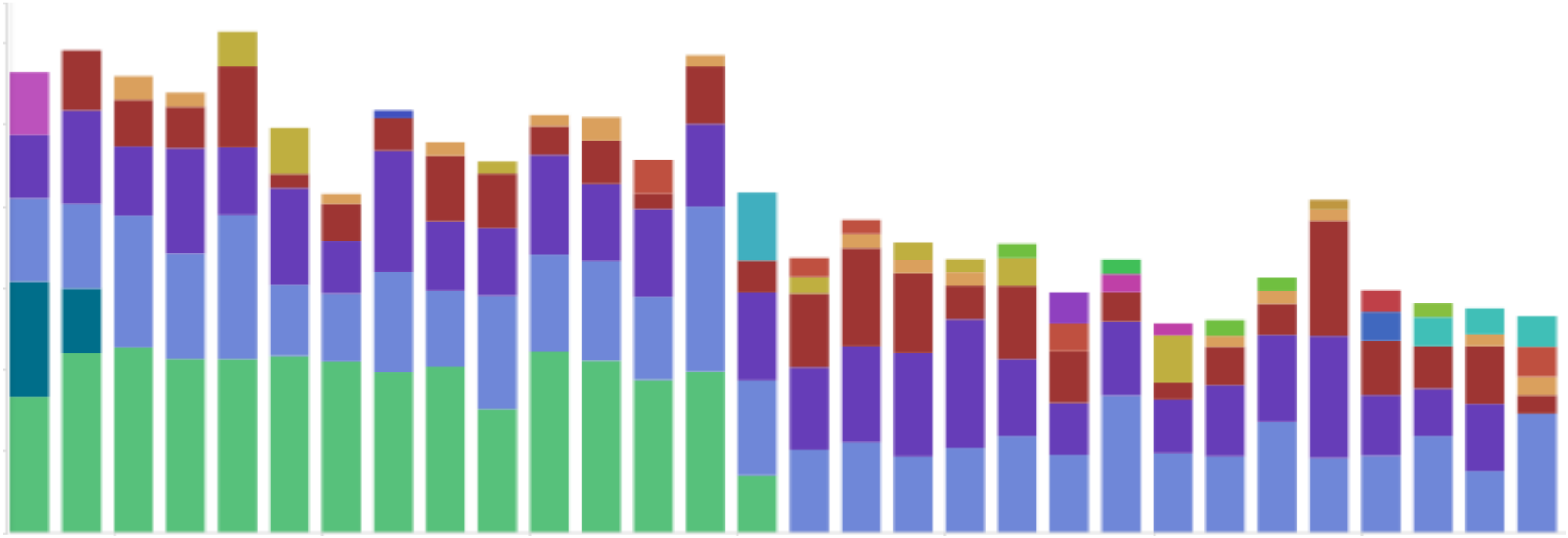


Introduction to ELK stack

– 巨量資料處理、搜尋、及分析工具介紹 –

計資中心網路組 邵喻美

madeline@ntu.edu.tw

Topics

- Why big data tool for network traffic and log analysis
- What is ELK stack, and why choose it
- ELK stack intro
- ELK use cases
- Implementation of ELK on network & account anomaly detection

Network operation and security management issues

- Lots of users
  - Faculty & staff & students → more than 40000 users on campus
- Lots of systems
  - Routers, firewalls, servers….
- Lots of logs
  - Netflow, syslogs, access logs, service logs, audit logs….
- Nobody cares until something go wrong….

Logs & events analysis for network managements

- Logs & events collection from multiple sources
- Accept and parse different log formats
- Large amount, and various formats of data
- Scalable architecture
- Expert knowledge requirement

How we “traditional” system managers treat logs

- Set up one or more log servers for receiving logs from servers/routers/appliances
- Unix commands -- grep + awk + sed + sort + uniq + perl + shell script ….
- Cronjobs executed periodically
  - compute stats and send out report/alert
  - detect possible abnormal behavior and react accordingly
- Plain text reports or stats trends webpage

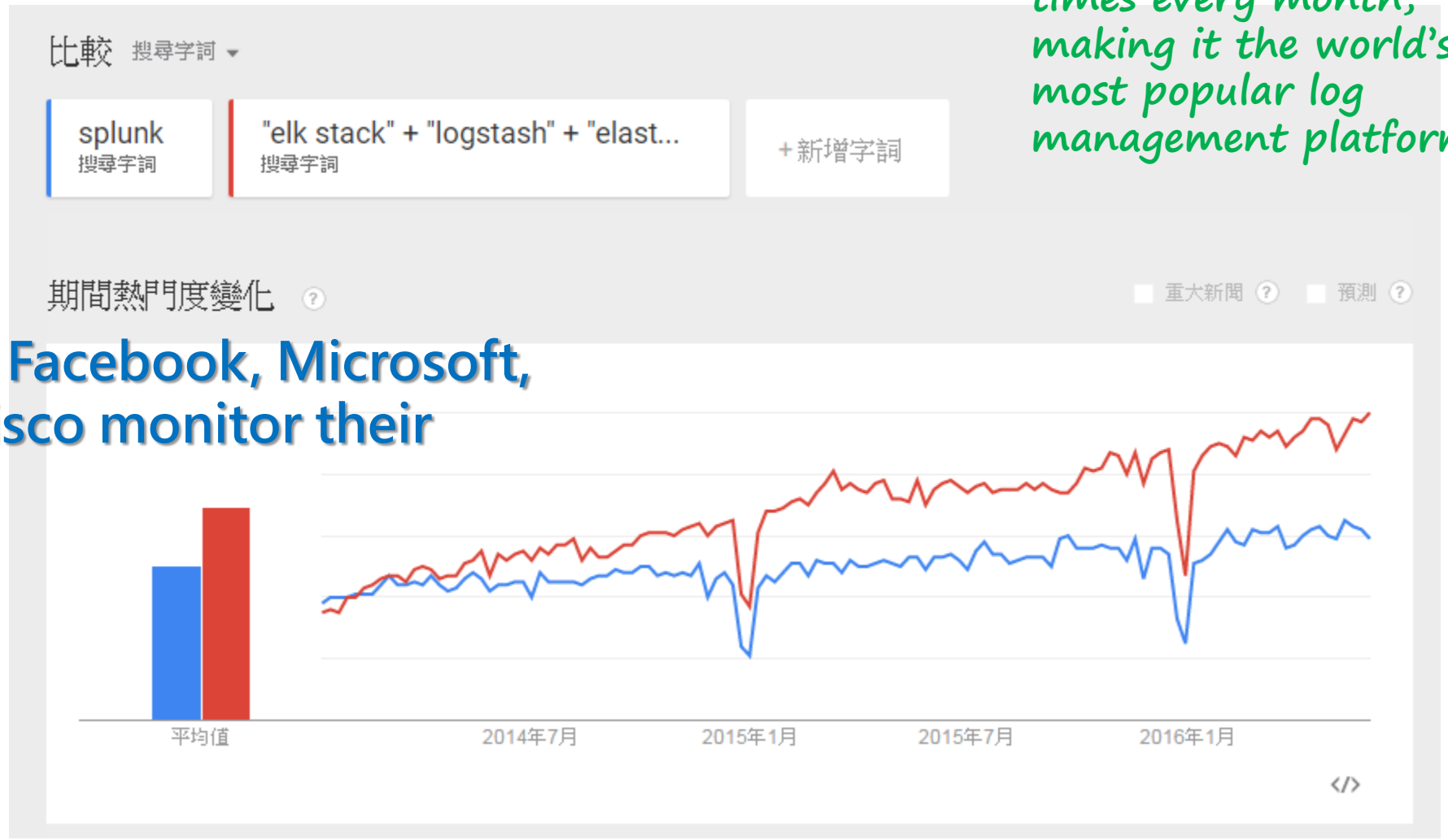
Amount of data….

- Router
  - Netflow – 43GB daily
- Wifi
  - NAT log – 4.8TB daily
  - Auth log
- WAF/Firewall
- Server access logs & events
- Mail server log ~18GBdaily
  - POP3 – avg. 7GB daily
  - SMTP – avg. 1.75GB daily
  - Exchange – avg. 140MB daily
  - OWA – avg. 8.4GB daily
  - MessageTrackingLog – avg. 100MB daily

What is ELK, and why choose it

Splunk vs. ELK on Google Trend

One of the leaders in security information and event management (SIEM) market



The ELK Stack is now downloaded 500,000 times every month, making it the world's most popular log management platform

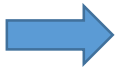
How do Netflix, Facebook, Microsoft, LinkedIn, and Cisco monitor their logs? With ELK.

Why ELK?

- Rapid on-premise (or cloud) installation and easy to deploy
- Scales vertically and horizontally
- Easy and various APIs to use
  - Ease of writing queries, a lot easier then writing a MapReduce job
- Availability of libraries for most programming/scripting languages
  - Elastic offers a host of language clients for Elasticsearch, including Ruby, Python, PHP, Perl, .NET, Java, and Javascript, and more
- Tools availability
- It's free (open source), and it's quick

Logstash

Data From Any Source



Elasticsearch

Instantly Search & Analyze

Kibana

Actionable Insight

Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets

➔ **collects and parses logs**

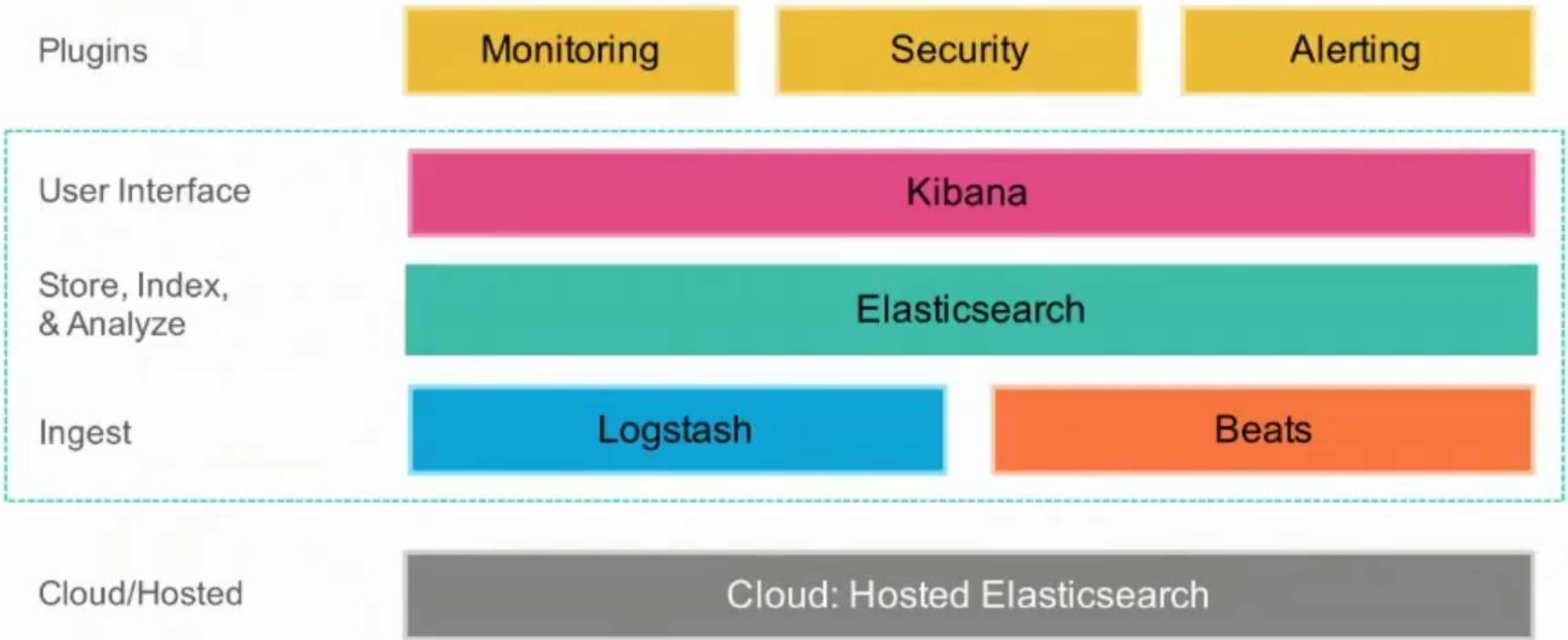
Elasticsearch is a NoSQL database that is based on the Lucene search engine

➔ **indexes and stores the information**

Kibana is a visualization layer that works on top of Elasticsearch

➔ **presents the data in visualizations that provide actionable insights**

The Elastic Stack



ELK modules

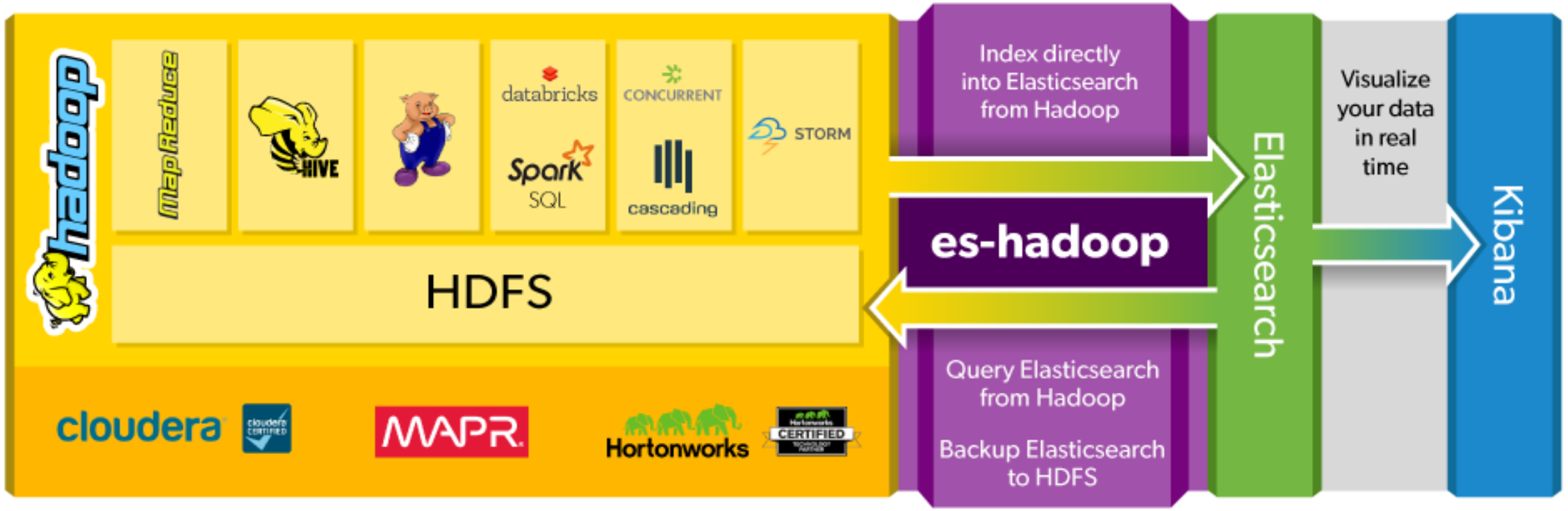
Open Source —

- ElasticSearch
- Logstash
- Kibana
- Beats
  - data shippers – collect, parse & ship

Extension plugins —

- Alerting (Watcher)
  - Proactively monitoring and alerting based on elasticsearch queries or conditions
- Security (Shield)
  - Protect and provide security to elastic stack
- Monitoring (Marvel)
  - Monitor and diagnose health and performance of elastics cluster
- Graph
  - discover and explore the relationships live in data by adding relevance to your exploration

Connect Speedy Search with Big Data Analytics – Elasticsearch for Apache Hadoop

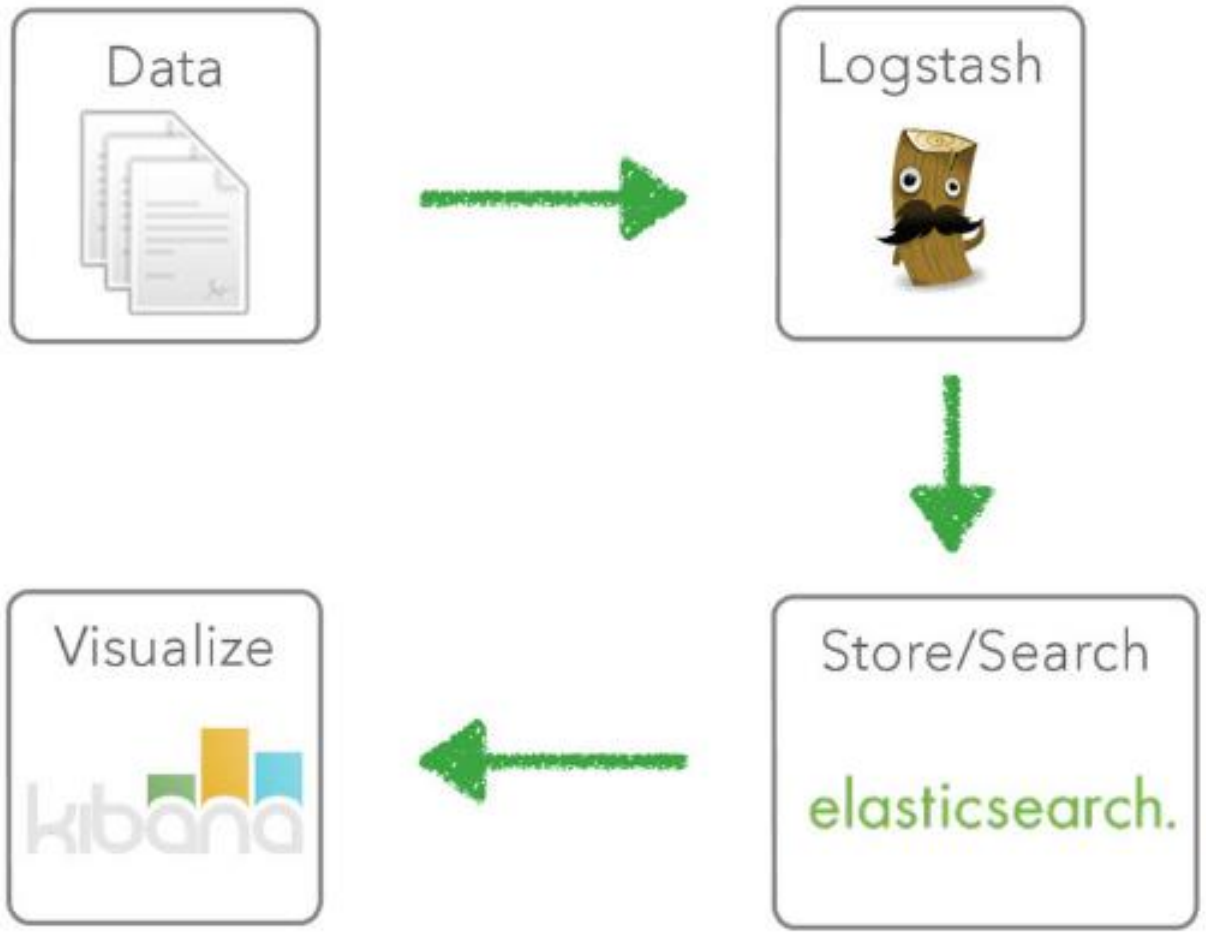


ES-Hadoop -- a two-way connector

- **Read and write data to ES and query it in real time**

let’s look into ELK stack

The ELK stack

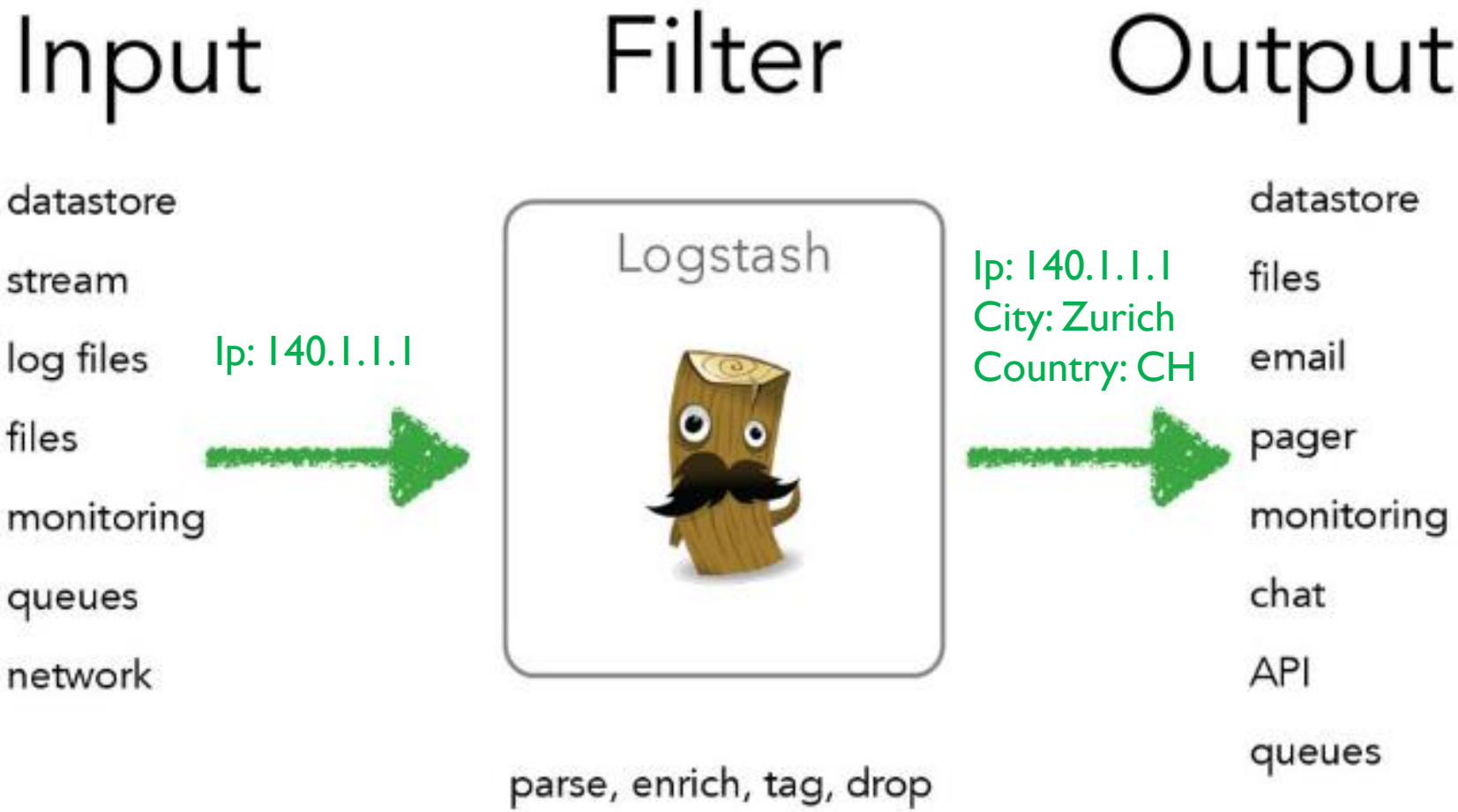


Elasticsearch-**Logstash**-Kibana

Logstash

- Managing events and logs
- Collect data
- Parse data
- Enrich data
- Store data
- Open Source: Apache License 2.0

Logstash architecture



How logstash works

```
input {
  file {
    path => "/tmp/access_log"
    start_position => "beginning"
  }
}
```

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

```
output {
    elasticsearch {
    }
}
```

Logstash Input plugins

- Stdin – Reads events from standard input
- File – Streams events from files (similar to “tail -0F”)
- Syslog – Reads syslog messages as events
- Eventlog – Pulls events from the Windows Event Log
- Imap – read mail from an IMAP server
- Rss – captures the output of command line tools as an event
- Snmptrap – creates events based on SNMP trap messages
- Twitter – Reads events from the Twitter Streaming API
- Irc – reads events from an IRC server
- Exec – Captures the output of a shell command as an event
- Elasticsearch – Reads query results from an Elasticsearch cluster
- ….

Logstash Filter plugins

- grok – parses unstructured event data into fields
- Mutate – performs mutations on fields
- Geoip – adds geographical information about an IP address
- Date – parse dates from fields to use as the Logstash timestamp for an event
- Cidr – checks IP addresses against a list of network blocks
- Drop – drops all events
- …

Logstash Output plugins

- Stdout – prints events to the standard output
- Csv – write events to disk in a delimited format
- Email – sends email to a specified address when output is received
- Elasticsearch – stores logs in Elasticsearch
- Exec – runs a command for a matching event
- File – writes events to files on disk
- mongoDB – writes events to MongoDB
- Redmine – creates tickets using the Redmine API
- ….

```
Dec 23 14:30:01 louis CRON[619]: (www-data) CMD (php /usr/share/cacti/site/poller.php
>/dev/null 2>/var/log/cacti/poller-error.log)
```

```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
%{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    date {
      match => [ "syslog_timestamp", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

```
{
       "message" => "Dec 23 14:30:01 louis CRON[619]: (www-data) CMD (php
/usr/share/cacti/site/poller.php >/dev/null 2>/var/log/cacti/poller-error.log)",
       "@timestamp" => "2013-12-23T22:30:01.000Z",
       "@version" => "1",
       "type" => "syslog",
       "host" => "0:0:0:0:0:0:0:1:52617",
       "syslog_timestamp" => "Dec 23 14:30:01",
       "syslog_hostname" => "louis",
       "syslog_program" => "CRON",
       "syslog_pid" => "619",
       "syslog_message" => "(www-data) CMD (php /usr/share/cacti/site/poller.php
>/dev/null 2>/var/log/cacti/poller-error.log)",
       "received_at" => "2013-12-23 22:49:22 UTC",
       "received_from" => "0:0:0:0:0:0:0:1:52617",
       "syslog_severity_code" => 5,
       "syslog_facility_code" => 1,
       "syslog_facility" => "user-level",
       "syslog_severity" => "notice"
}
```

Example: Web server log files

```
       "message" => "83.149.9.216 - - [28/May/2014:16:13:42 -0500] \"GET /
presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1\"
200 203023 \"http://semicomplete.com/presentations/logstash-
monitorama-2013/\" \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/
537.36\"",

       "@version" => "1",
     "@timestamp" => "2014-05-28T21:13:42.000Z",
          "host" => "kryptic.local",
       "clientip" => "83.149.9.216",
         "ident" => "-",
          "auth" => "-",
     "timestamp" => "28/May/2014:16:13:42 -0500",
          "verb" => "GET",
       "request" => "/presentations/logstash-monitorama-2013/images/
kibana-search.png",
   "httpversion" => "1.1",
      "response" => "200",
         "bytes" => "203023",
      "referrer" => "\"http://semicomplete.com/presentations/logstash-
monitorama-2013/\"",
         "agent" => "\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/
537.36\""
```

grok

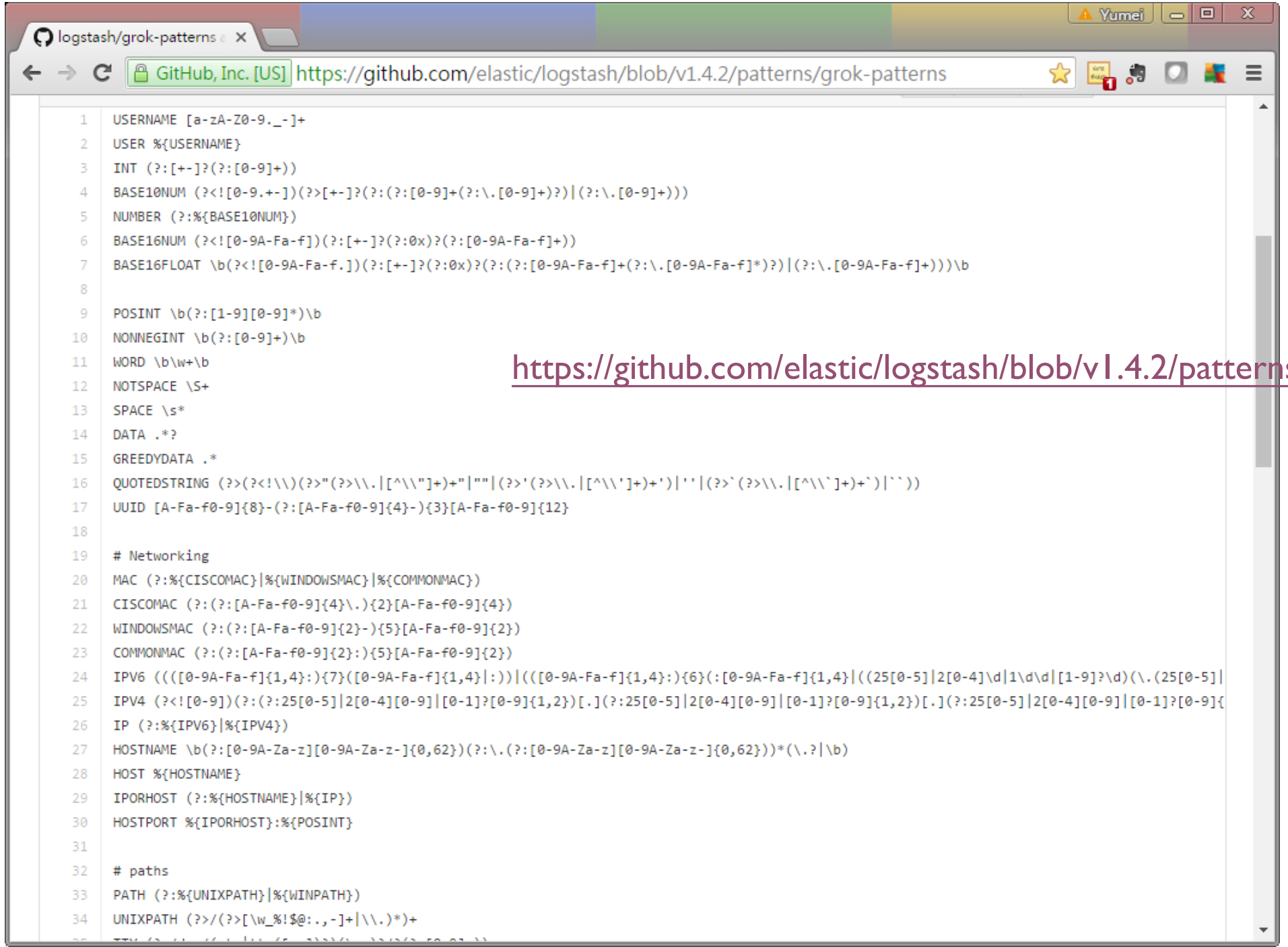
date

Example: Web server log files

```
    "geoip" => {
                  "ip" => "83.149.9.216",
       "country_code2" => "RU",
       "country_code3" => "RUS",
        "country_name" => "Russian Federation",
      "continent_code" => "EU",
         "region_name" => "48",
           "city_name" => "Moscow",
            "latitude" => 55.75219999999999,
           "longitude" => 37.6156,
            "timezone" => "Europe/Moscow",
    "real_region_name" => "Moscow City",
            "location" => [
        [0] 37.6156,
        [1] 55.75219999999999
    ]
},
  "useragent" => {
         "name" => "Chrome",
           "os" => "Mac OS X 10.9.1",
     "os_name" => "Mac OS X",
    "os_major" => "10",
    "os_minor" => "9",
      "device" => "Other",
       "major" => "32",
       "minor" => "0",
       "patch" => "1700"
}
```

geoip

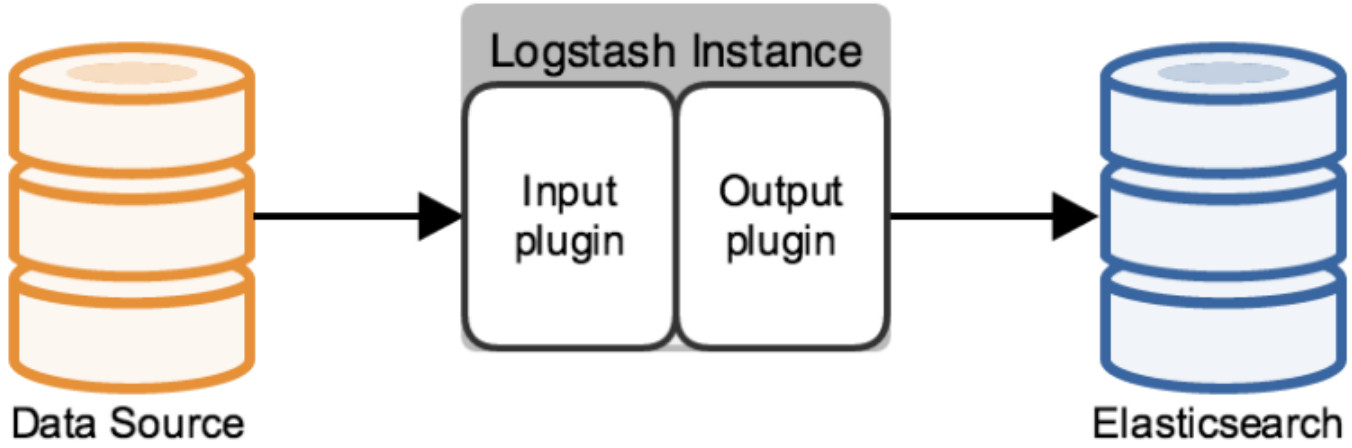
useragent

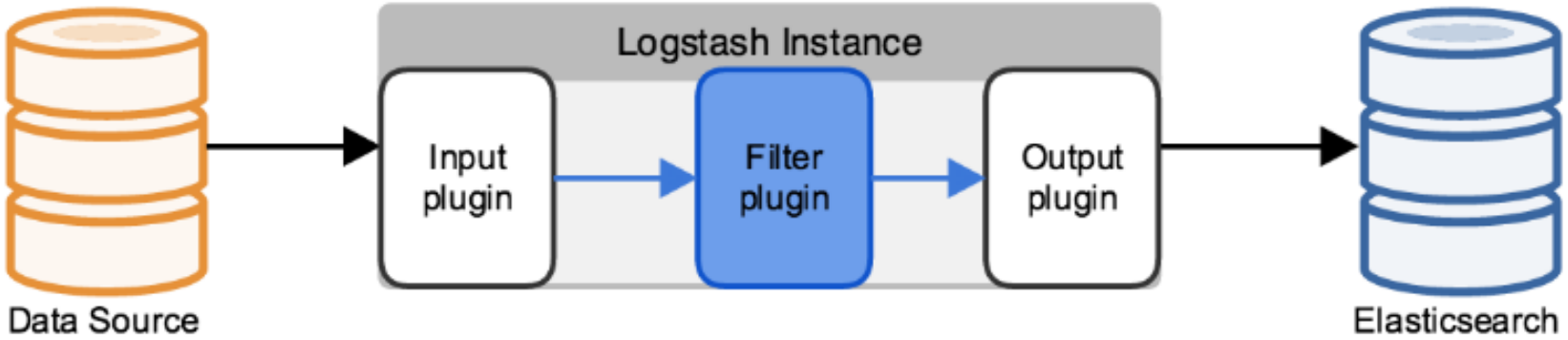


https://github.com/elastic/logstash/blob/v1.4.2/patterns/grok-patterns

Deploying and scaling Logstash

Minimal installation

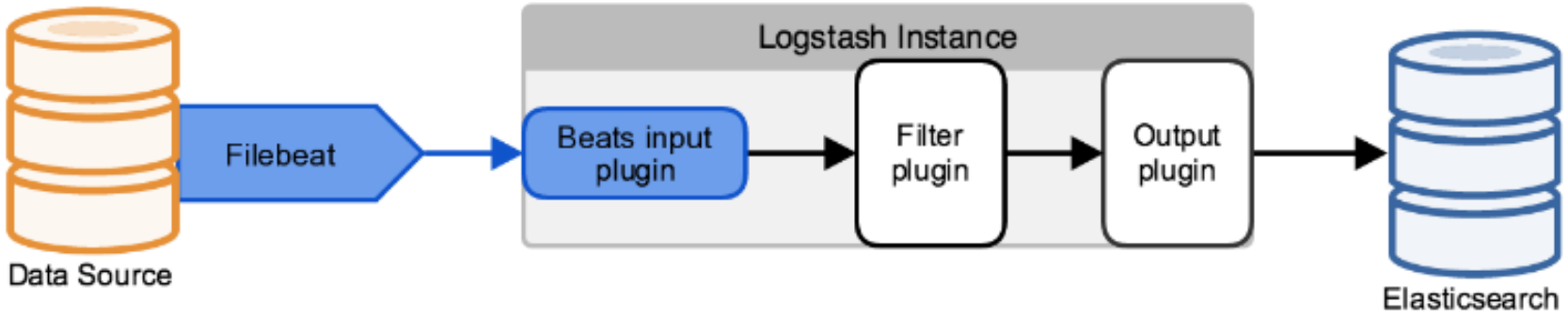


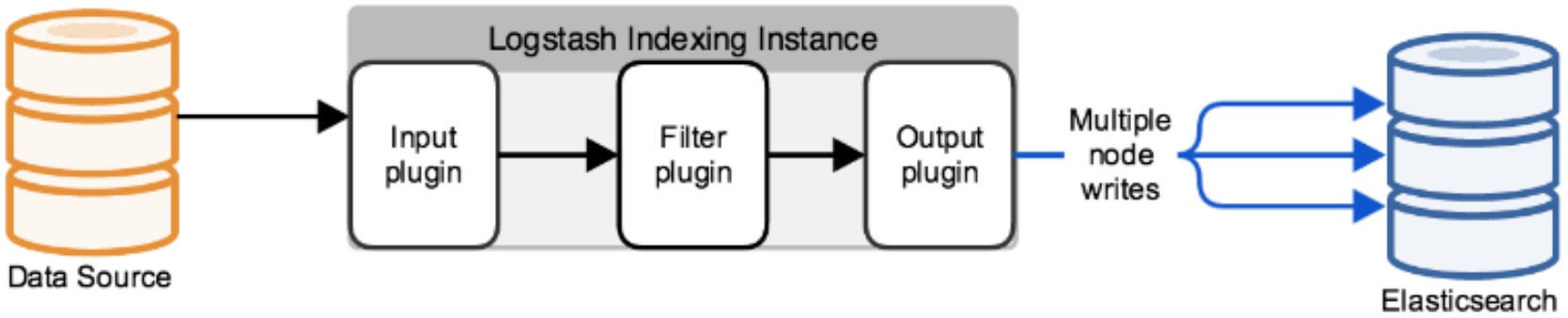


Using Filters

Deploying and scaling Logstash

Using log shipper to minimize the resource demands on Logstash

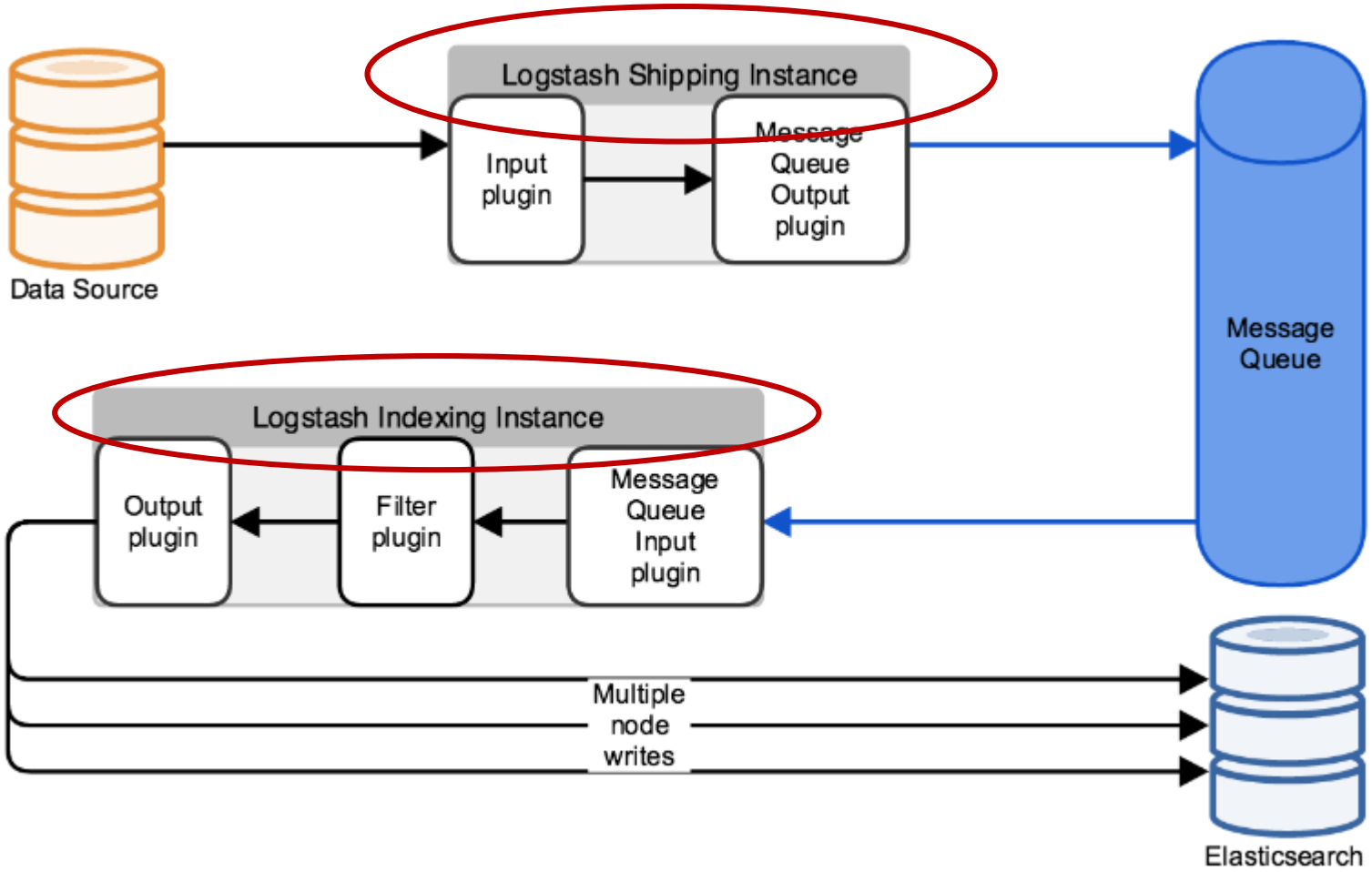




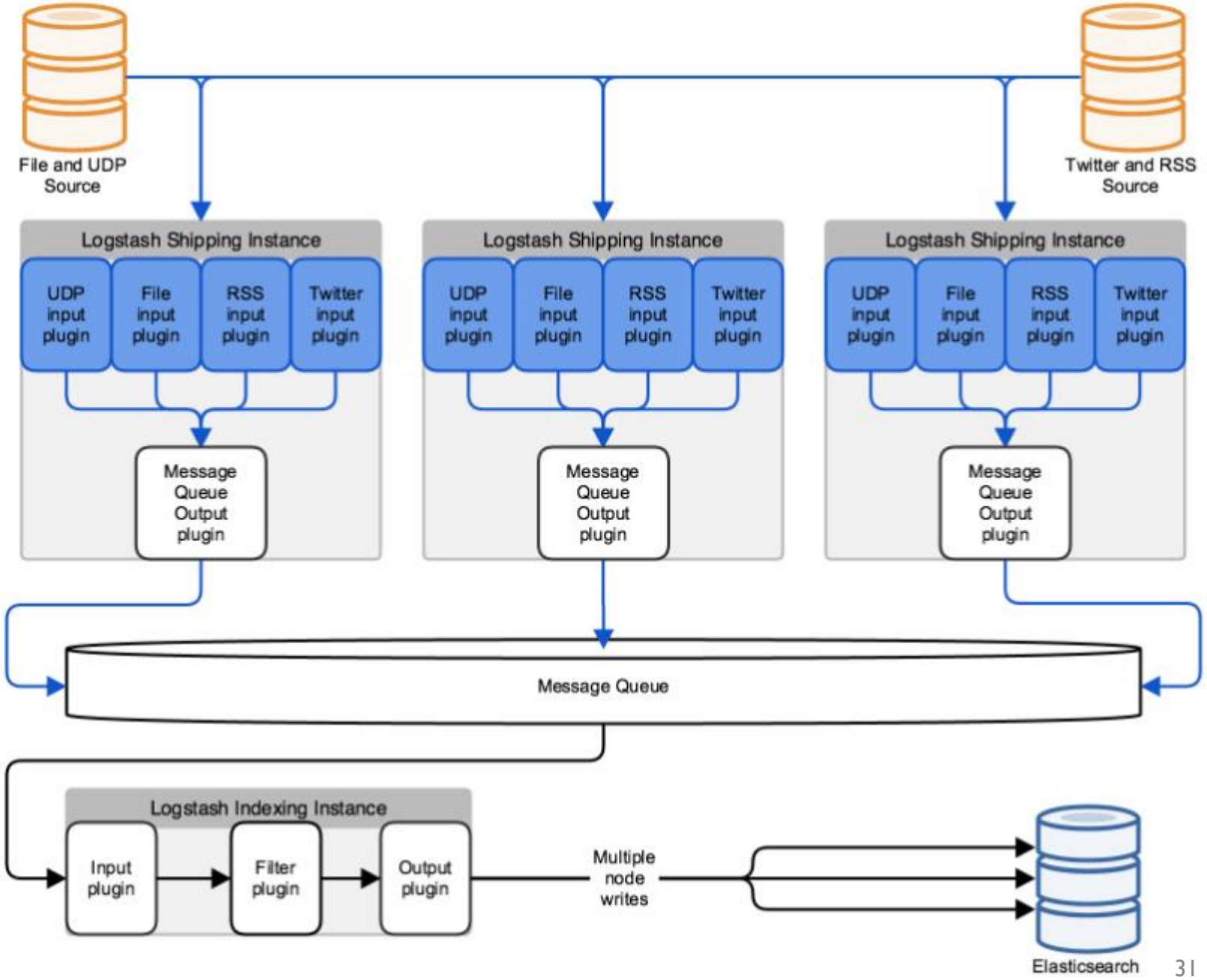
Scaling to a Larger Elasticsearch Cluster

Deploying and scaling Logstash

Managing Throughput Spikes with Message Queuing



Multiple Connections for Logstash High Availability



Elasticsearch-Logstash-Kibana

ElasticSearch

Schema-flexible

- Built on top Apache Lucene™, a full-text search-engine library
- A Schema-free, REST & JSON based distributed search engine with real-time analytics
- Capable of scaling to hundreds of servers and petabytes of structured and unstructured data
- Open Source: Apache License 2.0

Real scalability comes from horizontal scale

- Wikipedia uses Elasticsearch to provide full-text search with highlighted search snippets, and *search-as-you-type* and *did-you-mean* suggestions
- *The Guardian* uses Elasticsearch to combine visitor logs with social-network data to provide real-time feedback to its editors about the public's response to new articles
- Stack Overflow combines full-text search with geolocation queries and uses *more-like-this* to find related questions and answers
- GitHub uses Elasticsearch to query 130 billion lines of code

Elasticsearch vs. Relational DB

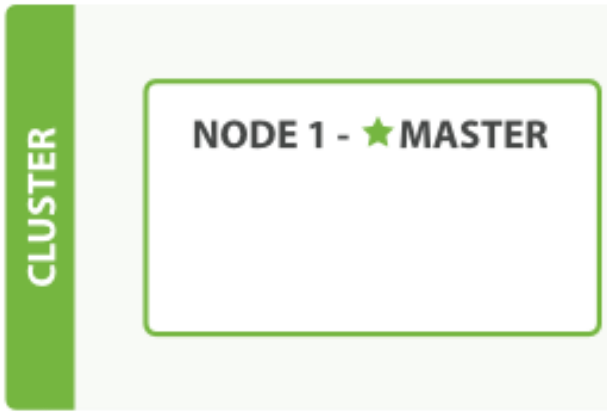
ElasticSearch	Relational DB
Index	Database
Type	Table
Documer	Row
Field	Column
Shard	Partition
Mapping	Schema
- (everything is indexed)	Index
Query DSL (*domain specific language*)	SQL

Shards are how Elasticsearch distributes data around your cluster

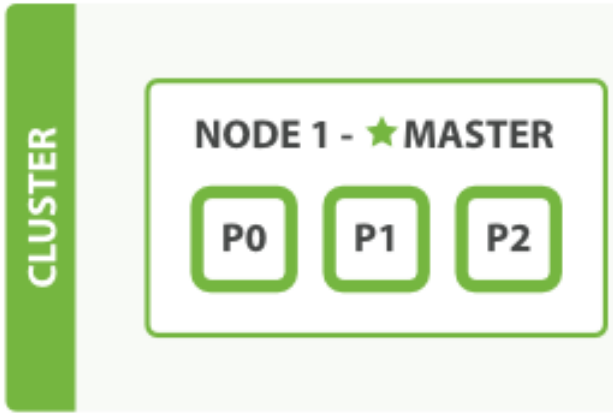
What is a shard

- a shard is a single instance of Lucene, and is a complete search engine in its own right
- Documents are stored and indexed in shards → shards are allocated to nodes in your cluster
- As your cluster grows or shrinks, Elasticsearch will automatically migrate shards between nodes so that the cluster remains balanced
- A shard can be either a *primary* shard or a *replica* shard
  - Each document in your index belongs to a single primary shard
  - A replica shard is just a copy of a primary shard

ElasticSearch clustering – single node cluster

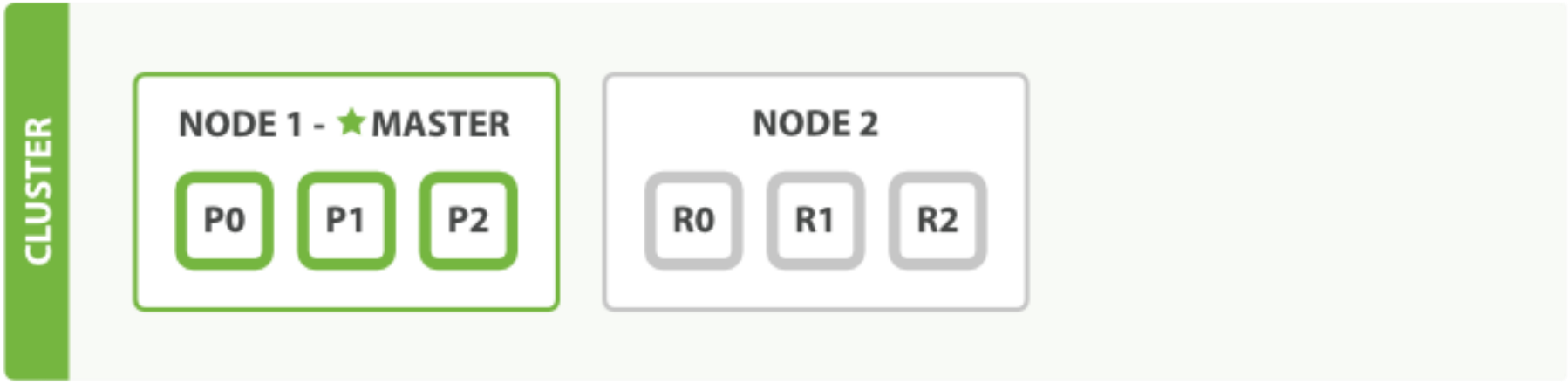


- **Node** = running instance of ES
- **Cluster** = 1+ nodes with the same cluster.name



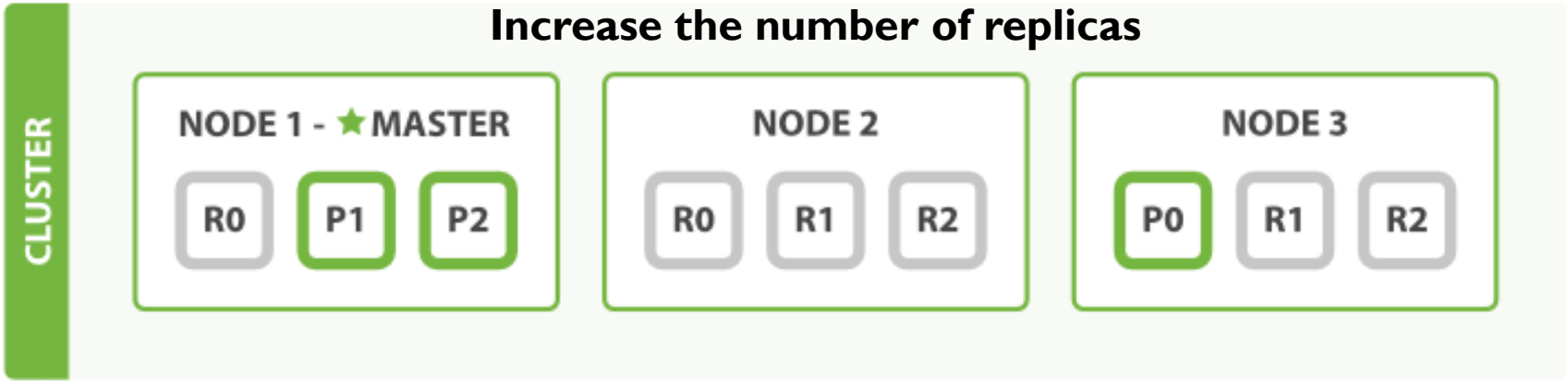
- Every cluster has **1 master node**
- 1 Cluster can have any number of indexes

ElasticSearch clustering – adding a second node



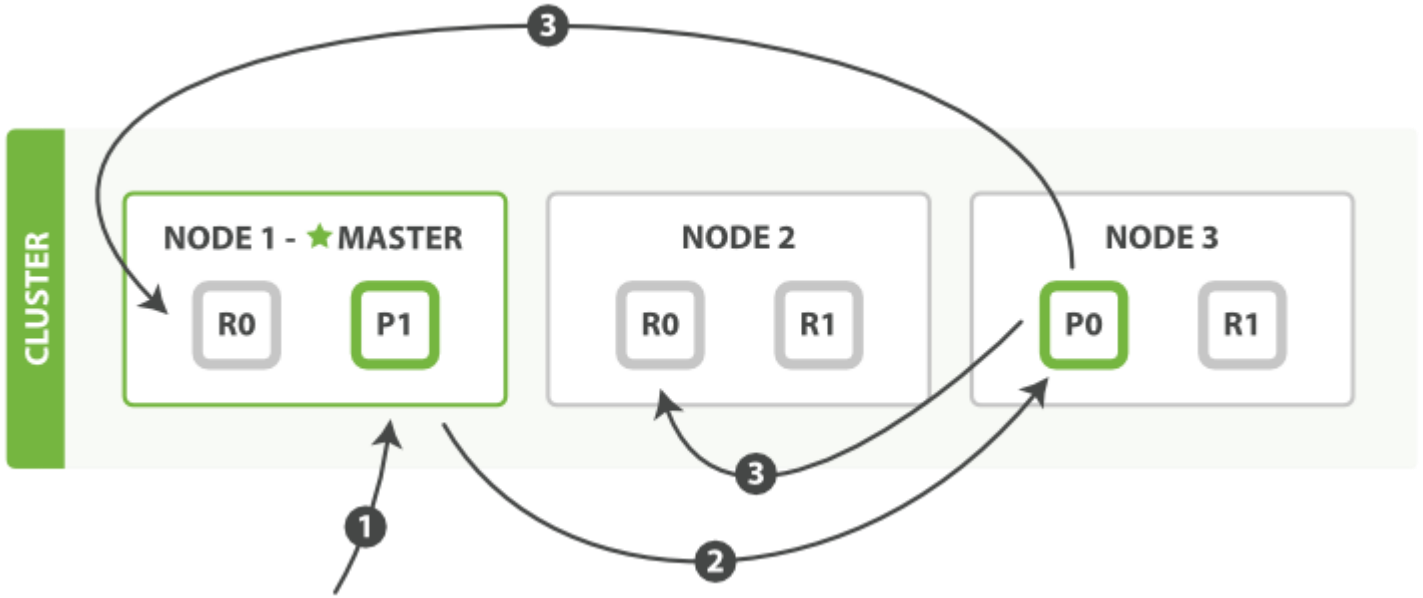
- ➢A cluster consists of one or more nodes with the same cluster.name
- All primary and replica shards are allocated
- Each index has one primary (P) and one replica (R) shard
- Clients talk to any node in the cluster

ElasticSearch clustering – adding a third node



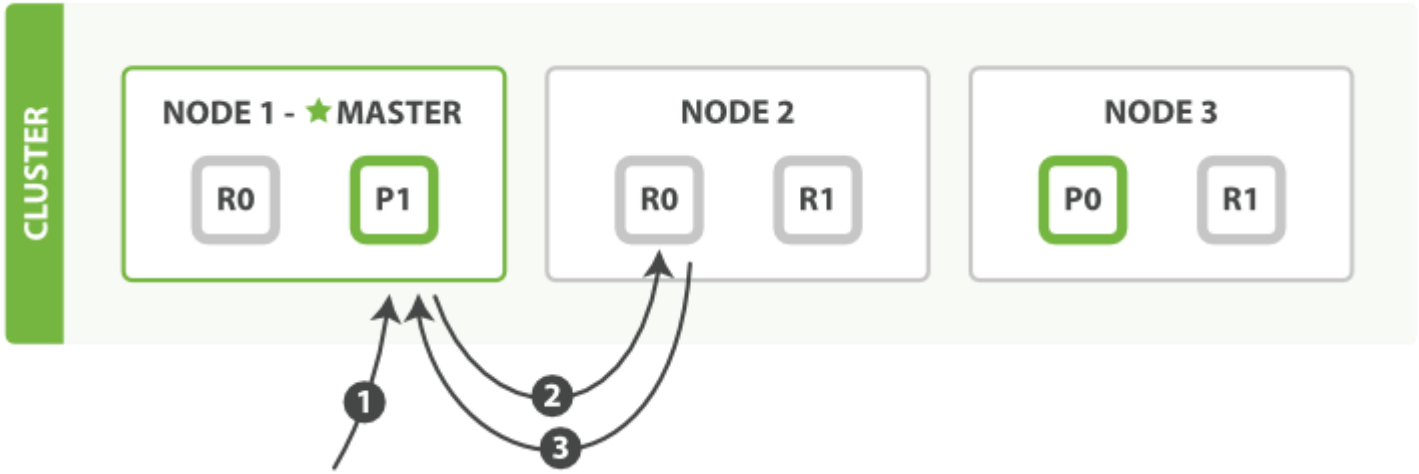
- More primary shards:
  - faster indexing
  - more scale
- More replicas:
  - faster searching
  - more failover

Creating, Indexing, and Deleting a document



1. The client sends a create, index, or delete request to Node 1
2. The node uses the document's _id to determine that the document belongs to shard 0. It forwards the request to Node 3, where the primary copy of shard 0 is currently allocated
3. Node 3 executes the request on the primary shard. If it is successful, it forwards the request in parallel to the replica shards on Node 1 and Node 2. Once all of the replica shards report success, Node 3 reports success to the coordinating node, which reports success to the client.

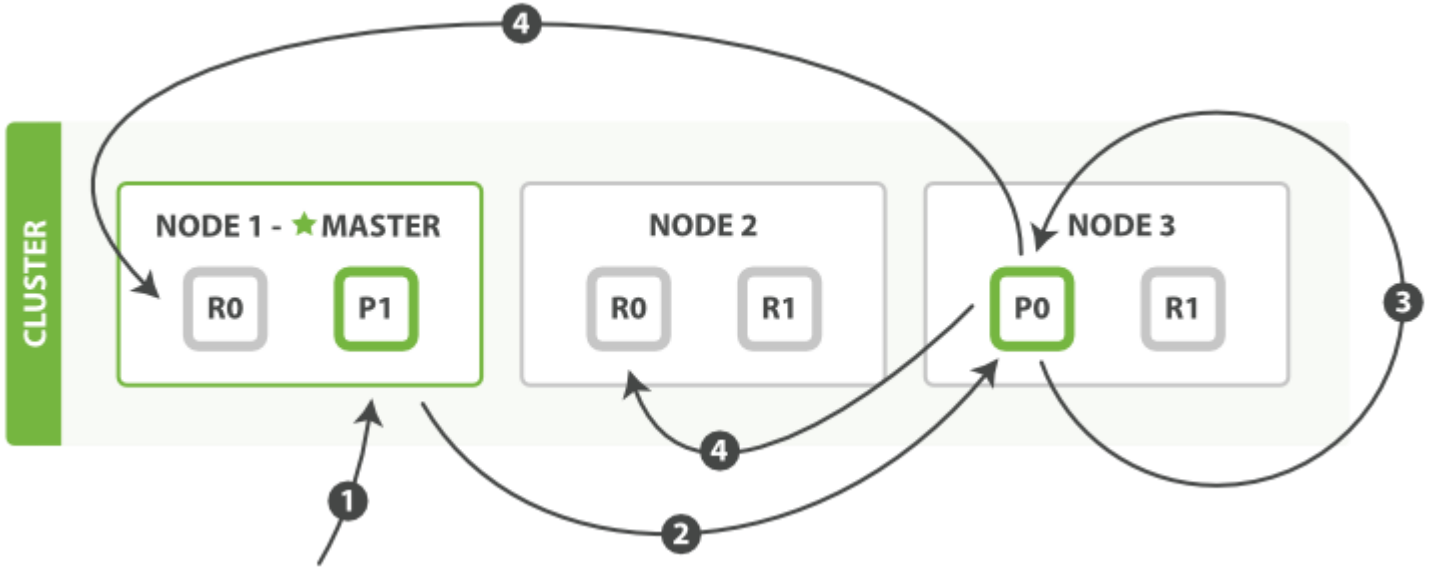
Retrieving a Document



For read requests, the coordinating node will choose a different shard copy on every request in order to balance the load

1. The client sends a get request to node 1
2. The node uses the document's _id to determine that the document belongs to shard 0. Copies of shard 0 exist on all three nodes. On this occasion, it forwards the request to node 2.
3. Node 2 returns the document to node 1, which returns the document to the client.

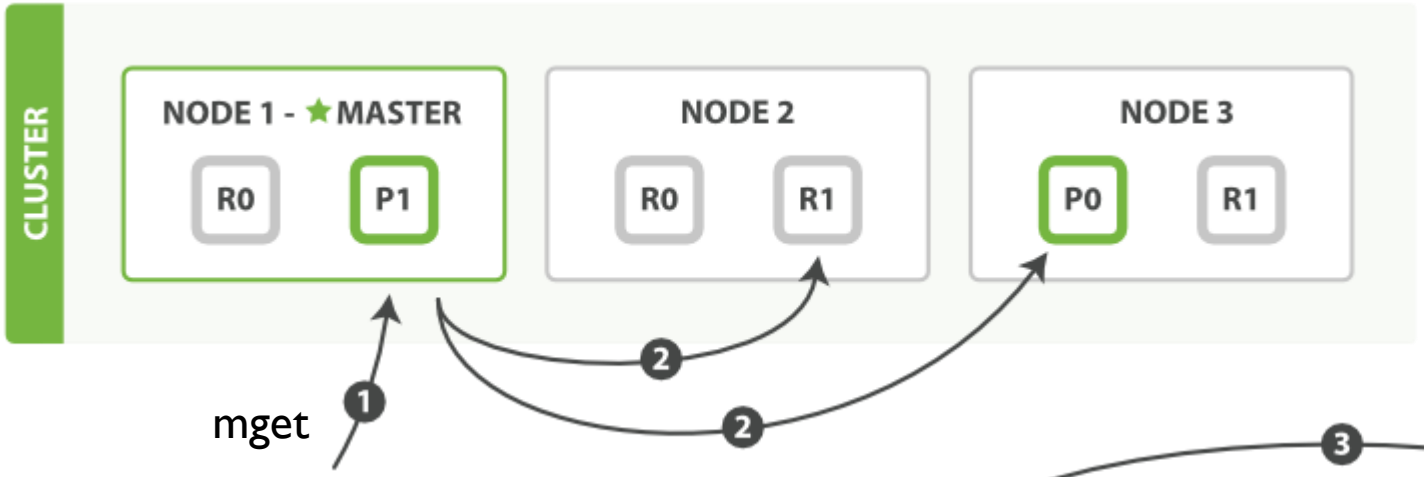
Partial update to a document

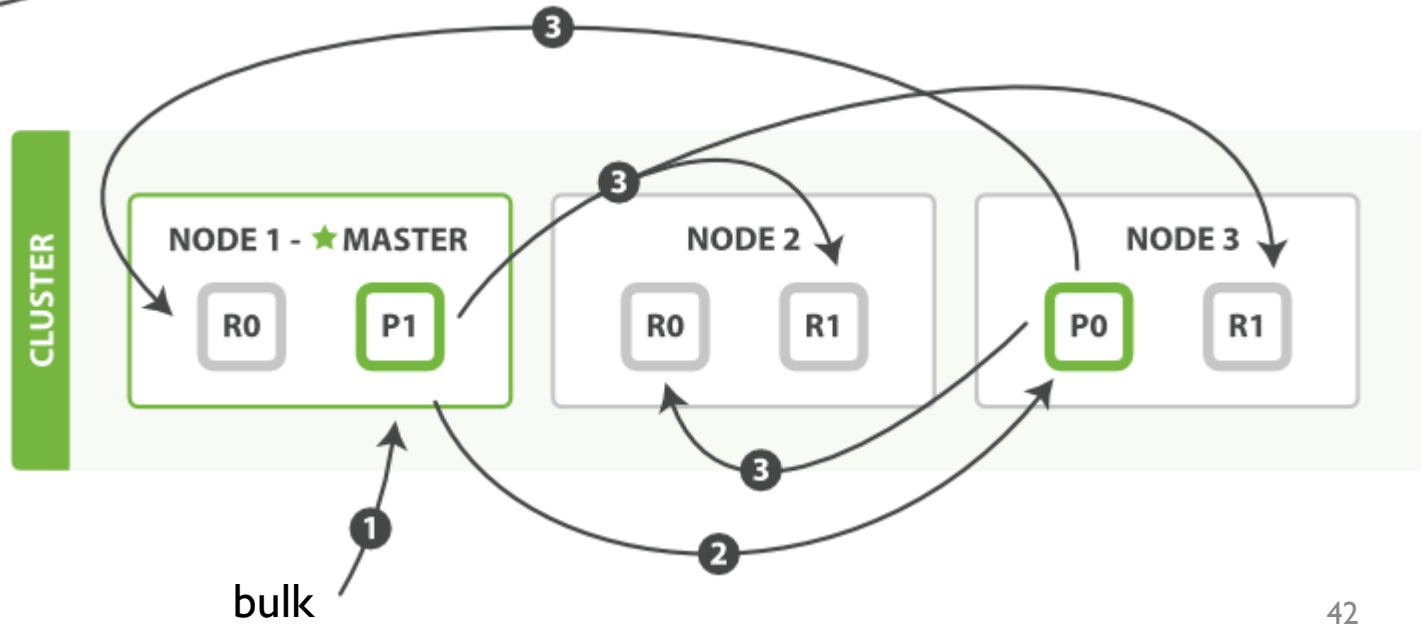


When a primary shard forwards changes to its replica shards, it doesn't forward the update request. Instead it forwards the new version of the full document.

Multidocument Patterns

- the coordinating node knows in which shard each document lives.
- It breaks up the multidocument request into a multidocument request ***per shard***, and forwards these in parallel to each participating node
- Once it receives answers from each node, it collates their responses into a single response





Talking to Elasticsearch

HTTP method or verb: GET, POST, PUT, HEAD, or DELETE

- RESTful API with JSON over HTTP
  - Over port 9200
  - Access via web client, or command line by `curl` command

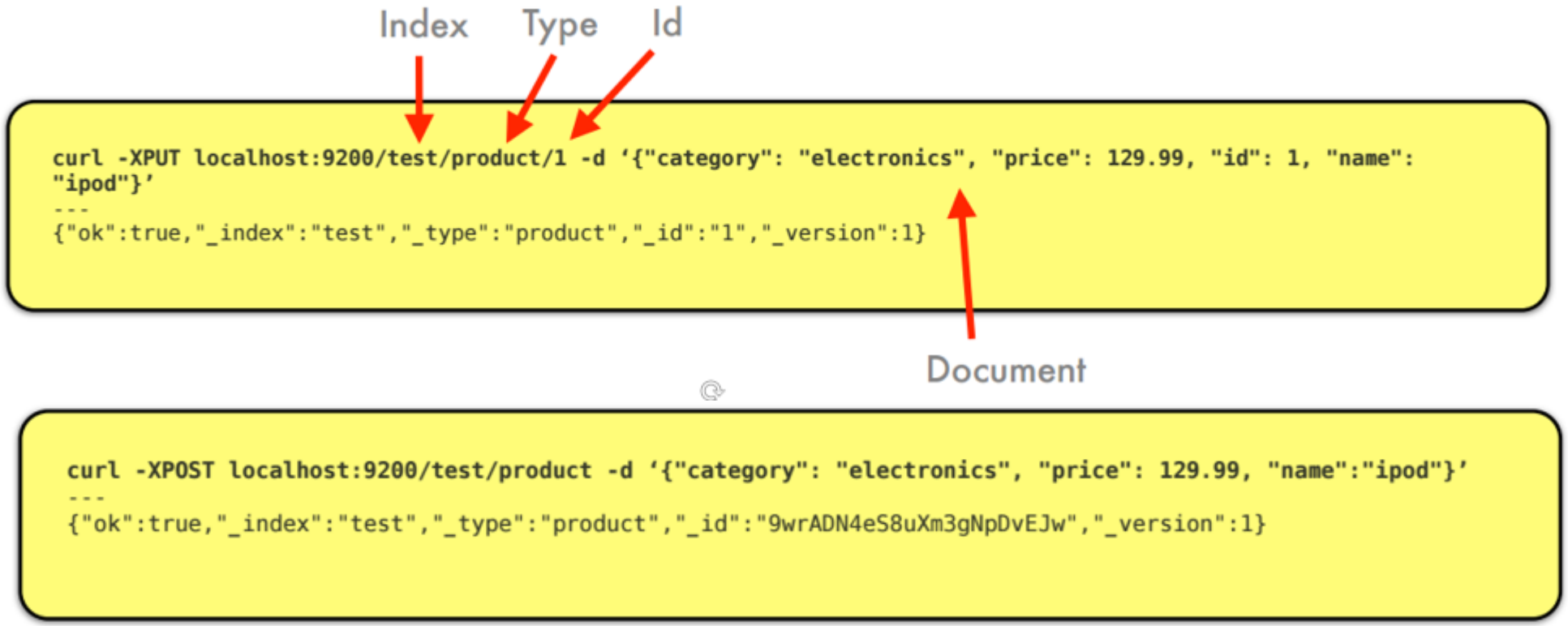
```
curl -X<VERB> '<PROTOCOL>://<HOST>:<PORT>/<PATH>?<QUERY_STRING>' -d '<BODY>'
```

  - JSON (JavaScript Object Notation) ← the standard format used by NoSQL

```
curl -XGET 'http://localhost:9200/_count?pretty' -d '
{
    "query": {
        "match_all": {}
    }
}
```

- Elasticsearch clients
  - Java API, Java REST client, JavaScript API, PHP API, Python API, Perl API…

Indexing a document



- Store a document in an index so that it can be retrieved and queried
- Like the INSERT keyword in SQL

Retrieving documents

```
curl -XGET 'localhost:9200/test/product/1?pretty'
---
{
  "_index" : "test",
  "_type" : "product",
  "_id" : "1",
  "_version" : 2,
  "exists" : true, "_source" : {"category": "electronics", "price":
129.99, "name": "ipod"}
}
```

- Using GET method to retrieve document
- We can retrieve a specific document if we happen to know its id

Performing Queries

- Using the q=<query> form performs a full-text search by parsing the query string value

```
curl -XGET 'localhost:9200/test/product/_search?
q="ipod"&format=yaml'
---
took: 104
timed_out: false
_shards:
  total: 1
  successful: 1
  failed: 0
hits:
  total: 1
  max_score: 0.15342641
  hits:
  - _index: "test"
    _type: "product"
    _id: "1"
    _score: 0.15342641
    _source:
      category: "electronics"
      price: 129.99
      name: "ipod"
```

- Query with ***query DSL***, which is specified using a JSON request body

```
GET /megacorp/employee/_search
{
    "query" : {
        "match" : {
            "last_name" : "Smith"
        }
    }
}
```

Query DSL – Combining Filters

```
SELECT product
FROM   products
WHERE  (price = 20 OR productID = "XHDK-A-1293-#fJ3")
  AND  (price != 30)
```

```
GET /my_store/products/_search
{
   "query" : {
      "constant_score" : { ❶
         "filter" : {
            "bool" : {
              "should" : [
                 { "term" : {"price" : 20}}, ❷
                 { "term" : {"productID" : "XHDK-A-1293-#fJ3"}} ❸
              ],
              "must_not" : {
                 "term" : {"price" : 30} ❹
              }
            }
         }
      }
   }
}
```

Bool Filter

```
{
   "bool" : {
      "must" :     [],
      "should" :   [],
      "must_not" : [],
      "filter":    []
   }
}
```

Query DSL – Nesting Boolean Queries

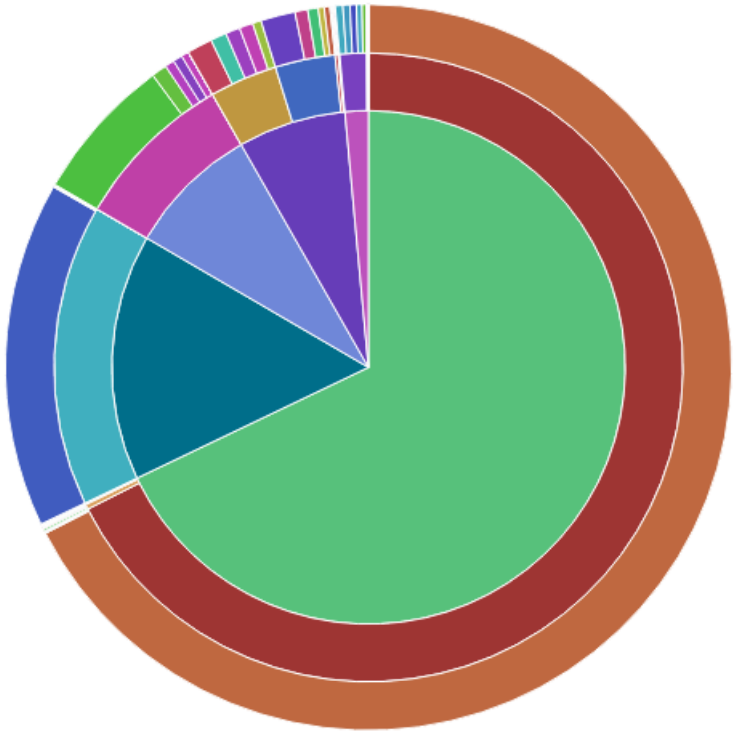
```
SELECT document
FROM   products
WHERE  productID      = "KDKE-B-9947-#kL5"
  OR (     productID = "JODL-X-1937-#pV7"
       AND price     = 30 )
```

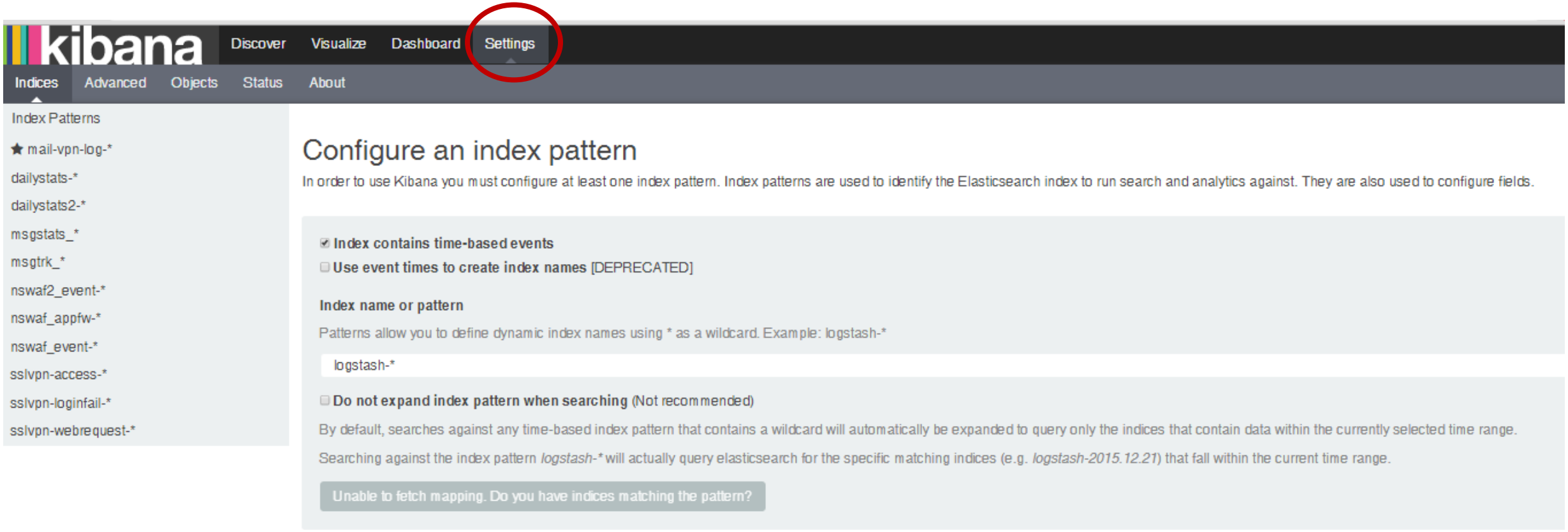
```
GET /my_store/products/_search
{
   "query" : {
      "constant_score" : {
         "filter" : {
            "bool" : {
              "should" : [
                { "term" : {"productID" : "KDKE-B-9947-#kL5"}}, ❶
                { "bool" : { ❷
                  "must" : [
                    { "term" : {"productID" : "JODL-X-1937-#pV7"}}, ❸
                    { "term" : {"price" : 30}} ❹
                  ]
                }}
              ]
            }
         }
      }
   }
}
```

Elasticsearch-Logstash-Kibana

Kibana

- Search, view, and interact with data stored in Elasticsearch indices
- Execute queries on data & visualize results in charts, tables, and maps
- Add/remove widgets
- Share/Save/Load dashboards
- Open Source: Apache License 2.0




kibana
Discover
Visualize
Dashboard
Settings
Indices
Advanced
Objects
Status
About
Index Patterns
mail-vpn-log-*
dailystats-*
dailystats2-*
msgstats_*
msgtrk_*
nswaf2_event-*
nswaf_appfw-*
nswaf_event-*
sslvpn-access-*
sslvpn-loginfail-*
sslvpn-webrequest-*
Configure an index pattern
In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.
Index contains time-based events
Use event times to create index names [DEPRECATED]
Index name or pattern
Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*
logstash-*
Do not expand index pattern when searching (Not recommended)
By default, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.
Searching against the index pattern logstash-* will actually query elasticsearch for the specific matching indices (e.g. logstash-2015.12.21) that fall within the current time range.
Unable to fetch mapping. Do you have indices matching the pattern?

Create a new visualization

Step 1

Area chart	Great for stacked timelines in which the total of all series is more important than comparing any two or more series. Less useful for assessing the relative change of unrelated data points as changes in a series lower down the stack will have a difficult to gauge effect on the series above it.
Data table	The data table provides a detailed breakdown, in tabular format, of the results of a composed aggregation. Tip, a data table is available from many other charts by clicking grey bar at the bottom of the chart.
Line chart	Often the best chart for high density time series. Great for comparing one series to another. Be careful with sparse sets as the connection between points can be misleading.
Markdown widget	Useful for displaying explanations or instructions for dashboards.
Metric	One big number for all of your one big number needs. Perfect for showing a count of hits, or the exact average a numeric field.
Pie chart	Pie charts are ideal for displaying the parts of some whole. For example, sales percentages by department.Pro Tip: Pie charts are best used sparingly, and with no more than 7 slices per pie.
Tile map	Your source for geographic maps. Requires an elasticsearch geo_point field. More specifically, a field that is mapped as type:geo_point with latitude and longitude coordinates.
Timeseries	Create timeseries charts using the timelion expression language. Perfect for computing and combining timeseries set with functions suchs as derivatives and moving averages
Vertical bar chart	The goto chart for oh-so-many needs. Great for time and non-time data. Stacked or grouped, exact numbers or percentages. If you are not sure which chart you need, you could do worse than to start here.

ELK use cases

Use Cases

Search

Logging

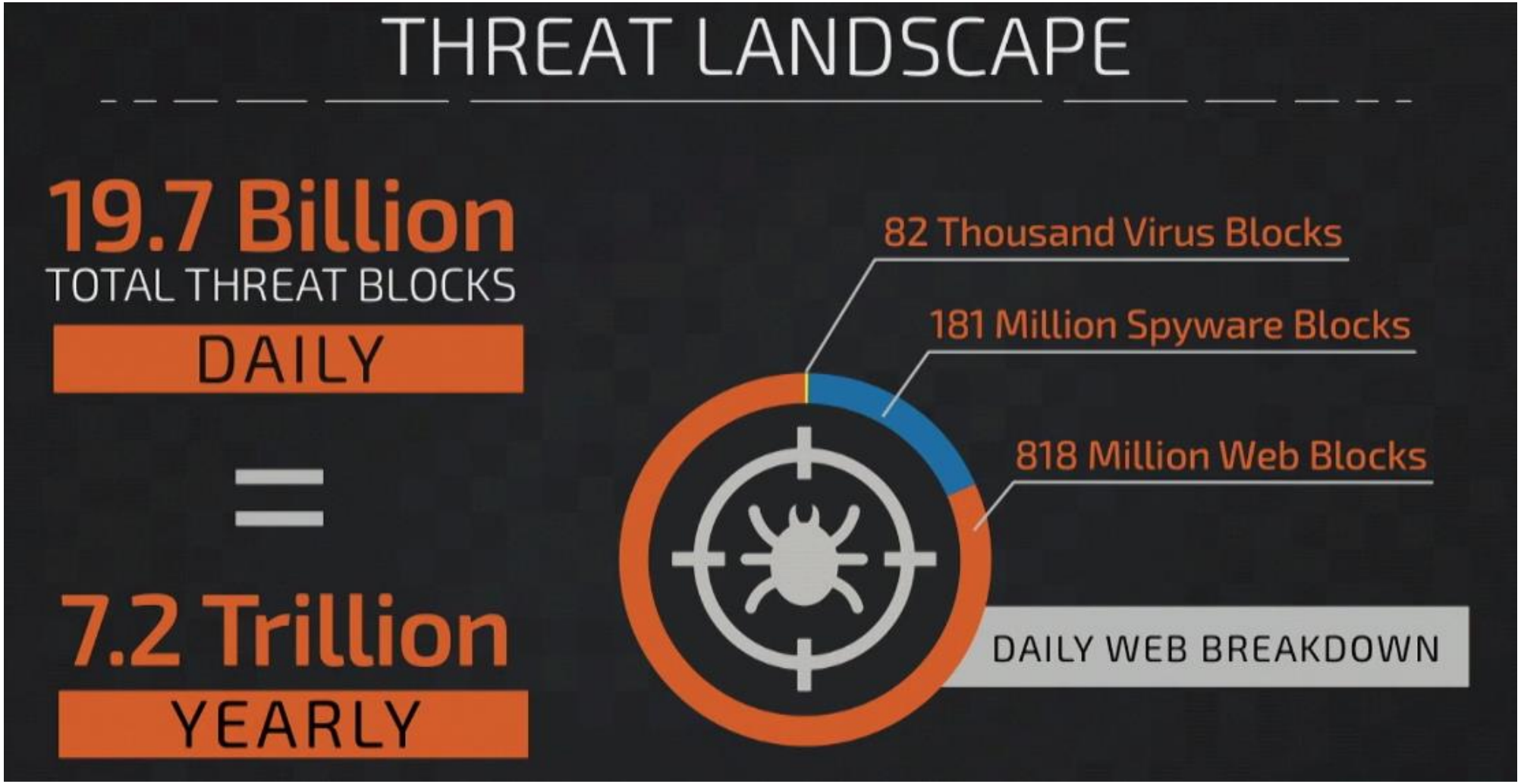
Security
Analytics

Analytics

User cases

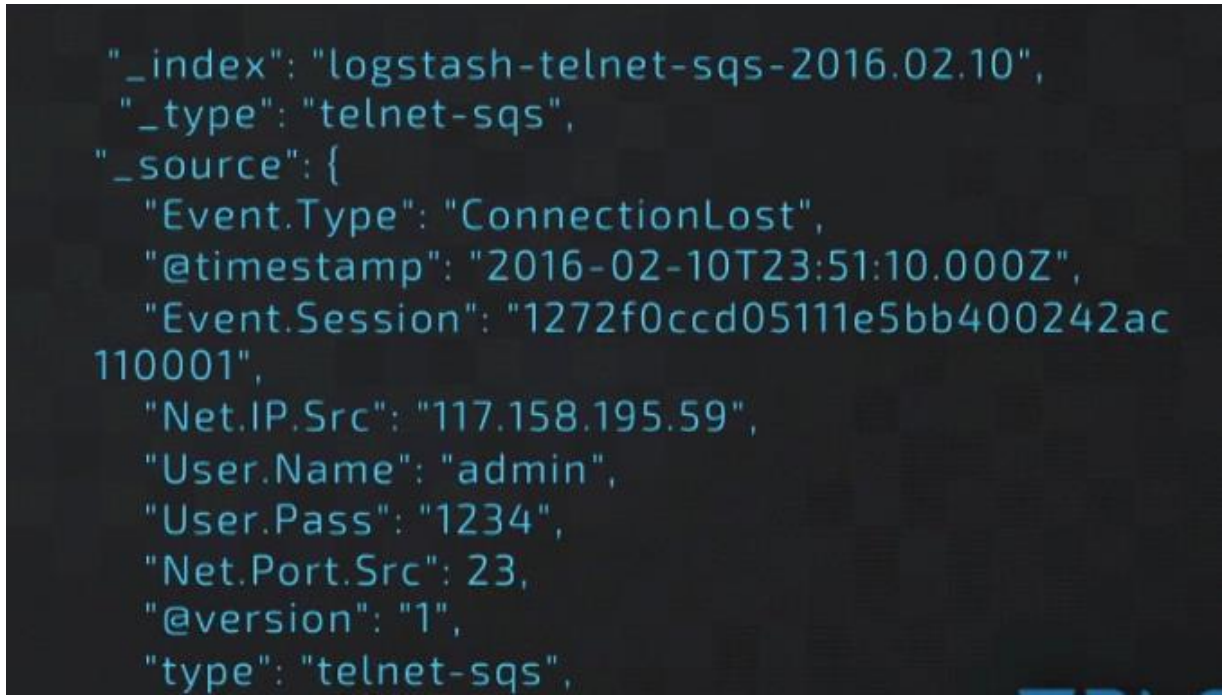

“ Elasticsearch, Logstash, and Kibana allow for real-time
verizon
Use Case | Logging, Analytics
Products | Elasticsearch, Logstash
NASA
“ With the ELK stack, we log more than 30K messages and 100K documents four times every day from the Mars Rover to optimize our space missions.”
Dan Isla, Data Scientist
Use Case | Search, Logging, Analytics
Products | Elasticsearch, Logstash, Kibana
eries a
and
internal
sing
ta is
ception
Engineer

Cisco Talos Security Intelligence and Research Group: Hunting for Hackers

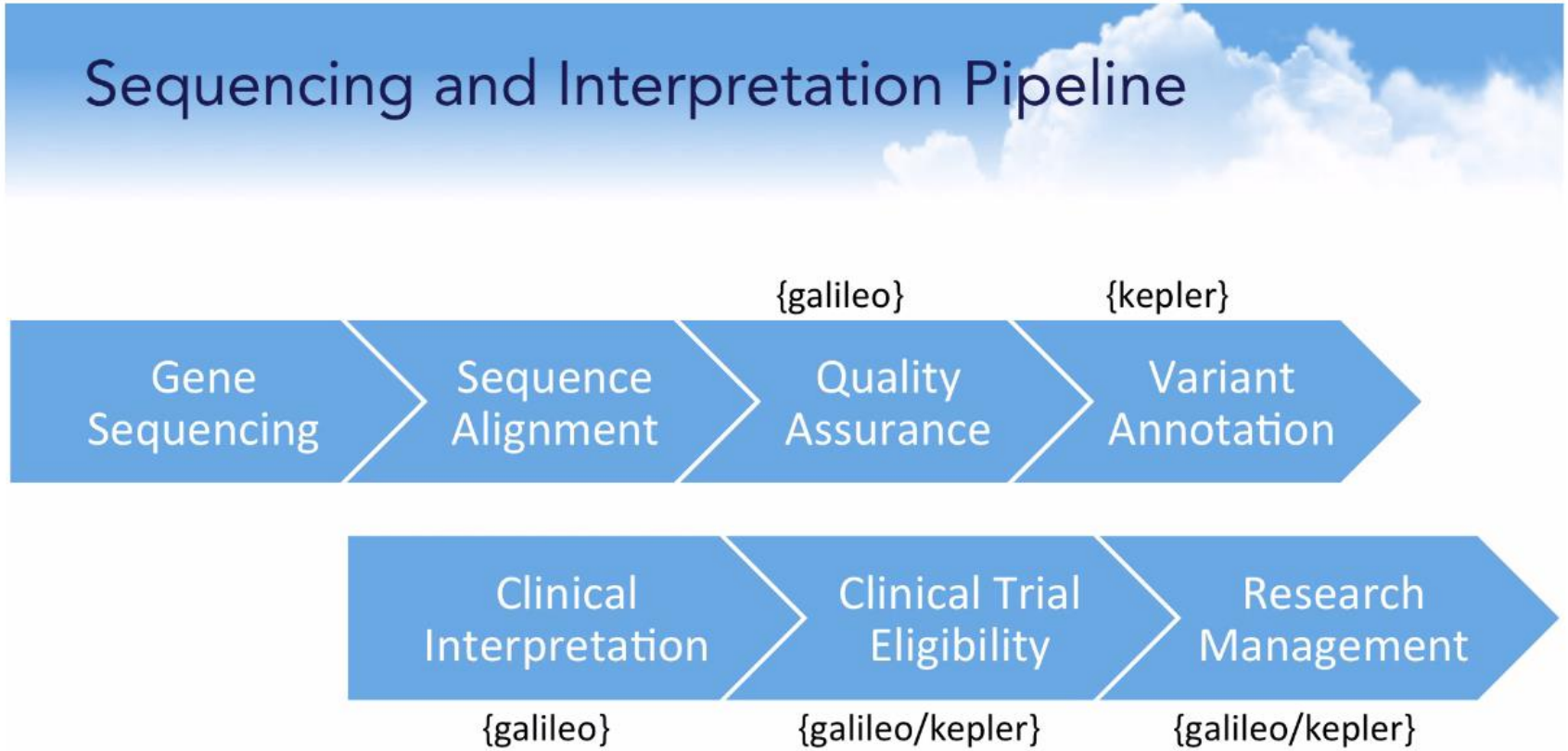


Cisco Talos use ELK to analyze…

- Sandbox data cluster
  - Dynamic malware analysis reports
    - Search for related pattern, malewares
  - ES stats
    - 10 nodes
    - 3 TB
    - 100k reports/day
    - ~8 months of data
- Honeypot cluster
  - Collect attackers' attempt
    - { Account, password } pair
    - Executed commands
    - url of download files
    - Suspicious command center for report back



Yale's {elastic}SEARCH – The Search for Cancer's Causes and Cures

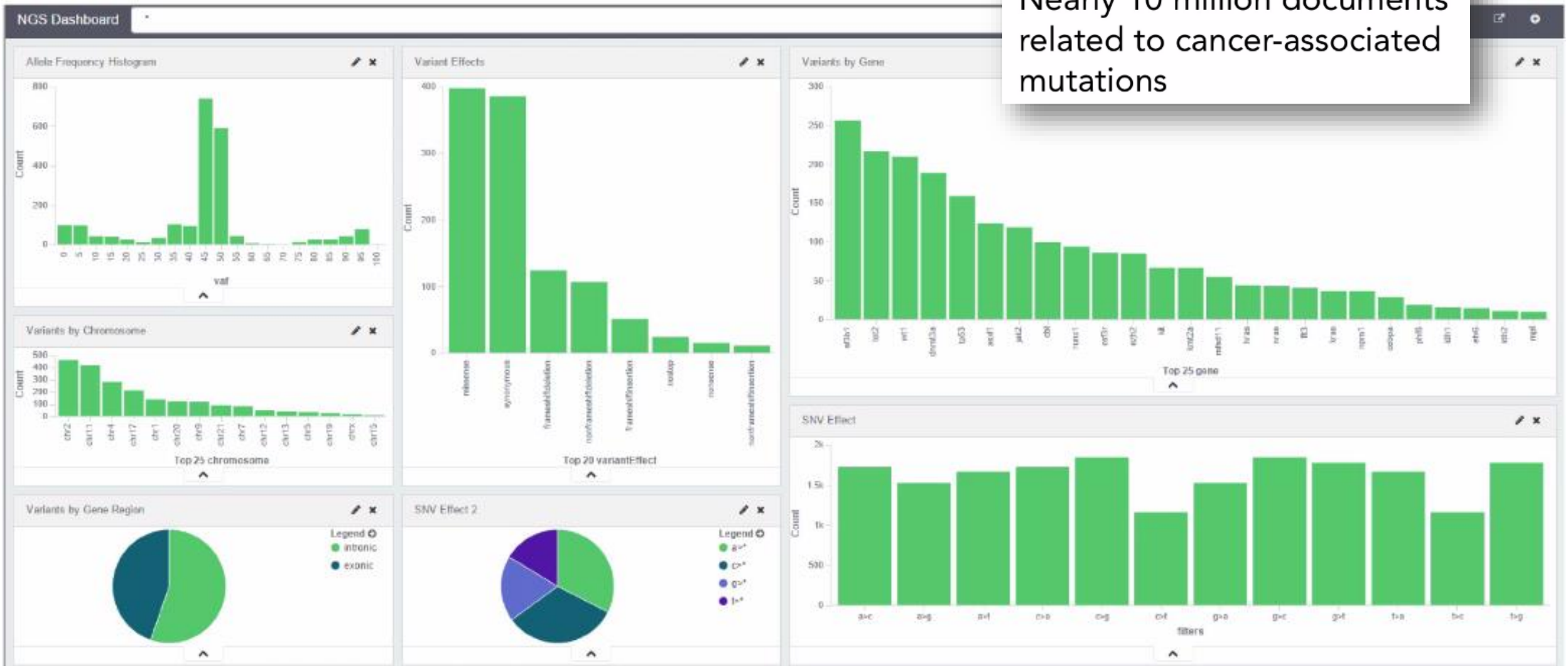


- With Next generation sequencing technology, the lab can process 8 million patients specimens yearly
- How to interpret this amount of data → what software can be used

https://www.elastic.co/elasticon/2015/sf/videos/search-for-cancer-causes-and-cures

Over 60 million variant annotations

Nearly 10 million documents related to cancer-associated mutations



NYC restaurants inspection @ELK