Watching File Storage and Transfers with Elastic Stack at BNL

Matt Snyder
Brookhaven National Laboratory
msnyder@bnl.gov

HEPiX Autumn Workshop
14 October 2020

BROOKHAVEN NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

BROOKHAVEN SCIENCE ASSOCIATES

Overview

- Elastic Overview
- Node configurations (elk01-04)
- Our Implementation
  - Logstash configurations
  - Elastic Indexes
- Storage Use and Growth
- Summary and Future Plans

Elastic Overview

- At simplest, it's "document" ingestion, enrichment and analysis and NOT single-point-of-truth database (a "document" could be log entry, wikipedia entry, blog post, db transaction etc. etc.)

  /var/log/app.log

  DATE=202009010800 USER=matt IP=1.2.3.4 EVENT=authentication

  DATE=202009010801 USER=matt IP=1.2.3.4 EVENT=file create FILE=stuff.txt

  DATE=202009010802 USER=matt IP=1.2.3.4 EVENT=file edit FILE=stuff.txt

- Elasticsearch stores each entry above
  - As an atomic document for representation in kibana etc.
  - But more importantly, as an inverted index that allows for fast searching
- To be clear: Apache Lucene is what's indexing these documents
- Elasticsearch is *really* the outer layer that handles what to do with these onto storage
  - Distributed (NoSQL), node-aware
  - Shard primaries and replicas
  - Not intended as an RBDMS with persistent, historic atomic points of truth

Elastic Overview

- Index <= Shards <= Documents
  - Index represents some logical thing sent to Elasticsearch
    - Could be log or some logical extraction of log (event or just time window)
    - Can combine indexes with aliases to search or visualize across indexes
  - Shards spread the indexed data across the nodes
    - Shard = 1 lucene index
    - Primary shard is stored on 1 node and can be written to
    - Replica shard is spread across nodes and is only read from
  - *continued...*

Elastic Overview

- Index <= Shards <= Documents
  - Document
    - Flat, independent, self-containing thing (perhaps event, metric, blog post)
    - Self-containing meaning it need not be initially joined to some other index like relational database; allows for “fuzziness”
    - Since the storage is NoSQL can support denormalized/semi-structured data
    - **Besides the initial index mapping no presupposition of a future purpose**
      - idea is to **store quick** then **extract** and **enrich** later

Node Configurations

- ELK cluster consists of 4 nodes (elk01,elk02,elk03,elk04)
- Sizes
  - 16GB of RAM
  - 4 CPUs
  - 300 GB disks
  - 256 GB of disk space is allocated to /var
- Elasticsearch 7.6.2 running on all 4 nodes
- Logstash 7.6.2 (elk04)
- Kibana 7.6.2 (elk04)
- Nightly incremental snapshots written to nfs01 mount

Our Implementation

- For our implementation so far, “documents” are
  - log entries
  - System metric snapshots
  - Database views (user matching for storage counts and file interactions)
- BNLBox
  - Nextcloud log index
  - Apache log index
  - 3 database states (count of users, total storage, storage by user)
- Globus
  - Globus log (level at highest verbosity)
- “beats” agents
  - Filebeats (we have all of these pointed to logstash)
  - Metricbeats (all system heartbeats are going directly to elasticsearch)

Our Implementation - Logstash

- 3 Logstash pre-processing pipelines

  # grokking of all nextcloud and apache logs

  - pipeline.id: user_data

    path.config: "/usr/share/logstash/config/conf.d/bnlbox/user_data.conf"

  # sql calls out to bnlbox postgres db for storage counts

  - pipeline.id: storage_data

    path.config: "/usr/share/logstash/config/conf.d/bnlbox/storage_data.conf"

  # parsing/grokking of globus server log

  - pipeline.id: globus_data

    path.config: "/usr/share/logstash/config/conf.d/globus/globus.conf"

Our Implementation - Logstash

- Enrichment Example → nextcloud.log

```
[message] =~ /Login successful:/ {
        mutate {
                add_field => {"action" => "Login Successful"}
                add_tag => ["login"]
        }
}
[message] =~ /File created:/ {
        mutate {
                add_field => {"action" => "File Created"}
                add_tag => ["file_edit"]
        }
        grok {
          match => {
            "message" => "File created: \"%{GREEDYDATA:file}\""
          }
        }
}
[message] =~ /File written to:/ {
        mutate {
                add_field => {"action" => "File Written"}
                add_tag => ["file_edit"]
        }
        grok {
          match => {
            "message" => "File written to: \"%{GREEDYDATA:file}\""
          }
        }
}
```

Our Implementation - Logstash

- Example → Globus.conf

```
input {
  beats {
  }
}
filter {
    #grok {
    #  patterns_dir => "/usr/share/logstash/config/conf.d/globus/patterns"
    #  match => { "message" => "%{GRIDFTP}" }
    #}
    kv {
      transform_key => "lowercase"
    }
    mutate {
      rename => {"[host]" => "[hostname]"}
    }
    mutate {
      convert => {"nbytes" => "integer"}
    }
    geoip {
      source => "dest"
    }
}
output {
  elasticsearch {
    index => "globus"
  }
}
```

Our Implementation - metricbeats

Metricbeats are sending system heartbeats every hour for bnlbox nodes and elk cluster nodes (bnlbox01-03, elk01-04)

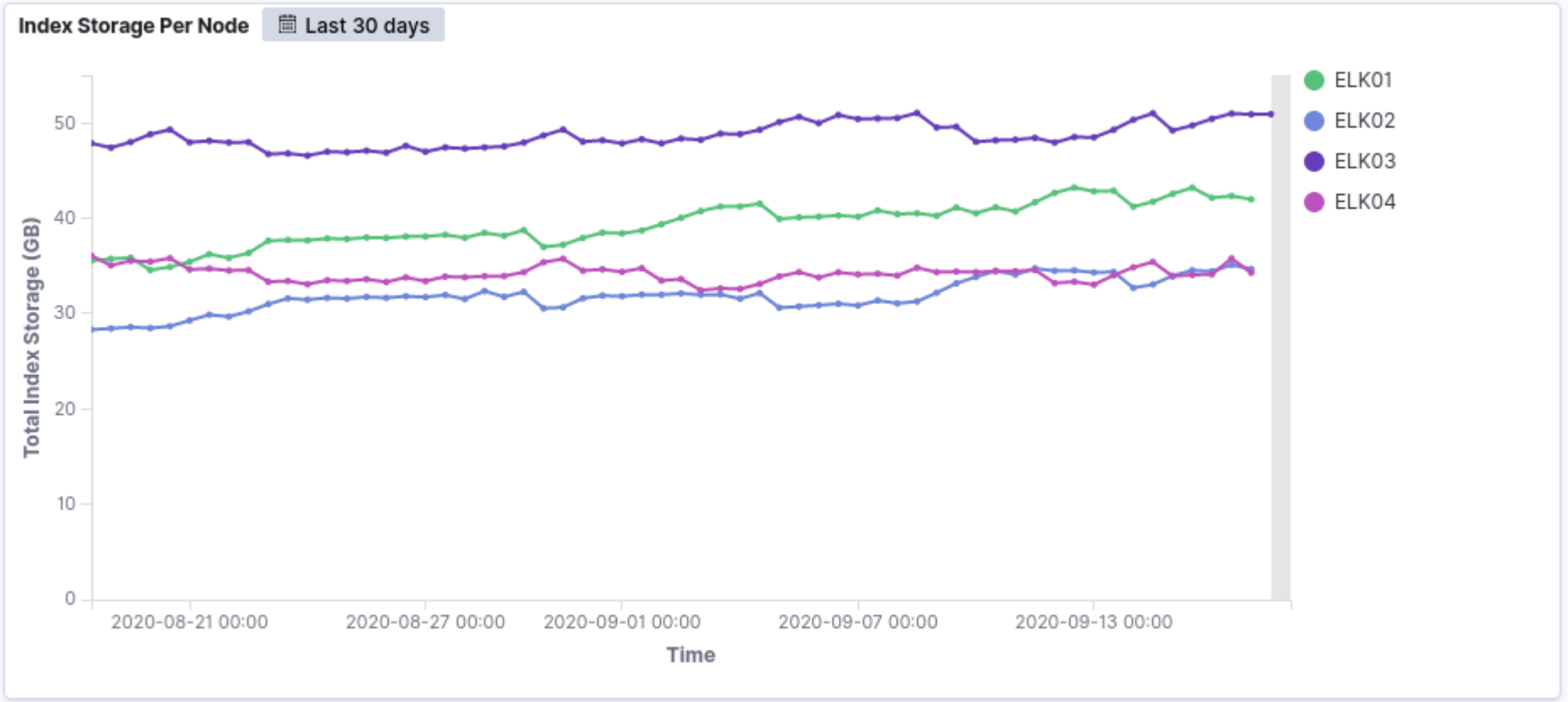
- - module: system
  - **period: 1h**
  - metricsets:
  - - cpu
  - - load
  - - memory
  - - network
  - - process
  - - process_summary
  - - socket_summary
  - - diskio
  - - uptime
  - - top 5 processes by cpu
  - - top 5 processes by memory

Index Lifecycle Management

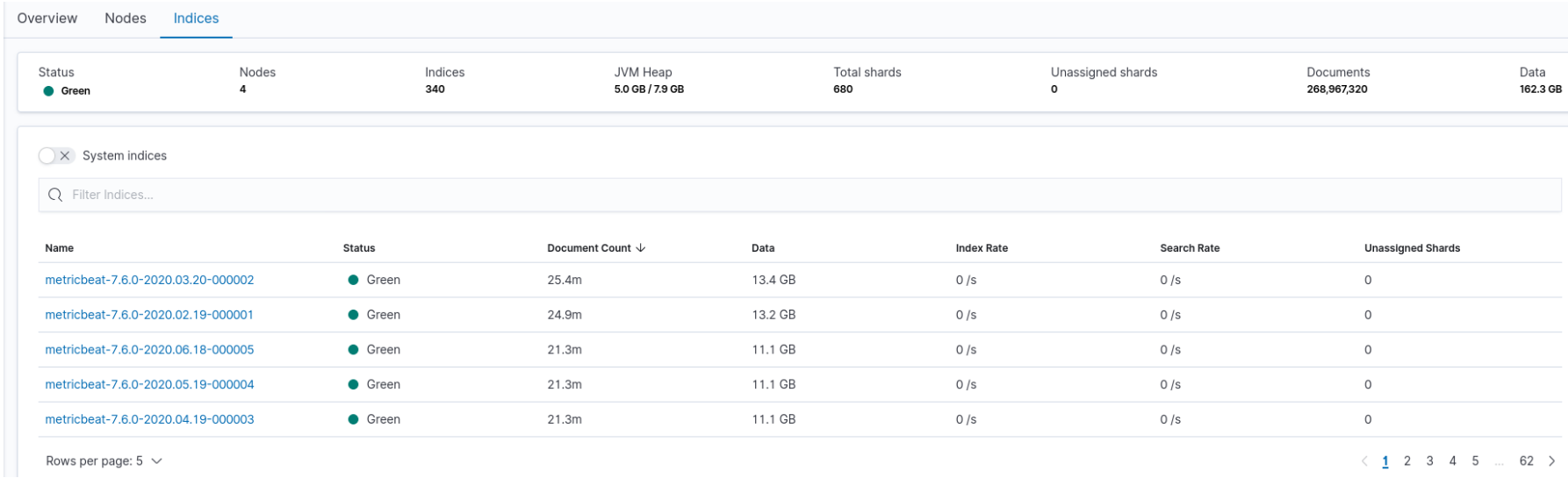
- Optimizing when to rollover indexes
- Previous policy for some indexes (nextcloud, apache) was set to a daily rollover which could be a problem with many hot indexes and if query interest is broader (across months)
- Current lifecycle policies
  - 50GB or 30 days age whichever is soonest
  - OK for now but need to look into including
    - Document count limiter based on rate inspected
    - Put older indexes to cold phase (read-only since so far none of these documents or indices need to be updated once they're ingested, transformed into Elasticsearch)

Kibana Visualizations

Storage Use and Growth



Kibana - Index Mgmt



Kibana API console

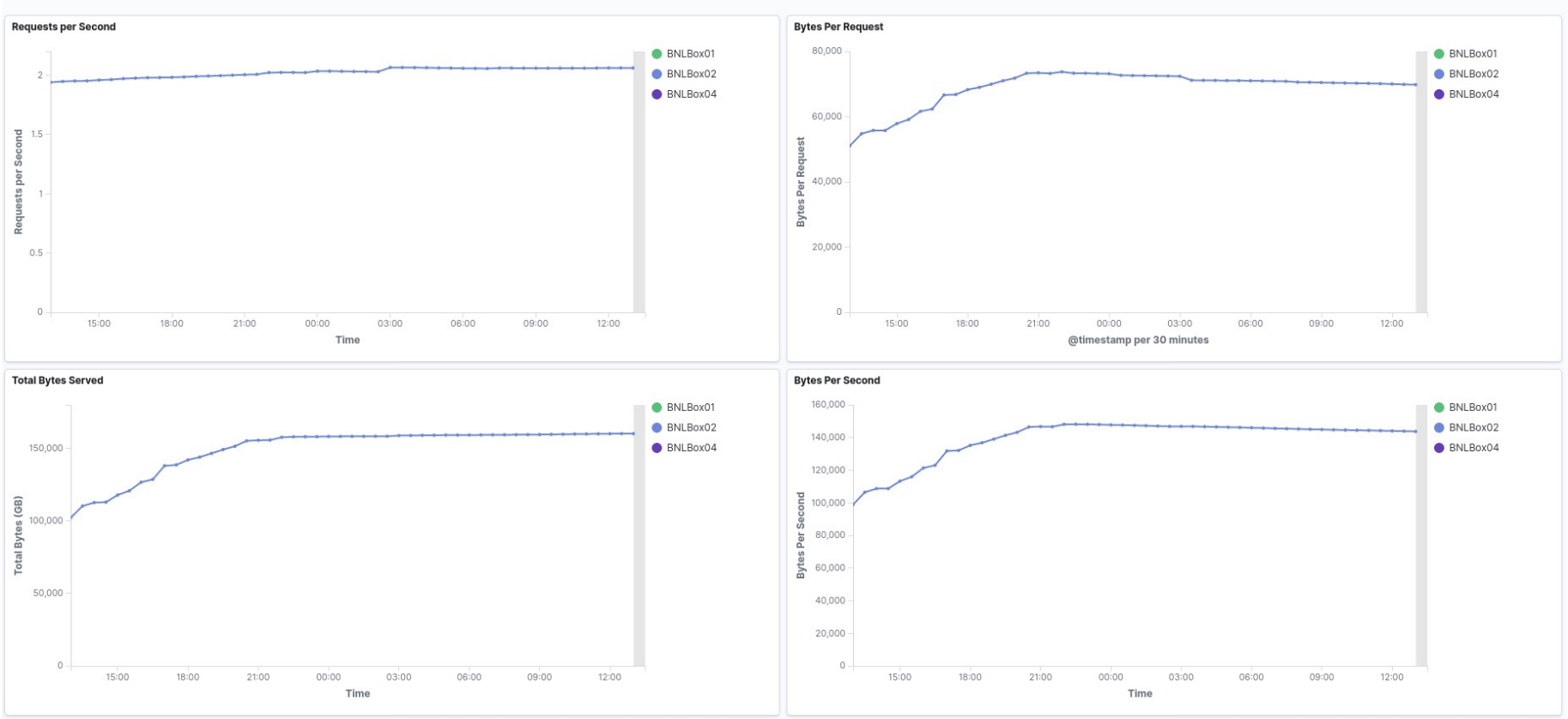
Console | Search Profiler | Grok Debugger

History | Settings | Help

```
GET /_cat/indices?v&human

GET /_cat/indices/metricbeat-7.6.0-2020.05.19-000004?v

GET /_cat/indices/metricbeat-7.6.0

POST /_analyze
{"text": "You'll love Elasticsearch 7.0."}

GET /_cat/indices/gridftp-test

PUT /gridftp-test/_mapping
{
  "properties": {
    "nbytes": {
      "type": "long"
    }
  }
}

GET /gridftp-test/_doc/uwShVXQB3A7iEXlyv7B0

GET /_cat/aliases?v

GET /_ilm/policy

GET /globus/_mapping

GET /_cat/shards

GET /globus/_mapping
```

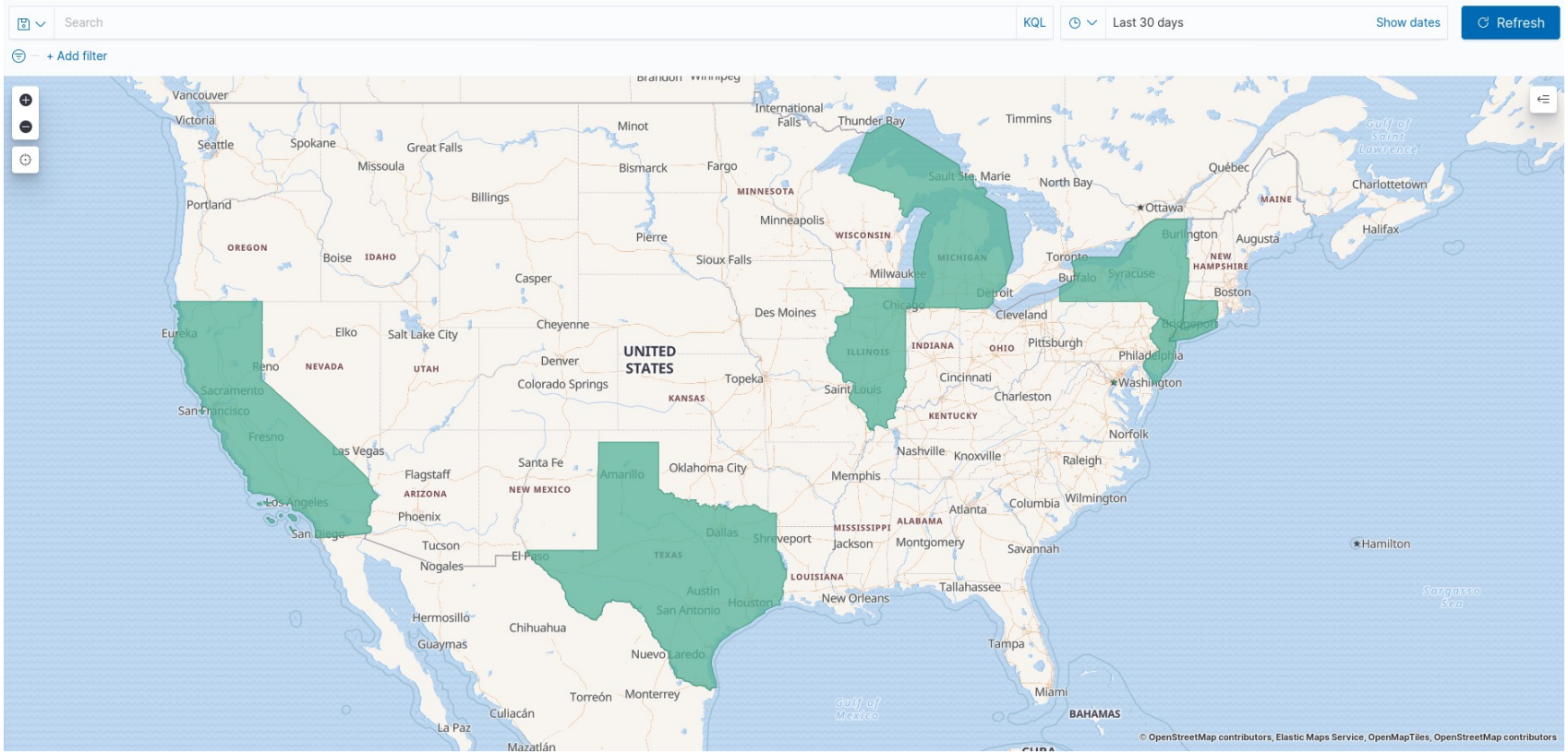
```
health status index                                uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   nextcloud-2020.06.29                 pojolXUBRdG7UrCApZO5mA   1   1       5418         5791     10.4mb          5.2mb
green  open   nextcloud-2020.06.23                 FgJb-7YeSmirBzaAVg5fmg   1   1      14051         1183     14.2mb          7.4mb
green  open   nextcloud-2020.06.24                 RyaNr-XWQc6R6j8-BPLXZg   1   1       4759         1249      6.8mb          3.3mb
green  open   nextcloud-2020.06.21                 K5cWu2IxSyaIaFQUSI2zZQ   1   1        564            0      2.5mb        438.5kb
green  open   nextcloud-2020.06.22                 VaHvSX9sTqq5p_5_NMdiuQ   1   1       4630         6620      9.3mb          4.8mb
green  open   nextcloud-2020.06.27                 GioQmp-TSjSbzJ4smQAKpw   1   1        924            0      1.1mb        616.1kb
green  open   nextcloud-2020.06.28                 m7cTa1y9Qqm_DOuMLRsAkg   1   1       1183            0      3.4mb        710.4kb
green  open   nextcloud-2020.06.25                 1xWf_N9YRyufFToPkpUB-Q   1   1       1620         3121      6.1mb          2.1mb
green  open   filebeat-7.6.1-2020.08.13-000006     9ANacb8rRsaaCZUa2-fu0w   1   1          0            0       566b           283b
green  open   nextcloud-2020.06.26                 -Rjcym6hT7a_L46E0T8hSQ   1   1       3439         5666      9.3mb          4.5mb
green  open   apache-ssl-2020.05.29                yeR1QNZATV-vDFh8IfqcbQ   1   1     162880            0      183mb         91.4mb
green  open   metricbeat-7.6.0-2020.02.19-000001   0Ub-LcbdRAOZ0LwpH9TVKA   1   1   24907963            0     13.2gb          6.6gb
green  open   apache-ssl-2020.05.27                R314J-UMSl-bWtYK3bsnyw   1   1     138887            0     159.4mb         79.5mb
green  open   apache-ssl-2020.05.28                cp1MuygFS2WSAnXg-hyrsQ   1   1     164671            0     174.3mb         87.1mb
green  open   apache-ssl-2020.05.25                3cELQtc4Qh-DrbLQkCtHkA   1   1      90030            0     104.7mb         52.3mb
green  open   nextcloud-2020.06.30                 ZRSEG0SBSHmuQeKdvqgdJQ   1   1       6603         6603      12.1mb            6mb
green  open   apache-ssl-2020.05.26                FHqR2osdT0qays7--prjIg   1   1     137512            0     151.4mb         76.6mb
green  open   apache-ssl-2020.05.23                FF1VgfAhR4WnsSkHj2Kpkg   1   1      75641            0      81.8mb         40.9mb
green  open   apache-ssl-2020.05.24                AP6XoPomQ3mf2DBGCHpHng   1   1      86008            0      99.5mb         49.7mb
green  open   apache-ssl-2020.05.21                WTqJhAERQYmejIXgbaKkmw   1   1     186894            0     194.4mb         97.2mb
green  open   apache-ssl-2020.05.22                aVfHUs2ATLyW3GT1SXuSJQ   1   1     155398            0     180.3mb         90.1mb
green  open   apache-ssl-2020.05.20                hsaxhfuCTzOq9Hf_tX8L6w   1   1     158574            0     177.2mb         89.2mb
green  open   nextcloud-2020.07.09                 XwPt5kh9RIuMdlURXAGLZA   1   1       2662         6666      10.2mb          5.1mb
green  open   nextcloud-2020.07.08                 LAHWYm9TSbm5N6_lelDiRg   1   1       3194         5701       7.4mb          4.5mb
green  open   nextcloud-2020.07.03                 WUQpkBT0R5mQCeGGFxQ1AA   1   1       1305            0      4.6mb        899.7kb
green  open   nextcloud-2020.07.02                 ELapii0VQJqC3EcRXut2hQ   1   1      22448         5557     21.3mb         11.2mb
green  open   nextcloud-2020.07.01                 blo958tLRAiB2zfAohjEnA   1   1      12594            0     13.9mb          5.6mb
green  open   nextcloud-2020.07.07                 2-58fyIIQeWfkQdXvuND2Q   1   1       4295         2020      8.3mb            3mb
green  open   nextcloud-2020.07.06                 nKghTDRoQQGTMJMTHdYs9g   1   1       6747            1      6.1mb            3mb
green  open   nextcloud-2020.07.05                 RPb7s9YfQFOq0kWxszd6KA   1   1       3152         2185      5.9mb          2.5mb
green  open   nextcloud-2020.07.04                 3kbGt-ufTo2rbtHGkqWrtw   1   1       2549         2751      6.4mb          2.5mb
green  open   .monitoring-es-7-2020.09.11          1Ph7_LD-RE6KBQamTBBr8g   1   1    3016666       590710      3.1gb          1.5gb
green  open   apache-ssl-2020.06.08                Ax85Sa1lRxuE9v4IEijNOQ   1   1     100825            0     116.4mb         58.2mb
green  open   connection-2020.08.20                qgipFNSqTC-ABho9kioDsA   1   1         23            0     142.7kb           63kb
green  open   apache-ssl-2020.06.09                q0ykZsUKTCqgownob6eQBA   1   1     122355            0     135.5mb         67.7mb
green  open   .monitoring-es-7-2020.09.13          KRaBvViORNqHOBjaeuX0Eg   1   1    3058818       260852      3.1gb          1.5gb
green  open   apache-ssl-2020.06.06                VuLmEOnHSoChMEYNft5ong   1   1     110142            0     121.9mb         60.8mb
```

BNLBox apache dashboard

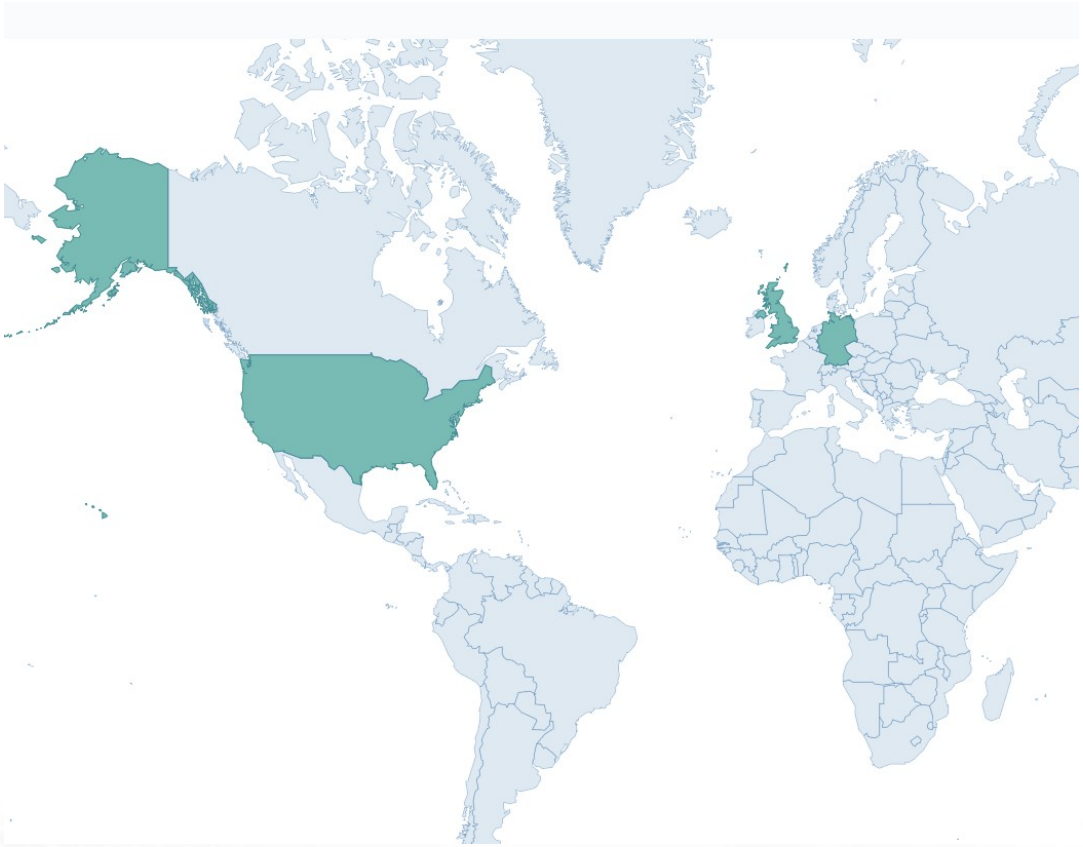


globus01 transfer size
(incl. tag cloud top 5 users by nbytes)

globus01 transfer regional destinations (todo: include geo_point heat map layer)

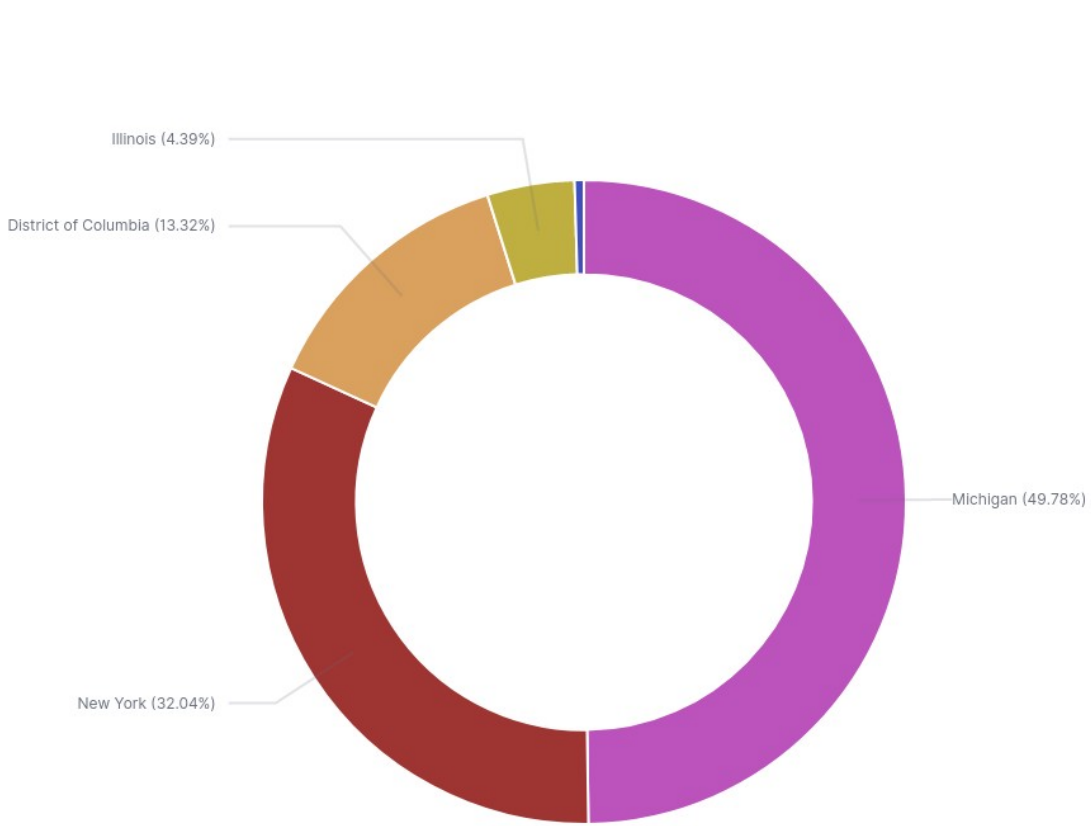


globus01 geoip transfer destination countries

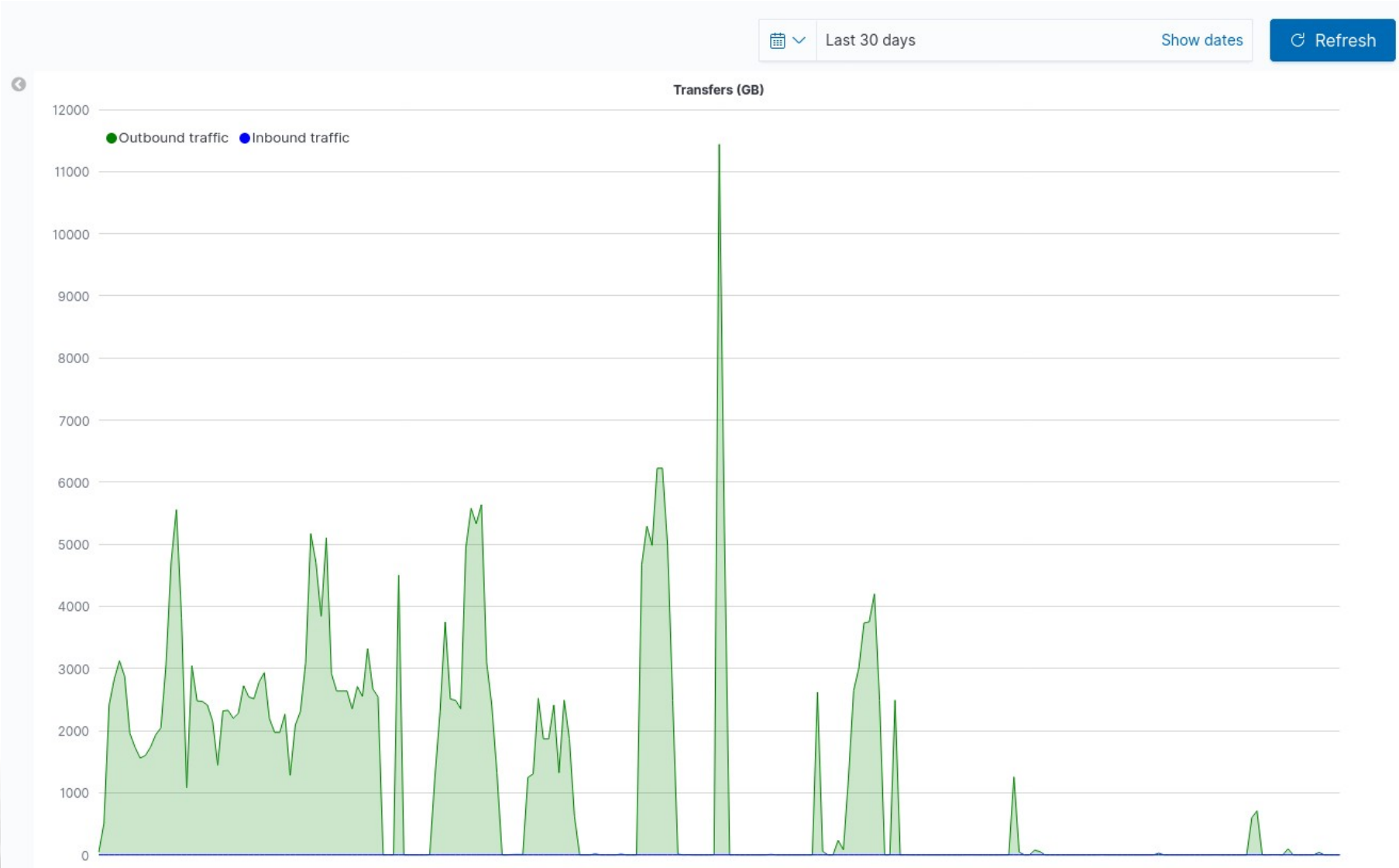


globus01 transfer distribution by US State

KQL | Last 30 days | Show dates | Refresh



globus01 transfer i/o



Summary and Future plans

- Showed current implementation of Elastic monitoring BNLBox and globus
- Need a better understanding around index lifecycle management and optimizations of shard replication across nodes
- At one point metricbeats accounted for ~75% of index storage and document count; reduced cadence; nagios monitoring probably makes this unnecessary
- More insights into globus transfers from raw logs
- Where are the “hot” and “cold” files? We now have a history of interactions by file in BNLBox and our Globus Connect Server
- Current vision is no need for disparate elastic systems since we can horizontally scale according to a specific index/shard (log ingestion)
- Combine all on 1 cluster “pool” of new storage and tune appropriately

Thanks for your time!

Questions?