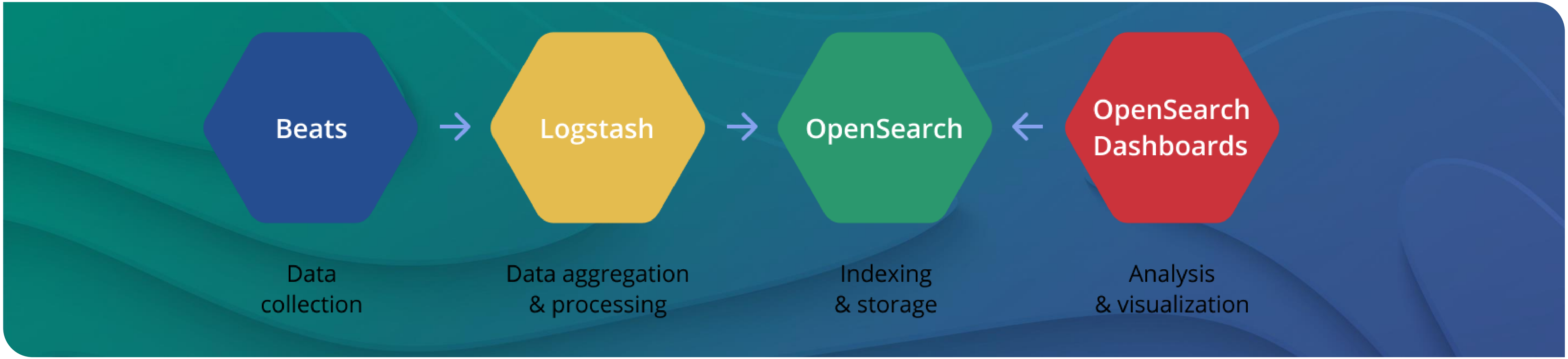
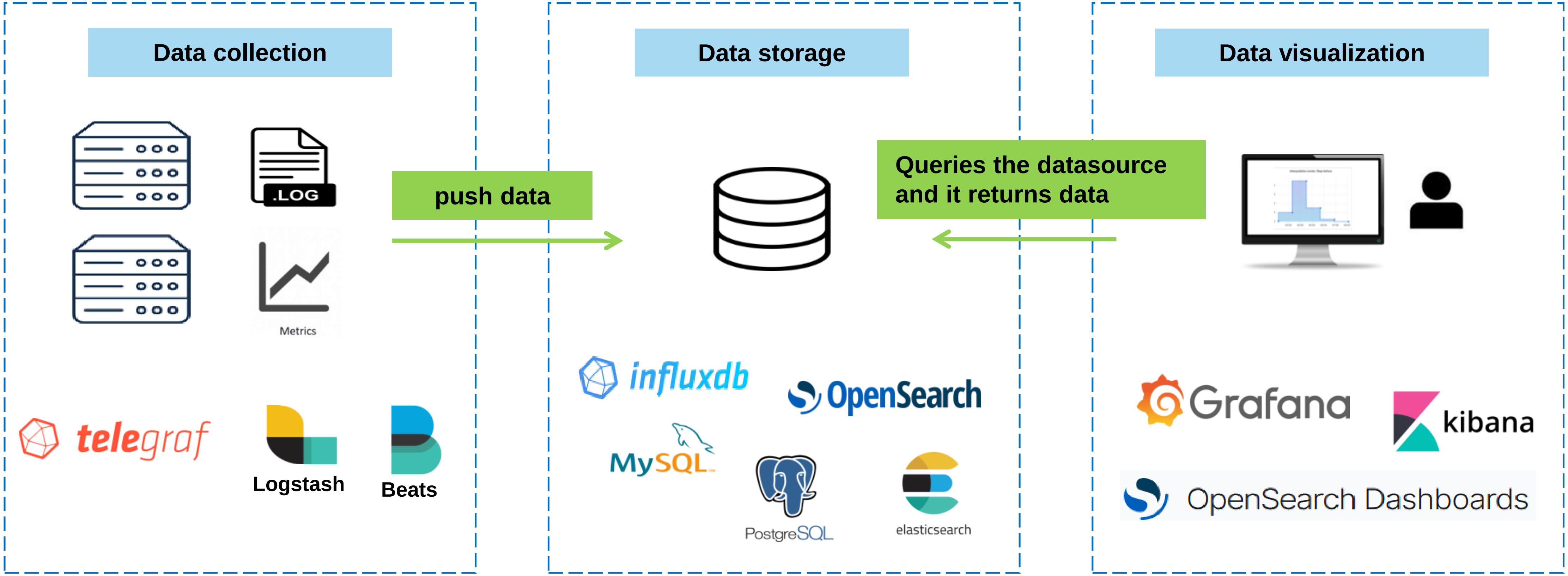
Karlsruhe Institute of Technology

Logstash and Beats



Components of monitoring architecture



Data collection

Data storage

Data visualization

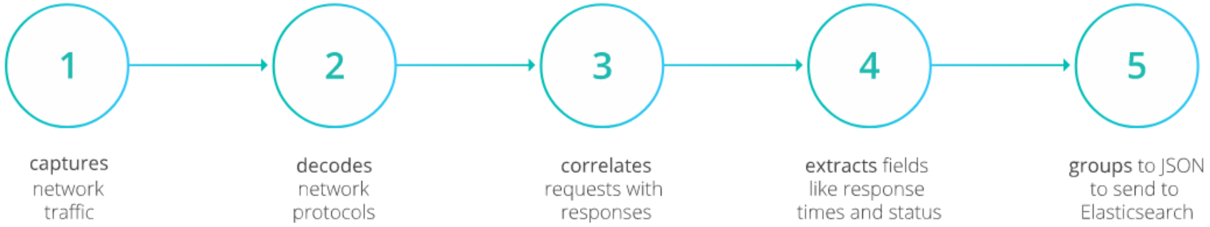
push data

Queries the datasource and it returns data

Beats

- Lightweight data shippers, meaning that Beats have a small installation footprint, use limited system resources, and have no runtime dependencies.
- Written in Go.
- Installed on the servers you want to monitor.
  - Filebeat: for logs.
  - Metricbeat: for metric data.
  - Packetbeat: for network data.
  - Winlogbeat: for Windows event logs.
  - Auditbeat: for audit data.
  - Heartbeat: for uptime monitoring.
  - Functionbeat: for cloud data (serverless)

packetbeat



1 captures network traffic

2 decodes network protocols

3 correlates requests with responses

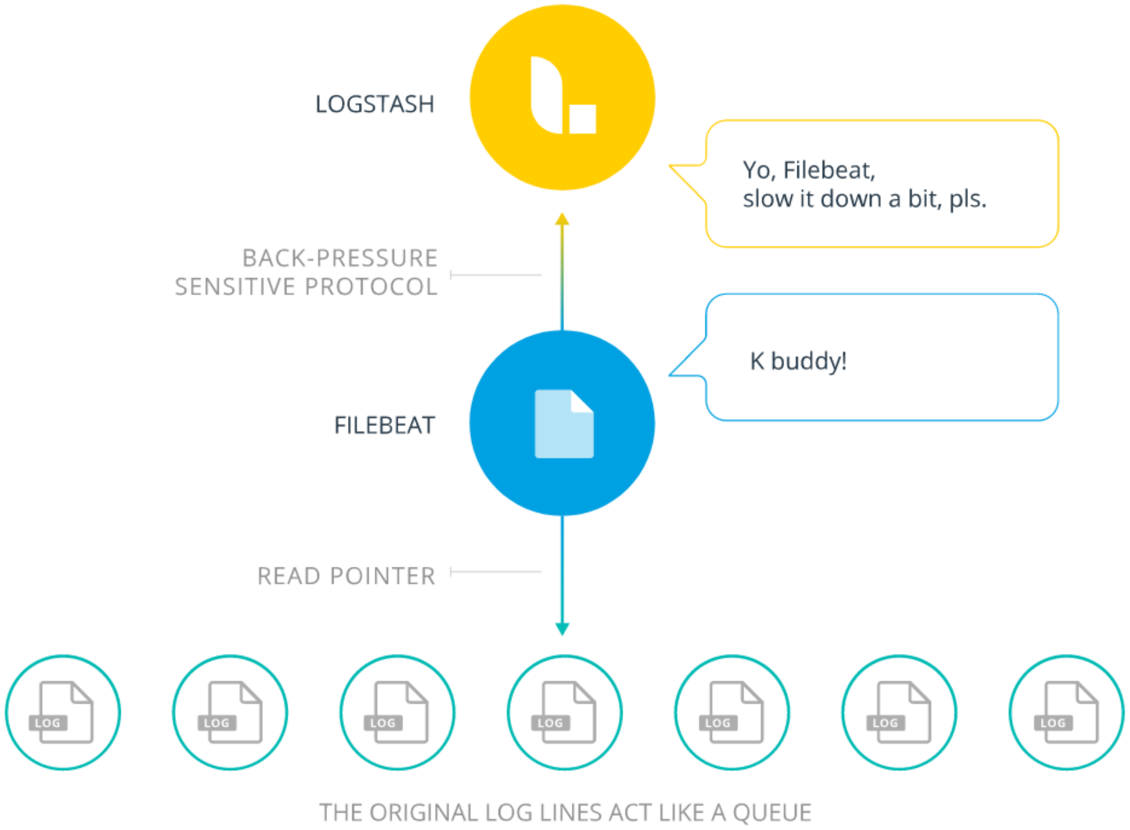
4 extracts fields like response times and status

5 groups to JSON to send to Elasticsearch

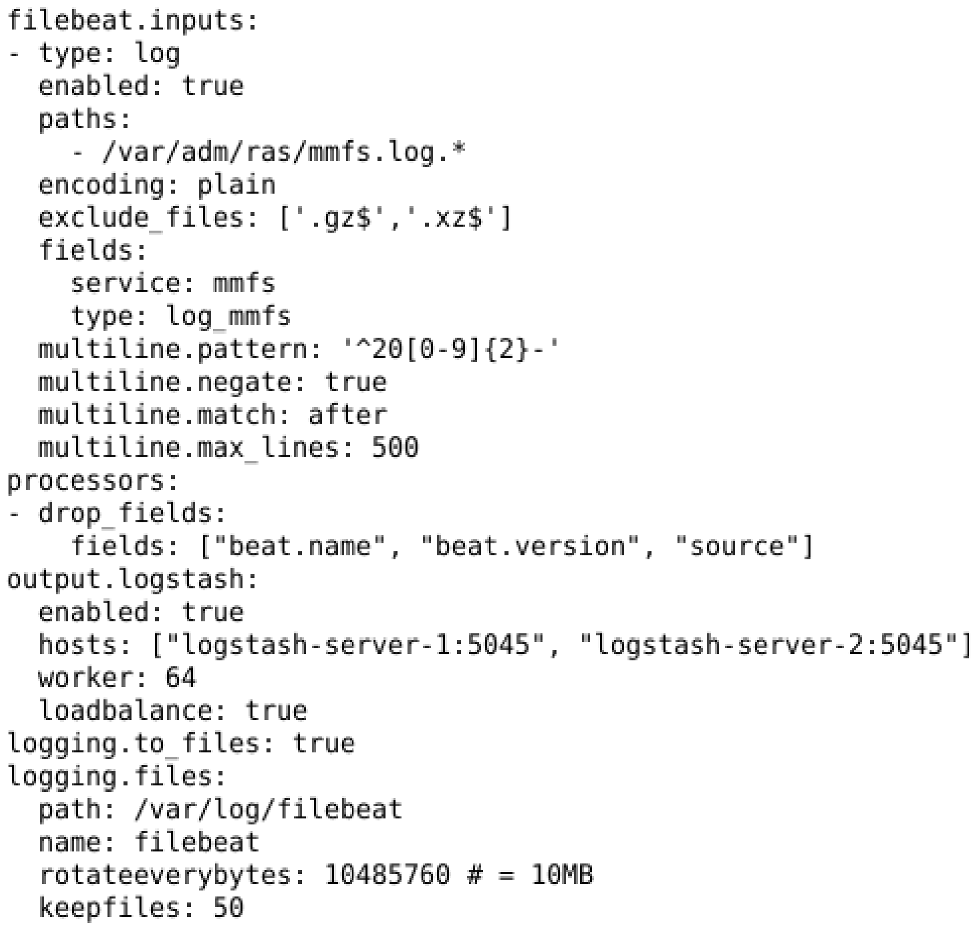
- Lightweight network packet analyzer which sends data directly to opensearch or logstash.
- Several sniffing options (pcap and af_packet).
- Flows can be configured: a group of packets sent over the same time period that share common properties, such as the same source and destination address and protocol.
- Following supported protocols: ICMP (v4 and v6) ,DHCP (v4), DNS, HTTP, AMQP 0.9.1, Cassandra, Mysql, PostgreSQL, Redis, Thrift-RPC, MongoDB, Memcache, NFS, TLS.
- Don’t miss an entry: send your network traffic info to disk. Problems downstream -> packetbeat retains your network data until things are back to normal.

filebeat

- Filebeat reads and forwards log lines and — if interrupted — remembers the location of where it left off when everything is back online.
- Filebeat uses a backpressure-sensitive protocol when sending data to Logstash or opensearch to account for higher volumes of data.
- Managed by puppet.


LOGSTASH
Yo, Filebeat, slow it down a bit, pls.
BACK-PRESSURE SENSITIVE PROTOCOL
K buddy!
FILEBEAT
READ POINTER
LOG
LOG
LOG
LOG
LOG
LOG
LOG
THE ORIGINAL LOG LINES ACT LIKE A QUEUE

filebeat: basic config



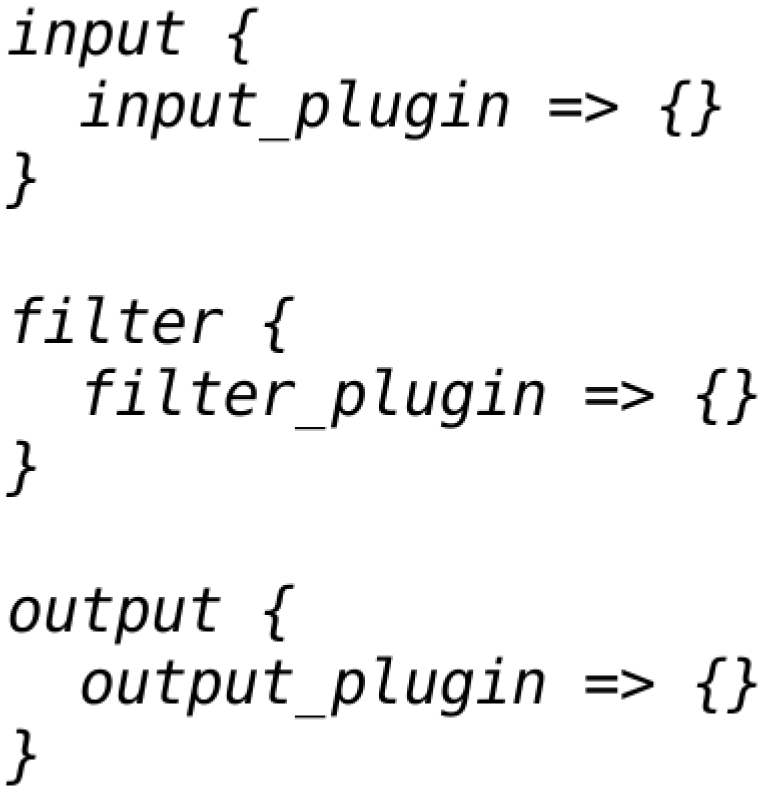
```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/adm/ras/mmfs.log.*
  encoding: plain
  exclude_files: ['.gz$','.xz$']
  fields:
    service: mmfs
    type: log_mmfs
  multiline.pattern: '^20[0-9]{2}-'
  multiline.negate: true
  multiline.match: after
  multiline.max_lines: 500
processors:
- drop_fields:
    fields: ["beat.name", "beat.version", "source"]
output.logstash:
  enabled: true
  hosts: ["logstash-server-1:5045", "logstash-server-2:5045"]
  worker: 64
  loadbalance: true
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  rotateeverybytes: 10485760 # = 10MB
  keepfiles: 50
```

What is logstash?

- Written in jRuby and requires a JVM to run.
- Logstash is a real-time event processing engine. It’s part of the OpenSearch stack which includes OpenSearch, Beats, and OpenSearch Dashboards.
- You can send events to Logstash from many different sources. Logstash processes the events and sends it one or more destinations.
- There are many plugins (input, filter, codec and output) to make this happen.
  - Codecs are essentially stream filters that can operate as part of an input or output. I will not talk about codecs.
- You can write your own plugin if one of the ~ 200 plugins does not fit your needs.
- Managed by puppet.

Structure of a pipeline in logstash

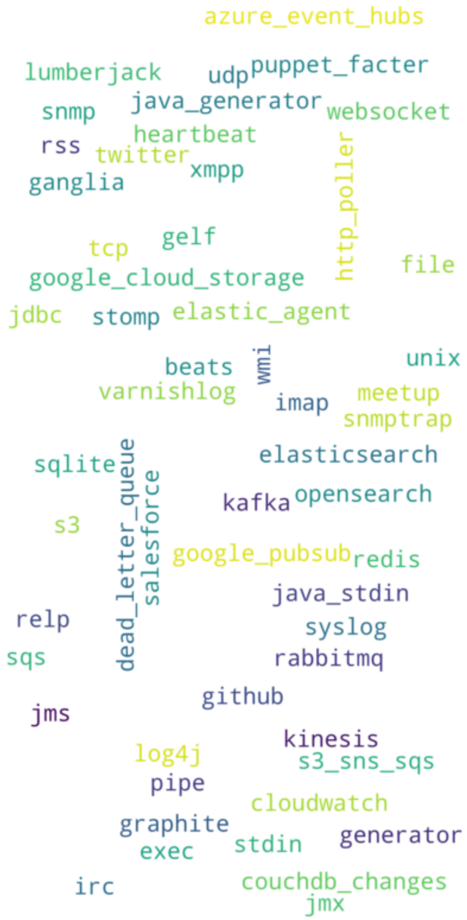
- Logstash works configuring a pipeline that has three phases—inputs, filters, and outputs.
- Each phase uses one or more plugins (Logstash has over 200 built-in plugins).



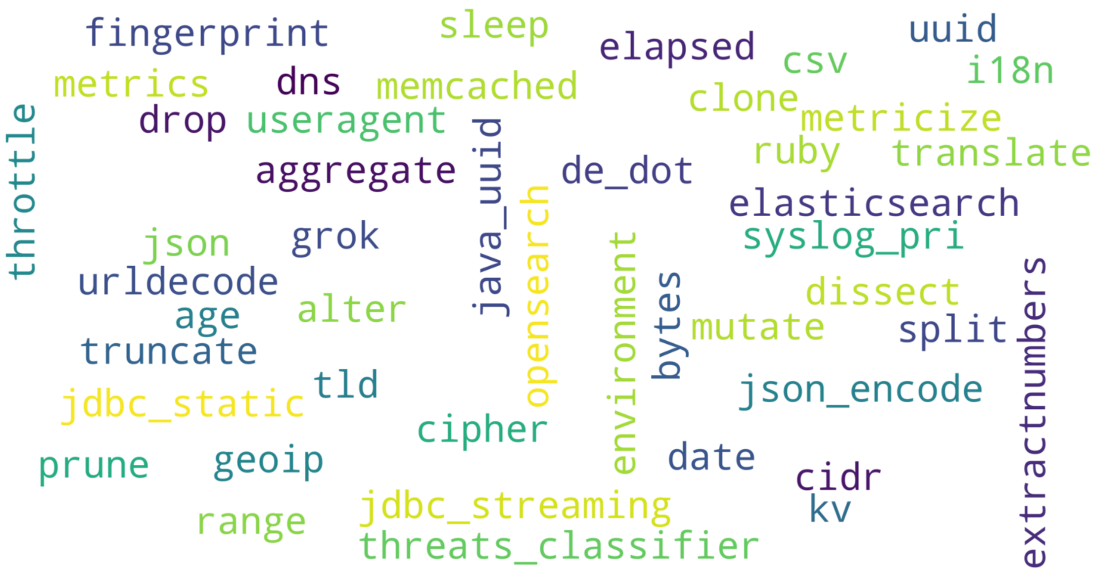
```
input {
  input_plugin => {}
}

filter {
  filter_plugin => {}
}

output {
  output_plugin => {}
}
```

Logstash plugins (codecs not included)

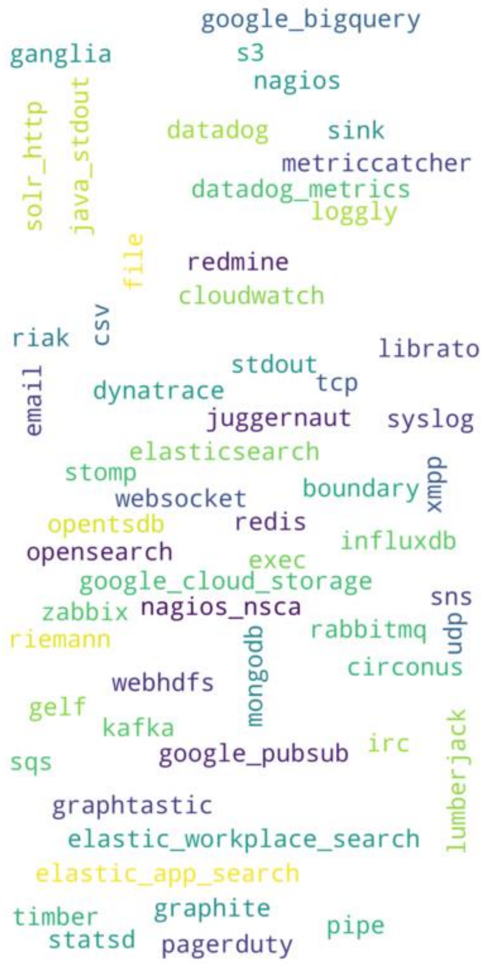
Inputs



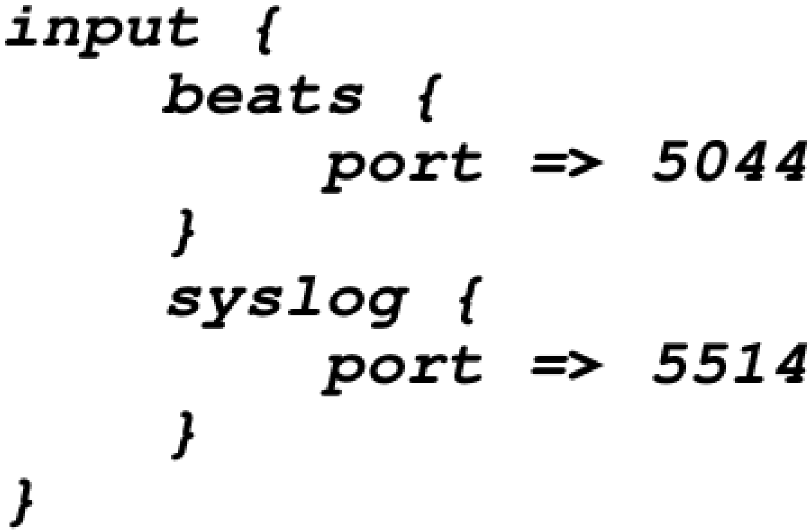
Filters



Outputs

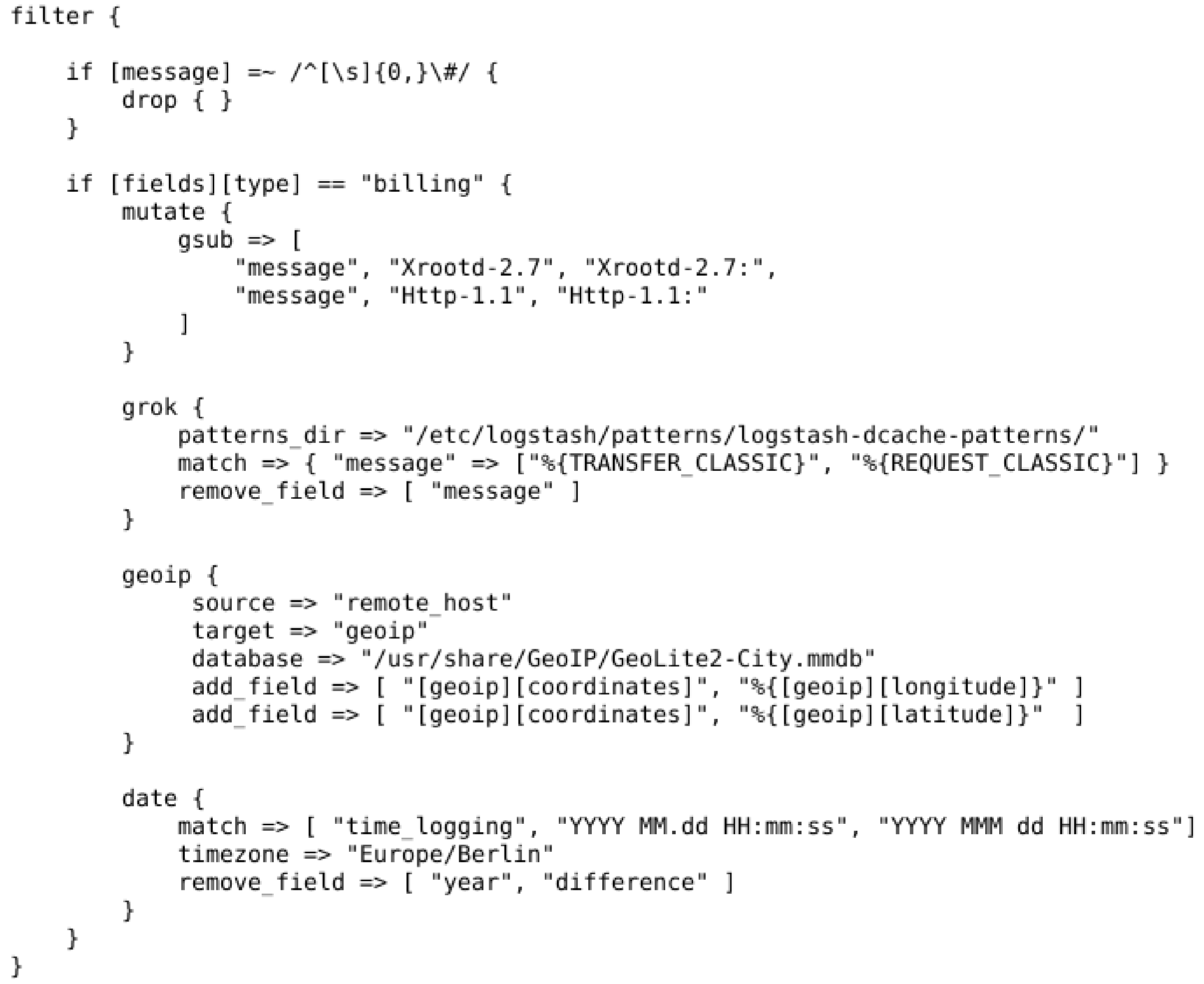


Logstash: input plugins (basic example)



```
input {
    beats {
        port => 5044
    }
    syslog {
        port => 5514
    }
}
```

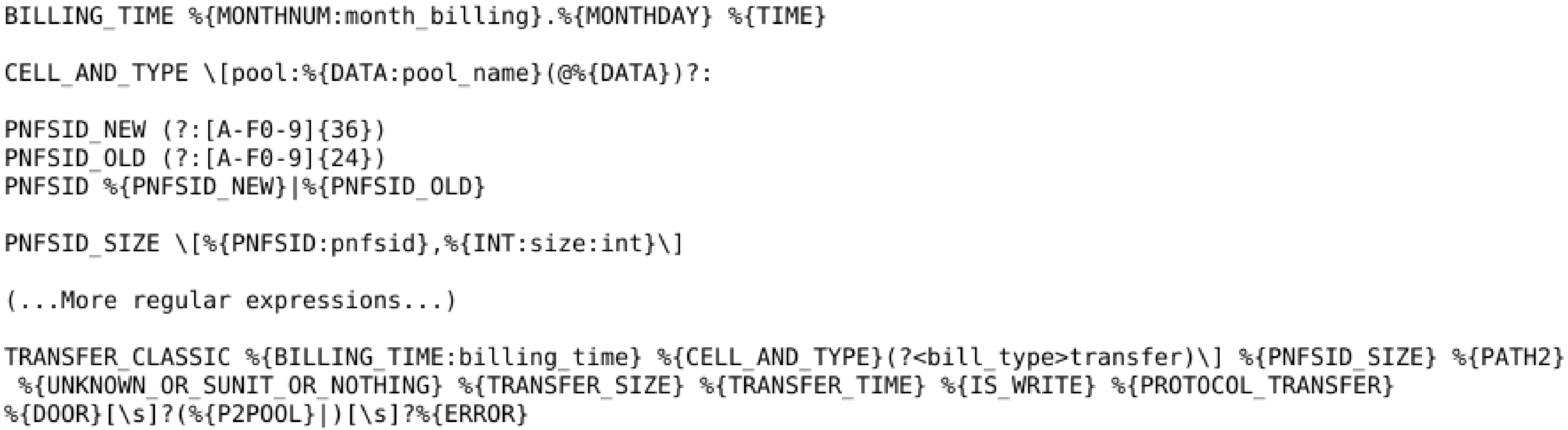
Logstash: filter plugins (basic example)



```
filter {

    if [message] =~ /^[\s]{0,}\#/ {
        drop { }
    }

    if [fields][type] == "billing" {
        mutate {
            gsub => [
                "message", "Xrootd-2.7", "Xrootd-2.7:",
                "message", "Http-1.1", "Http-1.1:"
            ]
        }

        grok {
            patterns_dir => "/etc/logstash/patterns/logstash-dcache-patterns/"
            match => { "message" => ["%{TRANSFER_CLASSIC}", "%{REQUEST_CLASSIC}"] }
            remove_field => [ "message" ]
        }

        geoip {
             source => "remote_host"
             target => "geoip"
             database => "/usr/share/GeoIP/GeoLite2-City.mmdb"
             add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
             add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
        }

        date {
            match => [ "time_logging", "YYYY MM.dd HH:mm:ss", "YYYY MMM dd HH:mm:ss"]
            timezone => "Europe/Berlin"
            remove_field => [ "year", "difference" ]
        }
    }
}
```

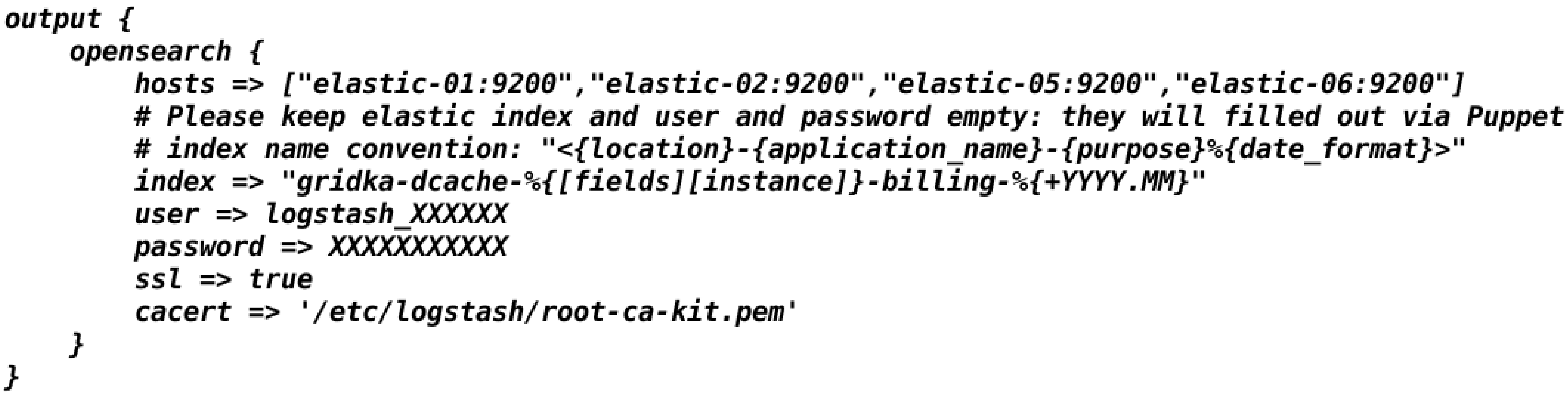
Filter plugin: grok

- Parse arbitrary text and structure it.
- Many default patterns.
- You can add your own patterns.
- Uses regular expressions (Oniguruma library)



```
BILLING_TIME %{MONTHNUM:month_billing}.%{MONTHDAY} %{TIME}

CELL_AND_TYPE \[pool:%{DATA:pool_name}(@%{DATA})?:

PNFSID_NEW (?:[A-F0-9]{36})
PNFSID_OLD (?:[A-F0-9]{24})
PNFSID %{PNFSID_NEW}|%{PNFSID_OLD}

PNFSID_SIZE \[%{PNFSID:pnfsid},%{INT:size:int}\]

(...More regular expressions...)

TRANSFER_CLASSIC %{BILLING_TIME:billing_time} %{CELL_AND_TYPE}(?<bill_type>transfer)\] %{PNFSID_SIZE} %{PATH2}
 %{UNKNOWN_OR_SUNIT_OR_NOTHING} %{TRANSFER_SIZE} %{TRANSFER_TIME} %{IS_WRITE} %{PROTOCOL_TRANSFER}
%{DOOR}[\s]?(%{P2POOL}|)[\s]?%{ERROR}
```

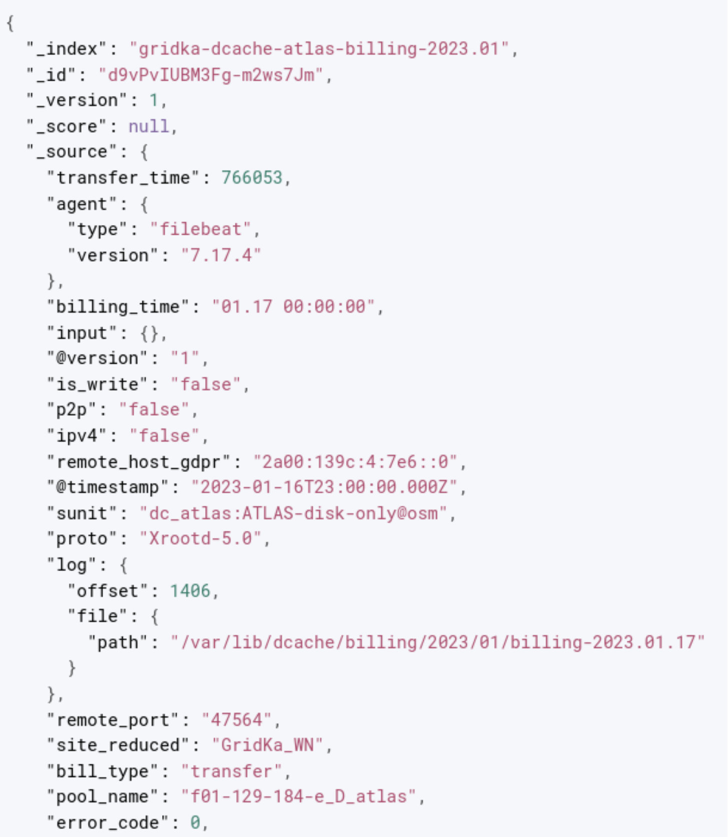
Logstash: output plugins (basic example)



```
output {
    opensearch {
        hosts => ["elastic-01:9200","elastic-02:9200","elastic-05:9200","elastic-06:9200"]
        # Please keep elastic index and user and password empty: they will filled out via Puppet
        # index name convention: "<{location}-{application_name}-{purpose}%{date_format}>"
        index => "gridka-dcache-%{[fields][instance]}-billing-%{+YYYY.MM}"
        user => logstash_XXXXXX
        password => XXXXXXXXXXX
        ssl => true
        cacert => '/etc/logstash/root-ca-kit.pem'
    }
}
```

The desired result

- In the end the logs (in JSON format) end up in OpenSearch.



```
{
  "_index": "gridka-dcache-atlas-billing-2023.01",
  "_id": "d9vPvIUBM3Fg-m2ws7Jm",
  "_version": 1,
  "_score": null,
  "_source": {
    "transfer_time": 766053,
    "agent": {
      "type": "filebeat",
      "version": "7.17.4"
    },
    "billing_time": "01.17 00:00:00",
    "input": {},
    "@version": "1",
    "is_write": "false",
    "p2p": "false",
    "ipv4": "false",
    "remote_host_gdpr": "2a00:139c:4:7e6::0",
    "@timestamp": "2023-01-16T23:00:00.000Z",
    "sunit": "dc_atlas:ATLAS-disk-only@osm",
    "proto": "Xrootd-5.0",
    "log": {
      "offset": 1406,
      "file": {
        "path": "/var/lib/dcache/billing/2023/01/billing-2023.01.17"
      }
    },
    "remote_port": "47564",
    "site_reduced": "GridKa_WN",
    "bill_type": "transfer",
    "pool_name": "f01-129-184-e_D_atlas",
    "error_code": 0,
```

Useful links

Internal: https://docs-sdm.scc.kit.edu/DocumentationELKClusterSDM/

OpenSearch: https://opensearch.org/docs/latest/

Beats: https://www.elastic.co/beats/
Filebeat (puppet module, SDM): https://git-cm.scc.kit.edu/Puppet-Modules/filebeat

Logstash: https://opensearch.org/docs/2.0/clients/logstash/index/
Logstash plugins: https://www.elastic.co/guide/en/logstash/current/input-plugins.html
Logstash config pipelines (SDM, git): https://git.scc.kit.edu/elk-sdm/

Grok debugger: https://grokdebugger.com/

Thanks a lot for your attention!

Questions???

Backup slides

Compatibility matrices for Beats

	Beats OSS 7.0.0 to 7.11.x**	Beats OSS 7.12.x*	Beats 7.13.x
Elasticsearch OSS 7.0.0 to 7.9.x	*Yes*	*Yes*	No
Elasticsearch OSS 7.10.2	*Yes*	*Yes*	No
ODFE 1.0 to 1.12	*Yes*	*Yes*	No
ODFE 1.13	*Yes*	*Yes*	No
OpenSearch 1.x to 2.x	Yes via version setting	Yes via version setting	No
Logstash OSS 7.0.0 to 7.11.x	*Yes*	*Yes*	*Yes*
Logstash OSS 7.12.x*	*Yes*	*Yes*	*Yes*
Logstash 7.13.x with OpenSearch output plugin	*Yes*	*Yes*	*Yes*

Compatibility matrices for logstash

	Logstash OSS 7.0.0 to 7.11.x	Logstash OSS 7.12.x*	Logstash 7.13.x-7.16.x without OpenSearch output plugin	Logstash 7.13.x-7.16.x with OpenSearch output plugin	Logstash 8.x+ with OpenSearch output plugin
Elasticsearch OSS 7.0.0 to 7.9.x	*Yes*	*Yes*	*No*	*Yes*	*Yes*
Elasticsearch OSS 7.10.2	*Yes*	*Yes*	*No*	*Yes*	*Yes*
ODFE 1.0 to 1.12	*Yes*	*Yes*	*No*	*Yes*	*Yes*
ODFE 1.13	*Yes*	*Yes*	*No*	*Yes*	*Yes*
OpenSearch 1.x to 2.x	Yes via version setting	Yes via version setting	*No*	*Yes*	Yes, with Elastic Common Schema Setting