (integer) 14  
   2) (integer) 1309448221  
   3) (integer) 15  
   4) 1) "ping"  
2) 1) (integer) 13  
   2) (integer) 1309448128  
   3) (integer) 30  
   4) 1) "slowlog"  
      2) "get"  
      3) "100"
```

# elastic stack

---



# Beats

---



## The Lightweight Shippers

- FileBeat
- Metricbeat
- Packetbeat
- Heartbeat
- WinlogBeat

## Build your own beat

- <https://www.elastic.co/blog/build-your-own-beat>

## Rsbeat

rsbeat

Ships redis slow logs to elasticsearch and analyze by Kibana.

Collect redis slowlog from redis servers

<https://github.com/yourdream/rsbeat>

# Rsbeat Usage rsbeat.yml

---

## rsbeat:

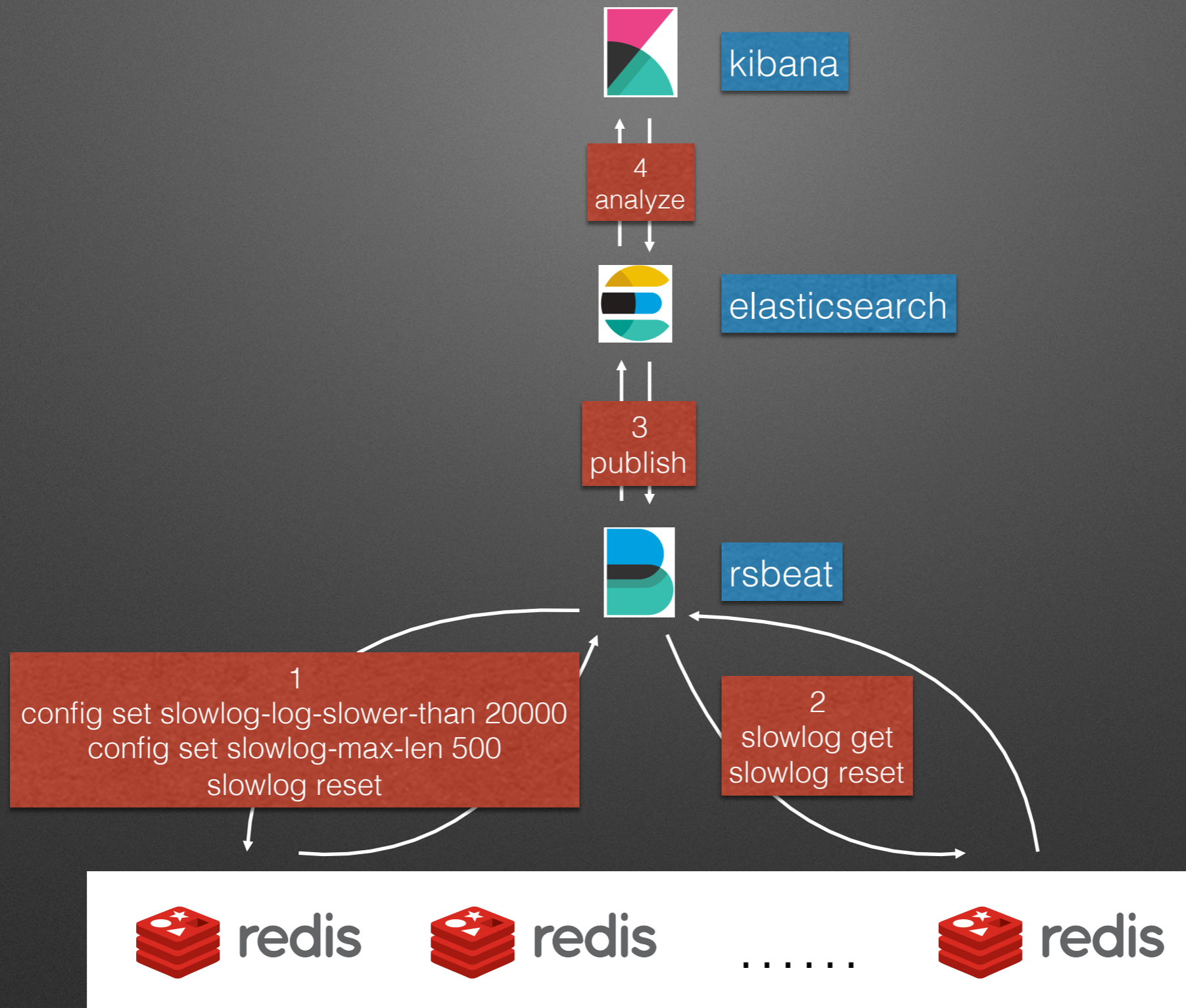
- period: 1s
- redis: ["127.0.0.1:6379"]
- slowerThan: 20000

## output.elasticsearch:

- hosts: ["localhost:9200"]

**rsbeat -c rsbeat.yml -e -d "\*"**

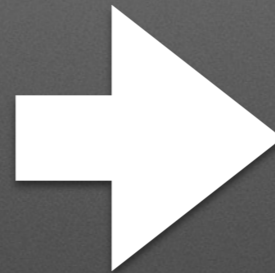
# Rsbeat Workflow



# Rsbeat Document

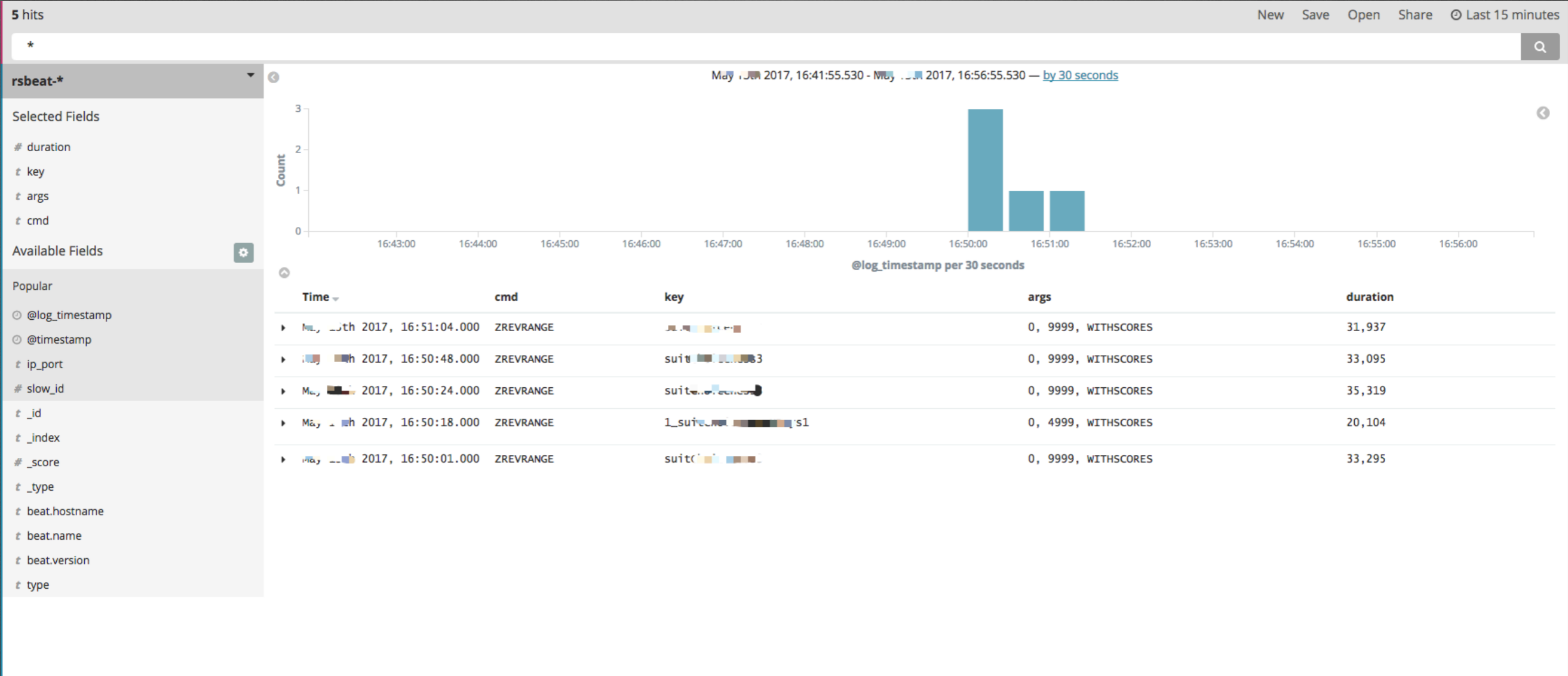


```
redis 127.0.0.1:6379> slowlog get 2
1) 1) (integer) 14
    2) (integer) 1309448221
    3) (integer) 15
    4) 1) "ping"
2) 1) (integer) 13
    2) (integer) 1309448128
    3) (integer) 30
    4) 1) "slowlog"
      2) "get"
      3) "100"
```



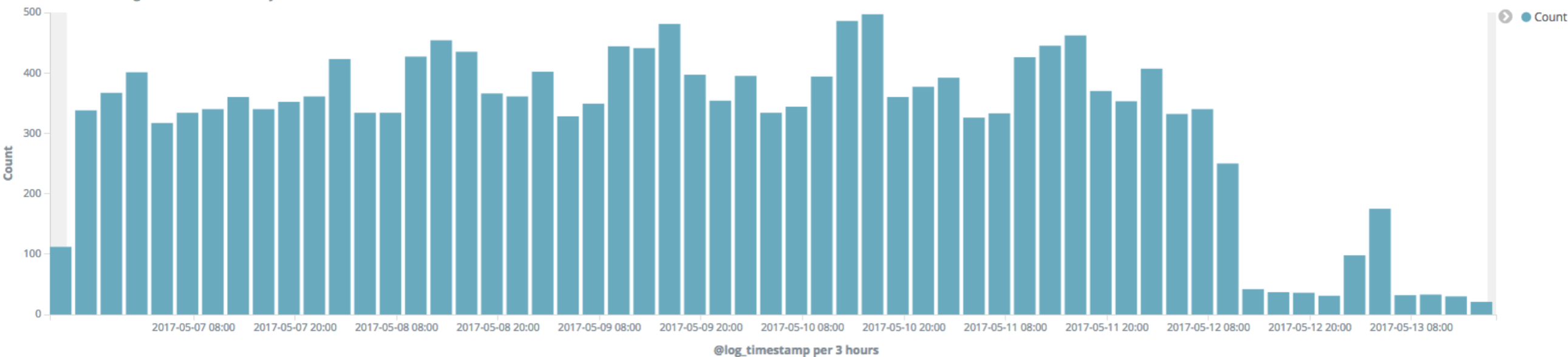
```
[
  {
    "@timestamp": "2017-05-14T00:23:21.385Z",
    "@log_timestamp": "2011-06-30 23:35:28Z",
    "slow_id": 13,
    "cmd": "slowlog",
    "key": "get",
    "args": [
      "100"
    ],
    "duration": 30,
    "ip_port": "127.0.0.1:6379",
    "type": "rsbeat"
  },
  {
    "@timestamp": "2017-05-14T00:23:21.385Z",
    "@log_timestamp": "2011-06-30T23:37:01Z",
    "slow_id": 14,
    "cmd": "ping",
    "duration": 15,
    "ip_port": "127.0.0.1:6379",
    "key": "",
    "type": "rsbeat"
  }
]
```

# Rsbeat Kibana Show



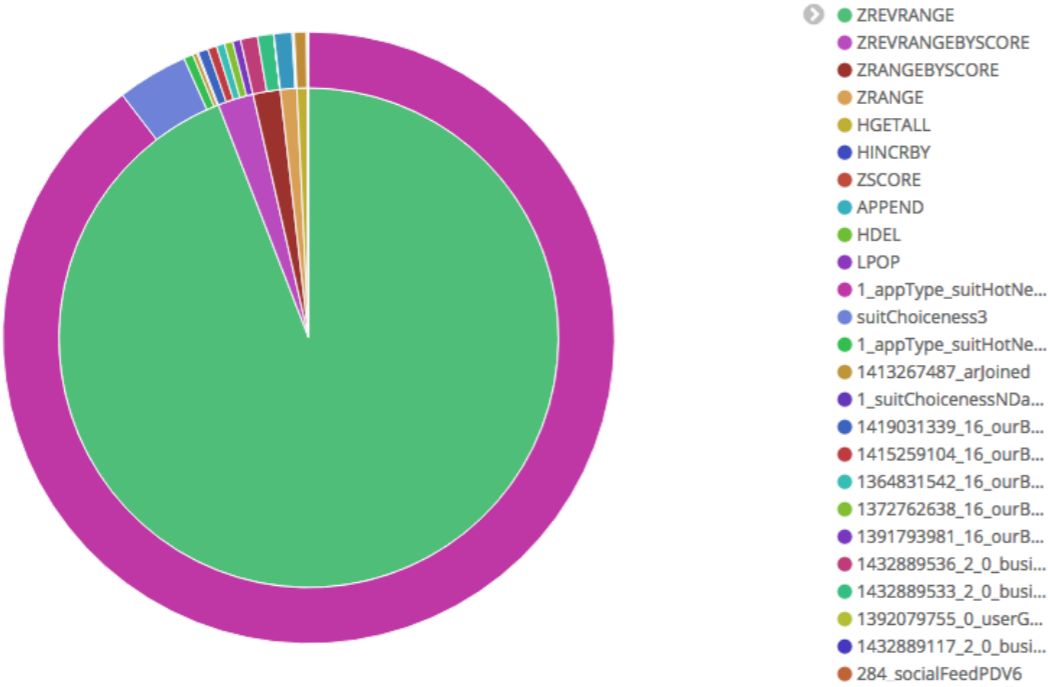
# Rsbeat Kibana Show

rsbeat - redis slowlog count distribution by time



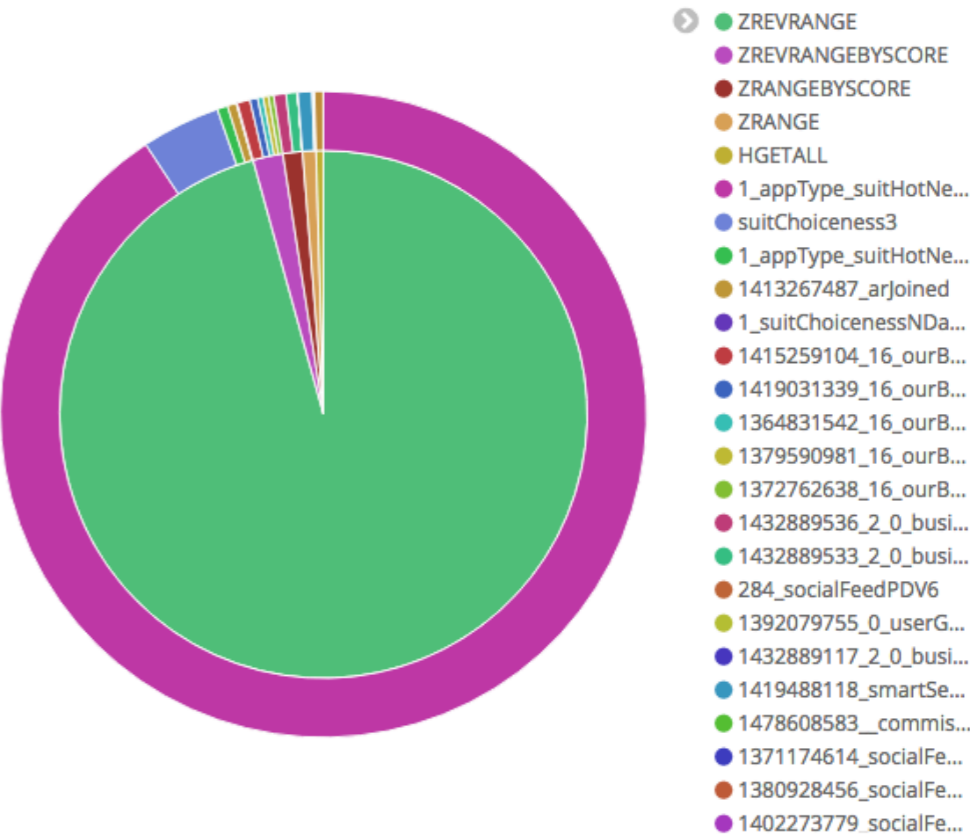
# Rsbeat Kibana Show

rsbeat - redis command and key count distribution



rsbeat - top 10 slow commands and keys by count

rsbeat - redis command and key duration distribution



# Rsbeat Kibana Show

rsbeat - top 10 slow commands and keys by count

ZREVRANGE: cmd: Descending		ZRANGEBYSCORE: cmd: Descending		ZRANGE: cmd: Descending		HGETALL: cmd: Descending		ZREVRANGEBYSCORE: cmd: Descending	
key: Descending ⚡ Q	Count ⚡	key: Descending ⚡ Q	Count ⚡	key: Descending ⚡ Q	Count ⚡	key: Descending ⚡ Q	Count ⚡	key: Descending ⚡ Q	Count ⚡
1_appType_data2	16,083	1432889536_2_0_h...User,Can,inter...t	161	1419488118_...a...User,Can,inter...g	153	spider...	117	1419031339_16_c...ess...?	39
suic...	671	1432889533_2_0_bu...t	153	1478608583_...t	13			1415259104_16_...in2	35
1_appType...	90							1364831542_16_...ata2	32
1413267487_...t	40							1372762638_16_...ata2	32
1_...t	13							1391793981_16_ou...ata2	31
1_00_...t								1379590981_16_ou...ata2	30
								1390614041_16_our...ata2	23
								1389303205_16_ourE...ata2	17
								1391829451_16_ourB...ata2	15
								1384805330_16_ourB...ata2	13

rsbeat - top 10 slow commands and keys by average duration

MGET: cmd: Descending		APPEND: cmd: Descending		ZRANGE: cmd: Descending		ZREVRANGE: cmd: Descending		ZREVRANGEBYSCORE: cmd: Descending	
key: Descending ⚡ Q	Average duration ⚡	key: Descending ⚡ Q	Average duration ⚡	key: Descending ⚡ Q	Average duration ⚡	key: Descending ⚡ Q	Average duration ⚡	key: Descending ⚡ Q	Average duration ⚡
27021599157902995_...	195,907	79bd59079...8a3	68,789	137117...t	65,819	1413267487_...	61,859.5	1415259104_16_...2	46,649.11
27021599158169530_...	40,700			138092...D	65,501	1413269_...	37,188.333	1405794264_...2	35,957
				138487...D	60,967	1_00_rec...t	35,770.714		
				137117...D	44,124	suic...	32,227.782		
				139255...D	40,420	1_appType...3	32,205.111		
				1391475...t	40,135	1_appTyp...2	30,983.081		
				1388035...t	39,328	2_1_app...t	30,182		
				138448789_...t	35,946				
				1381032...t	31,818				
				137792349_...t	31,306				

# Rsbeat Kibana Show

**rsbeat-\***

Data

Options

▶

✕

**metrics**

▶

 Slice Size

Count

**buckets**

▼

 Split Slices

⌵

✕

Aggregation

Terms

Field

cmd

Order By

metric: Count

Order

Descending

Size

10

Custom Label

⌵ Advanced

▼

 Split Slices

⌵

✕

Sub Aggregation

Terms

Field

key

Order By

metric: Count

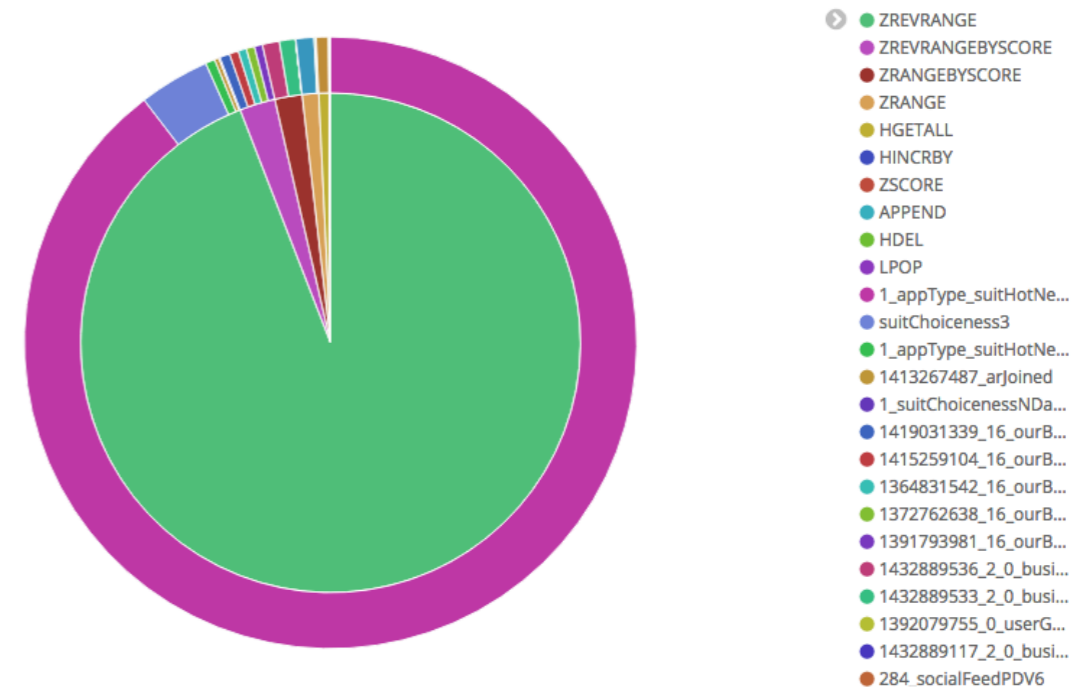
Order

Descending

Size

5

rsbeat - redis command and key count distribution



rsbeat - top 10 slow commands and keys by count

# Rsbeat Sourcecode

---

```
.
├── beater
│   └── rsbeat.go
├── meta
│   ├── beat.yml
│   ├── fields.yml
│   └── kibana
│       ├── dashboard
│       │   └── slowlogDash.json
│       ├── export.json
│       └── index-pattern
│           └── rsbeat.json
├── config
│   ├── config.go
│   └── config_test.go
├── docker
├── docs
│   ├── fields.asciidoc
│   └── index.asciidoc
├── main.go
├── rsbeat-dashboard.json
├── rsbeat-dashboard.png
├── rsbeat.full.yml
├── rsbeat.template-es2x.json
├── rsbeat.template.json
└── rsbeat.yml
```

rsbeat core code

rsbeat es fields config

rsbeat docker

rsbeat config

# Rsbeat Sourcecode

```
func poolInit(server string, slowerThan int) *redis.Pool {
    return &redis.Pool{
        MaxIdle:    3,
        MaxActive:  3,
        IdleTimeout: 240 * time.Second,
        Dial: func() (redis.Conn, error) {
            c, err := redis.Dial("tcp", server, redis.DialConnectTimeout(3*time.Second), redis.DialReadTimeout(3*time.Second))
            if err != nil {
                logp.Err("redis: error occurs when connect %v", err.Error())
                return nil, err
            }
            c.Send("MULTI")
            c.Send("CONFIG", "SET", "slowlog-log-slower-than", slowerThan)
            c.Send("CONFIG", "SET", "slowlog-max-len", 500)
            c.Send("SLOWLOG", "RESET")
            r, err := c.Do("EXEC")

            if err != nil {
                logp.Err("redis: error occurs when send config set %v", err.Error())
                return nil, err
            }

            logp.Info("redis: config set %v", r)
            return c, err
        },
        TestOnBorrow: func(c redis.Conn, t time.Time) error {
            _, err := c.Do("PING")
            logp.Info("redis: PING")
            return err
        },
    }
}
```

# Rsbeat Sourcecode

```
func (bt *Rsbeat) redisc(beatname string, init bool, c redis.Conn, ipPort string) {
    defer c.Close()
    logp.Info("conn:%v", c)

    c.Send("SLOWLOG", "GET")
    c.Send("SLOWLOG", "RESET")
    logp.Info("redis: slowlog get. slowlog reset")

    c.Flush()
    reply, err := redis.Values(c.Receive()) // reply from GET
    c.Receive()                             // reply from RESET

    logp.Info("reply len: %d", len(reply))

    for _, i := range reply {
        rp, _ := redis.Values(i, err)
        var itemLog itemLog
        var args []string
        redis.Scan(rp, &itemLog.slowId, &itemLog.timestamp, &itemLog.duration, &args)
        argsLen := len(args)
        if argsLen >= 1 {
            itemLog.cmd = args[0]
        }
        if argsLen >= 2 {
            itemLog.key = args[1]
        }
        if argsLen >= 3 {
            itemLog.args = args[2:]
        }
        logp.Info("timestamp is: %d", itemLog.timestamp)
        t := time.Unix(itemLog.timestamp, 0).UTC()

        event := common.MapStr{
            "type":      beatname,
            "@timestamp":  common.Time(time.Now()),
            "@log_timestamp": common.Time(t),
            "slow_id":     itemLog.slowId,
            "cmd":         itemLog.cmd,
            "key":         itemLog.key,
            "args":        itemLog.args,
            "duration":    itemLog.duration,
            "ip_port":     ipPort,
        }

        bt.client.PublishEvent(event)
    }
}
```

# Rsbeat Docker docker

---

```
docker-compose up -d elasticsearch kibana
```

```
docker-compose run -e "REDIS_LIST=\"10.0.0.40:6379\"" rsbeat
```

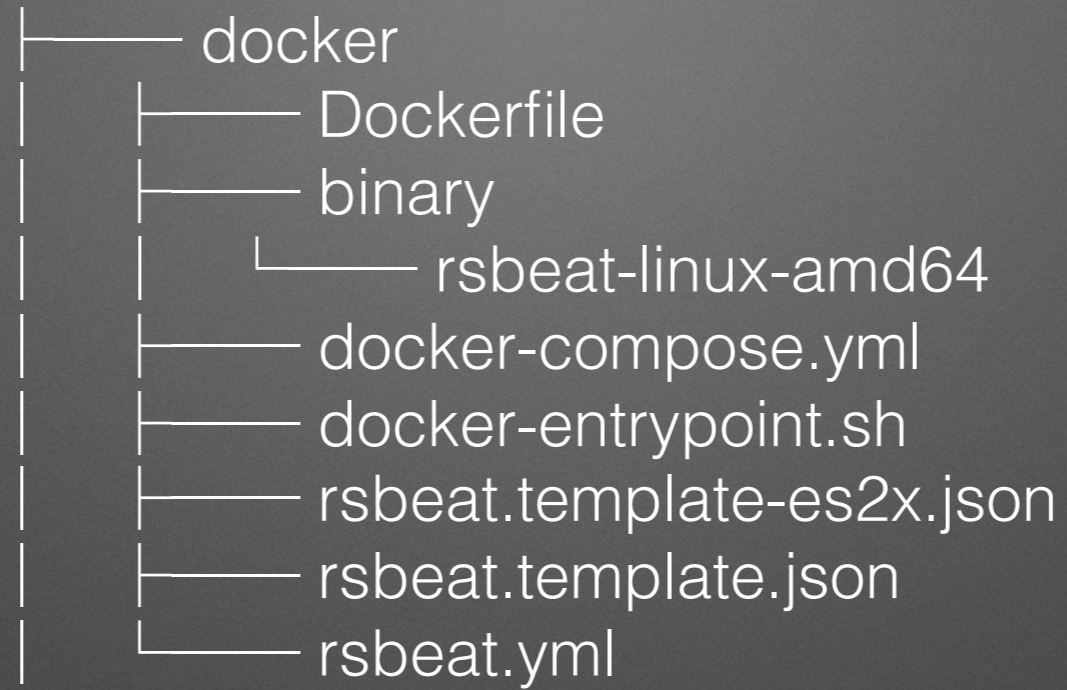
~~127.0.0.1~~

~~localhost~~

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html>
- <https://www.elastic.co/guide/en/kibana/current/docker.html>

# Rsbeat Docker

---



# Rsbeat Docker

## Dockerfile

---

```
FROM busybox
```

```
COPY docker-entrypoint.sh /docker-entrypoint.sh
```

```
COPY binary/rsbeat-linux-amd64 /rsbeat
```

```
COPY rsbeat.yml /rsbeat.yml
```

```
COPY rsbeat.template.json /rsbeat.template.json
```

```
COPY rsbeat.template-es2x.json /rsbeat.template-es2x.json
```

```
ENTRYPOINT ["/docker-entrypoint.sh"]
```

```
CMD ["/rsbeat", "-e", "-d", "*", "-c", "/rsbeat.yml"]
```

# Rsbeat Docker

docker-entrypoint.sh

---

```
#!/bin/sh

if [ ! $ES_URL ];then
    ES_URL="127.0.0.1:9200"
fi
if [ ! $REDIS_LIST ];then
    REDIS_LIST="127.0.0.1:6379","10.0.0.40:6379"
fi
if [ ! $REDIS_SLOWER_THAN ];then
    REDIS_SLOWER_THAN=200
fi
if [ ! $PERIOD ];then
    PERIOD="1s"
fi
# Render config file
cat rsbeat.yml | sed "s|ES_URL|${ES_URL}|g" | sed "s|REDIS_LIST|${REDIS_LIST}|g" | sed
"s|REDIS_SLOWER_THAN|${REDIS_SLOWER_THAN}|g" | sed "s|PERIOD|${PERIOD}|g" >
rsbeat.yml.tmp
cat rsbeat.yml.tmp > rsbeat.yml
rm rsbeat.yml.tmp
```

# Rsbeat Opensource

---

- Readme
- Release binary files
- Contribute to community

<https://github.com/elastic/beats/pull/4108>

<https://www.elastic.co/contributor-agreement>

<https://www.elastic.co/guide/en/beats/libbeat/current/community-beats.html>

- Ship data from any source
- Support for large scale data set
- Opensource community

Q&A