Karlsruhe Institute of Technology

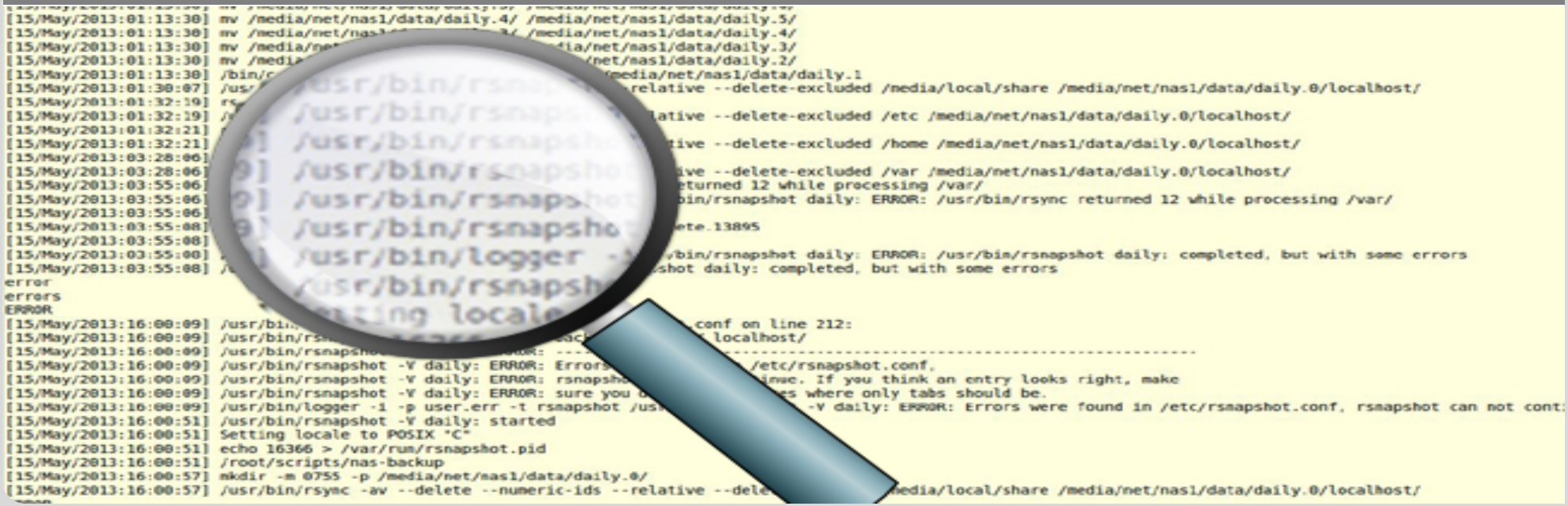
GridKa School 2015

Modern Programming
Virtualization
Big Data

Introduction to Elasticsearch and Logstash.

Samuel Ambroj Pérez, Kajorn Pathomkeerati | September 9, 2015 | ELK tutorial GKS 2015

STEINBUCH CENTER FOR COMPUTING (SCC), INSTITUTE FOR APPLIED COMPUTER SCIENCE (IAI)



Outline

A few lines ... about logs

log = timestamp + data

Listing 1: /var/log/ermm/lsmess logfile

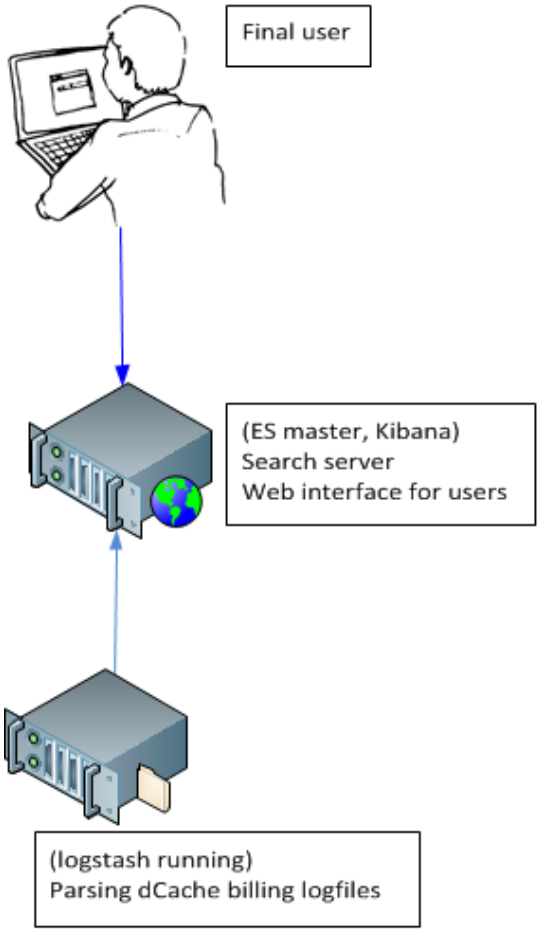
```
2014-10-30T00:00:12 EMM1110I: tsm.tsmserver_grid1:1 has unmounted cartridge UR7467
     from drive 000,00,01,09.
2014-10-30T00:00:13 EMM1021I: tsm.tsmstg_f01-075-111:1 mount request for cartridge
     UR4635 queued because all drives in library are in use.
2014-10-30T00:00:13 EMM1867I: ERMMSystem:ERMMAdmin:1 version:1.1.1.41 on host
     127.0.0.1 has been connected to MediaManager.
2014-10-30T00:00:13 EMM1867I: tsm:tsmserver_grid1:1 version:1.1.1.6 on host
     10.97.13.115 has been connected to MediaManager.
2014-10-30T00:00:13 EMM1867I: tsm:tsmserver_grid1:1 version:1.1.1.6 on host
     10.97.13.115 has been connected to MediaManager.
2014-10-30T00:00:14 EMM1867I: ERMMSystem:ERMMAdmin:1 version:1.1.1.41 on host
     127.0.0.1 has been connected to MediaManager.
2014-10-30T00:00:15 EMM1020I: tsm.tsmserver_grid1:1 mount request for cartridge UR7467
      and drive 000,00,01,09 dispatched.
2014-10-30T00:01:00 EMM1019I: tsm.tsmserver_grid1:1 has mounted volume UR7467 into
     drive 000,00,01,09.
2014-10-30T00:01:31 EMM1867I: tsm:tsmstg_f01-075-111 version:1.1.1.6 on host
     10.65.75.111 has been connected to MediaManager.
2014-10-30T00:01:31 EMM1867I: tsm:tsmstg_f01-075-111 version:1.1.1.6 on host
     10.65.75.111 has been connected to MediaManager.
```

Is it funny?

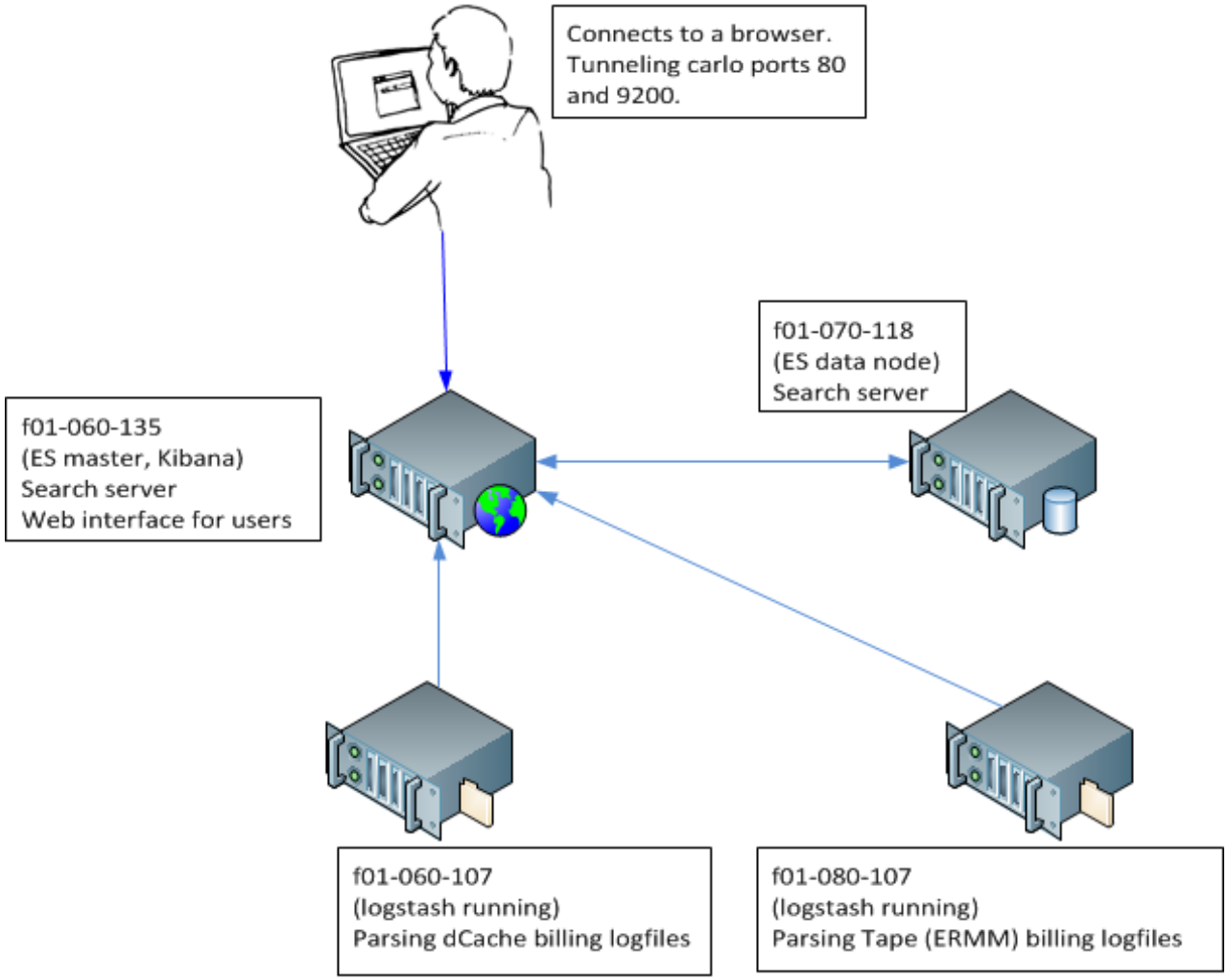
- Ten lines are ok.
- Thousand or million lines are a pain in the neck.
- grep, awk, sed, perl help.
- NOT FOR IMPATIENT PEOPLE.

One pleasant solution: ELK

Simplest configuration



Current experimental setup



General ideas about logstash

General ideas

- Ship logs from any source and send to ES.
- Parse them.
- Get the right timestamp.
- Highly scalable.
- Possibility of redundant setups.

Fully free and fully open source. License is Apache 2.0.
logstash is now a part of the Elasticsearch family.
More info: `https://www.elastic.co/products/logstash`

Logstash structure

Listing 2: Basic logstash structure.

```
input {
  file {
    path => ""
    sincedb_path => "/dev/null"
    start_position => beginning
    type => ""
  }
}
filter {
    grok {
      patterns_dir => "/etc/logstash/patterns"
      match => [ "message", "%{TRANSFER_CLASSIC}" ]
    }
  date { #To get the correct timestamp
    locale => "billing_time"
    match => [ "billing_time", "dd MMM YYYY HH:mm:ss" ]
    timezone => "UTC"
  }
}
output {
  elasticsearch {
    host => "localhost"
    index => "name-%{+YYYY.MM.dd}"
  }
}
```

logstash plugins

There are four different types:

- Input plugins ▸ input-plugins
- Output plugins ▸ output-plugins
- Codec plugins ▸ codec-plugins
- Filter plugins ▸ filter-plugins

input plugins (not the whole list)

Input plugins

An input plugin enables a specific source of events to be read by Logstash.

The following input plugins are available:

c d e f g h i j k l m p r s t u v w x z

- couchdb_changes
- drupal_dblog
- elasticsearch
- exec
- eventlog
- file
- ganglia
- gelf
- generator
- graphite
- github
- heartbeat
- heroku
- http
- http_poller

output plugins (not the whole list)

Output plugins

An output plugin sends event data to a particular destination. Outputs are the final stage in the event pipeline.

The following output plugins are available:

b c d e f g h i j k l m n o p r s t u w x z

- boundary
- circonus
- csv
- cloudwatch
- datadog
- datadog_metrics

- email
- elasticsearch
- exec
- file
- google_bigquery
- google_cloud_storage
- ganglia
- gelf
- graphtastic

codec plugins (not the whole list)

Codec plugins

A codec plugin changes the data representation of an event. Codecs are essentially stream filters that can operate as part of an input or output.

The following codec plugins are available:

a c d e f g j l m n o p r s

- avro
- compress_spooler
- cloudtrail
- cloudfront
- collectd
- dots
- edn_lines
- edn
- es_bulk
- fluent
- gzip_lines
- graphite

filter plugins (not the whole list)

Filter plugins

A filter plugin performs intermediary processing on an event. Filters are often applied conditionally depending on the characteristics of the event.

The following filter plugins are available:

a c d e f g i j k m p r s t u x z

- aggregate
- alter
- anonymize
- collate
- csv
- cidr
- clone
- cipher
- checksum
- date
- dns
- drop

- elasticsearch
- extractnumbers
- fingerprint
- geoip
- grok

Some comments about grok filter

Basics of grok filter

- Parse arbitrary text and structure it.
- $\sim$ 120 patterns by default (/opt/logstash/patterns/grok-patterns file).
- You can add your own patterns (/etc/logstash/patterns/<your_file>)
- Uses regular expressions (Oniguruma library) ▸ Oniguruma site

Listing 3: Some customised regular expressions.

```
PROTOCOL_WEBDAV \{(%{USERNAME:proto}):(\d+):(%{USERNAME}):(%{USERNAME}):%{PATH}\}
PROTOCOL_XROOTD \{(%{PROTO_XROOTD:proto})(%{USERNAME:remote_host}):(%{POSINT:
     remote_port:int})\}
PROTOCOL %{PROTOCOL_GENERAL}|%{PROTOCOL_HTTP}|%{PROTOCOL_XROOTD}
ERROR \{%{NONNEGINT:error_code:int}:\"%{DATA:error_msg}\"\}
TRANSFER %{BILLING_TIME:billing_time} %{CELL_AND_TYPE} %{PNFSID_SIZE} %{DOOR} %{PATH}
      %{SUBJECTS} %{SUNIT} %{TRANSFER_SIZE} %{TRANSFE
R_TIME} %{IS_WRITE} %{PROTOCOL} %{ERROR}
TRANSFER_CLASSIC %{BILLING_TIME:billing_time} %{CELL_AND_TYPE} %{PNFSID_SIZE} %{PATH}
     %{SUNIT} %{TRANSFER_SIZE} %{TRANSFER_TIME} %{IS_
WRITE} %{PROTOCOL} %{DOOR} %{ERROR}
```

General ideas about ES

General ideas

- Search server based on Lucene.
- Full text search.
- Schema free (Toss it a JSON document).
- Built to scale horizontally.
- ES clusters are resilient (high availability).
- Restful API (JSON over HTTP).

- Fully free and fully open source. License is Apache 2.0.
- More info: `http://www.elasticsearch.org/`

Some important concepts for ES

Some important concepts for ES

- **Index**: like a database in a RD. Logical namespace which maps to one or more prim. shards and can have zero or more repl. shards.
- **Document**: JSON document stored in ES. Like a row in a table in a RD. Each document is stored in an index and has a type and an id.
- **Shard**: single Lucene instance. A low-level “worker” unit managed automatically by ES. ES distributes shards amongst all nodes.
- **Primary shard**: Each document is stored in a single primary shard. When you index a document, it is indexed first on the primary shard, then on all replicas of the primary shard.
- **Replica shard**: A replica is a copy of the primary shard:
  1. increase failover: a replica shard can be promoted to a primary shard.
  2. increase performance: get and search requests can be handled by primary or replica shards.

Talking to ES

There are two ways:

- JAVA API (port 9300).
- RESTful API with JSON over HTTP (port 9200).

More info: ▸ Talking to Elasticsearch

Talking to ES: JAVA API

Java API

edit

If you are using Java, Elasticsearch comes with two built-in clients that you can use in your code:

Node client

The node client joins a local cluster as a *non data node*. In other words, it doesn't hold any data itself, but it knows what data lives on which node in the cluster, and can forward requests directly to the correct node.

Transport client

The lighter-weight transport client can be used to send requests to a remote cluster. It doesn't join the cluster itself, but simply forwards requests to a node in the cluster.

Both Java clients talk to the cluster over port *9300*, using the native Elasticsearch *transport* protocol. The nodes in the cluster also communicate with each other over port 9300. If this port is not open, your nodes will not be able to form a cluster.

TIP

The Java client must be from the same version of Elasticsearch as the nodes; otherwise, they may not be able to understand each other.

Talking to ES: RESTful API with JSON over HTTP

Karlsruhe Institute of Technology

edit

All other languages can communicate with Elasticsearch over port *9200* using a RESTful API, accessible with your favorite web client. In fact, as you have seen, you can even talk to Elasticsearch from the command line by using the `curl` command.

NOTE

Elasticsearch provides official clients for several languages—Groovy, JavaScript, .NET, PHP, Perl, Python, and Ruby—and there are numerous community-provided clients and integrations, all of which can be found in the Guide.

A request to Elasticsearch consists of the same parts as any HTTP request:

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

The parts marked with `< >` above are:

`VERB`	The appropriate HTTP *method* or *verb*: `GET`, `POST`, `PUT`, `HEAD`, or `DELETE`.
`PROTOCOL`	Either `http` or `https` (if you have an `https` proxy in front of Elasticsearch.)

Basic curl commands with ES

Listing 4: Basic curl commands with ES.

```
Check the health and the nodes:
[root@f01-060-135 ~]# curl 'localhost:9200/_cat/health?v'
epoch      timestamp cluster       status node.total node.data shards  pri relo init
     unassign
1423043710 10:55:10  clustersamuel green           2         2   3590 1795    0    0
            0

List the current indexes (Showing only a few lines):
[root@f01-060-135 ~]# curl localhost:9200/_cat/indices?v
health index                    pri rep docs.count docs.deleted store.size pri.store.
     size
green  lsmess-billing-2014.03.11   5   1      16471            0      9.2mb
     4.5mb
green  dcache-billing-2015.12.04   5   1     180336            0    110.4mb
     55.3mb
green  lsmess-billing-2014.08.07   5   1       7775            0        5mb
     2.5mb

Checking the number of primary shards:
[root@f01-060-135 ~]# curl localhost:9200/_cat/indices?v | awk '{x=x+$3} END {print x
     }'
1795

Deleting all dcache indexes:
[root@f01-060-135 ~]# curl -XDELETE 'localhost:9200/dcache*?pretty'
```

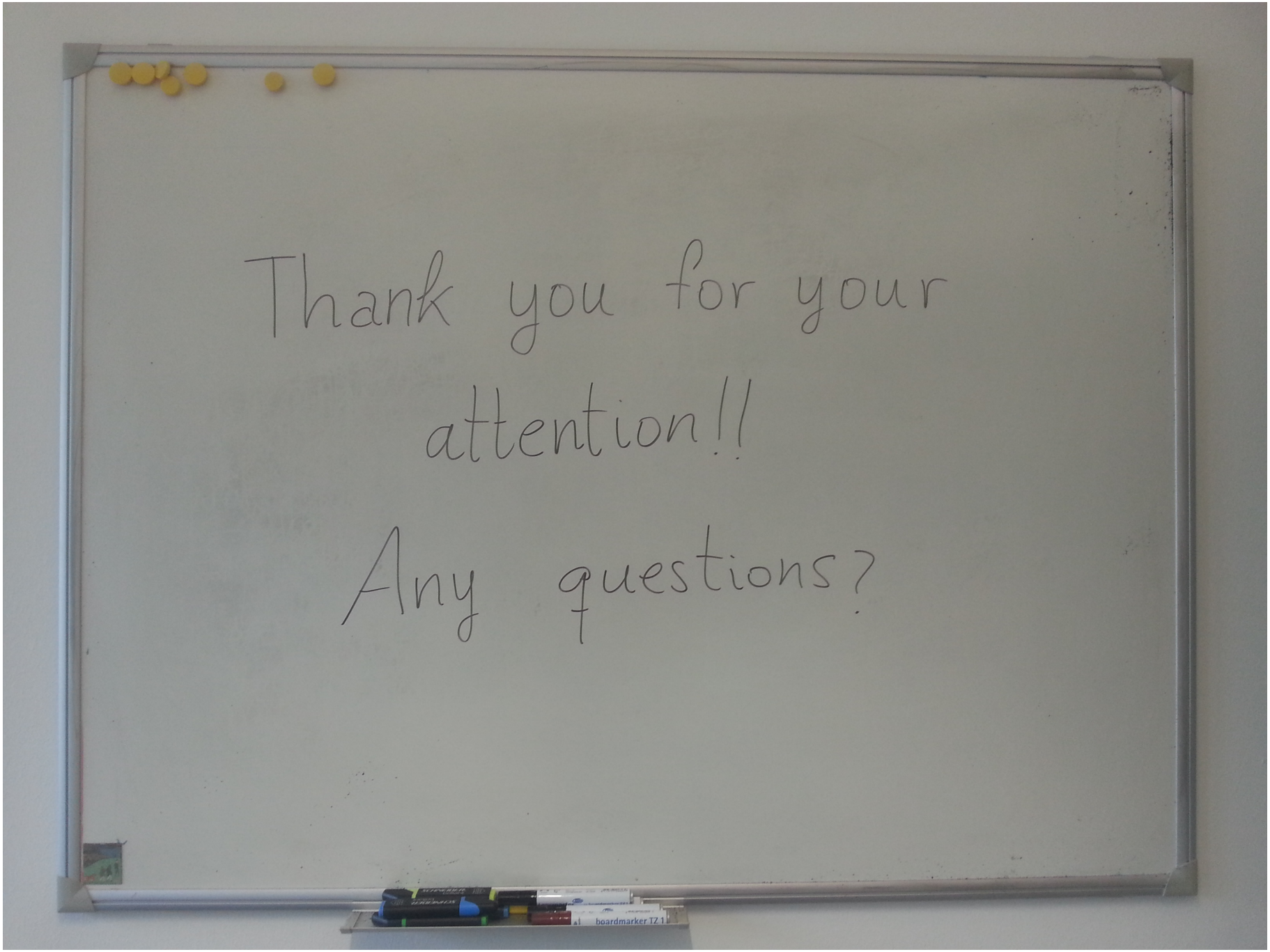
ES configuration files

Karlsruhe Institute of Technology

Listing 5: Configuration ES files for master and data node.

```
[root@f01-060-135 ~]# egrep -v '^#' /etc/elasticsearch/elasticsearch.yml | sed '/^$/ d
    '
cluster.name: clustersamuel
node.master: true
node.data: true
index.number_of_shards: 5
index.number_of_replicas: 1
bootstrap.mlockall: true
indices.recovery.max_bytes_per_sec: 100mb

[root@f01-070-118-e ~]# egrep -v '^#' /etc/elasticsearch/elasticsearch.yml | sed '/^$/
    d'
cluster.name: clustersamuel
node.master: false
node.data: true
index.number_of_shards: 5
index.number_of_replicas: 1
bootstrap.mlockall: true
network.publish_host: 10.65.70.118
indices.recovery.max_bytes_per_sec: 100mb
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["f01-060-135"]
```


Thank you for your
attention!!
Any questions?
boardmarker TZ 1