

Elastic Stack - SDCC

Matt Snyder
Brookhaven National Laboratory
msnyder@bnl.gov

SDCC Liaison Meeting
8 October 2020



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 \leq Shards \leq 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 \leq Shards \leq Documents
 - Document
 - Flat, independent, self-containing thing (perhaps event, metric, blog post)
 - Self-containing meaning it is not 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, extract and enrich later

Node Configurations

- ELK cluster (“sdcc-test”) 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 view
- BNLBox
 - Nextcloud log index
 - Apache log index
 - 3 database states (count of users, total storage, storage by user)
- Globus
 - Globus log
- “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 → user_data.conf

```
[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

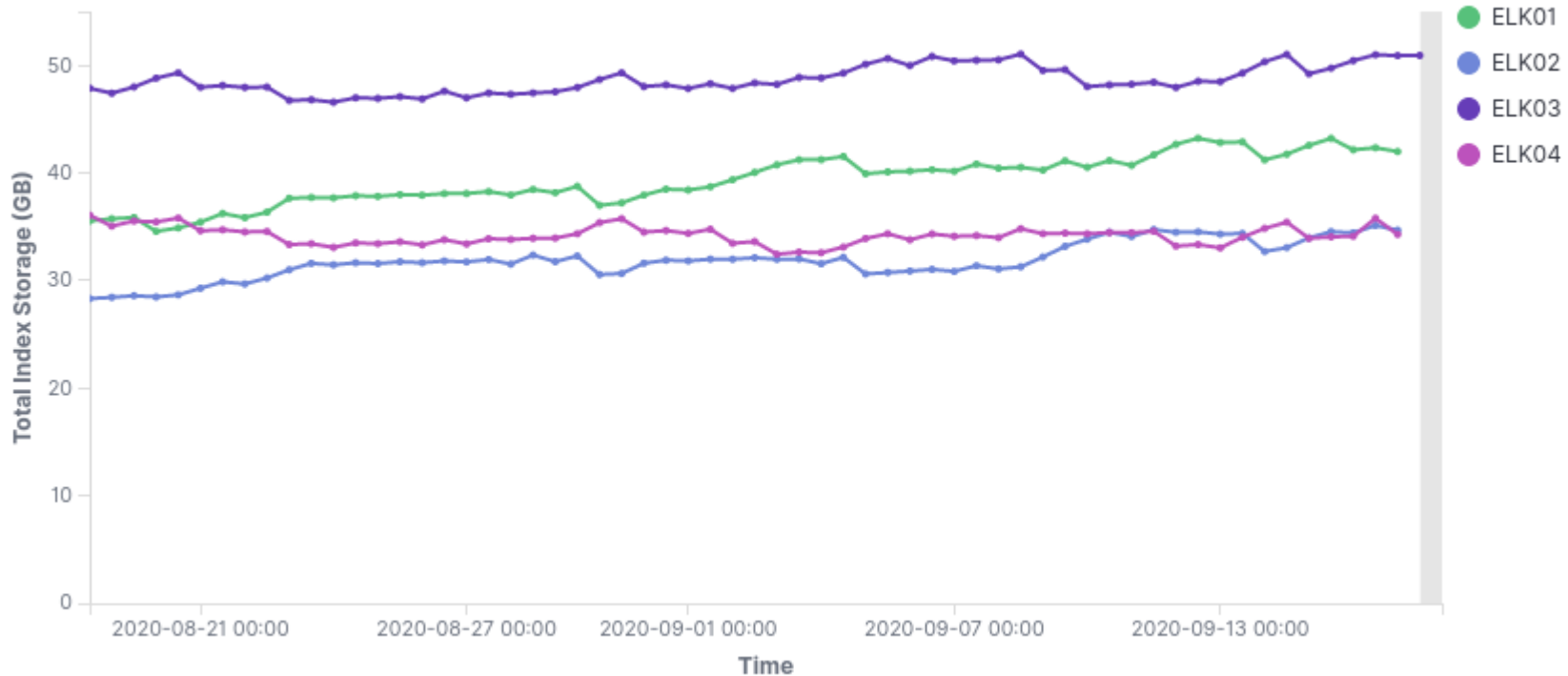
Index Lifecycle Management

- 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)

Storage Use and Growth

Index Storage Per Node

📅 Last 30 days



Summary and Future plans

- Showed current sdcc-test ELK cluster
- Implementations of bnlbox and globus log monitoring on ELK stack
- 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?
- 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 of new storage and tune appropriately

Thanks for your time!

Questions?

Extra Slides

Kibana - Index Mgmt

OverviewNodesIndices

Status

Green

Nodes

4

Indices

340

JVM Heap

5.0 GB / 7.9 GB

Total shards

680

Unassigned shards

0

Documents

268,967,320

Data

162.3 GB

×

System indices

Filter Indices...

Name	Status	Document Count ↓	Data	Index Rate	Search Rate	Unassigned Shards
metricbeat-7.6.0-2020.03.20-000002	Green	25.4m	13.4 GB	0 /s	0 /s	0
metricbeat-7.6.0-2020.02.19-000001	Green	24.9m	13.2 GB	0 /s	0 /s	0
metricbeat-7.6.0-2020.06.18-000005	Green	21.3m	11.1 GB	0 /s	0 /s	0
metricbeat-7.6.0-2020.05.19-000004	Green	21.3m	11.1 GB	0 /s	0 /s	0
metricbeat-7.6.0-2020.04.19-000003	Green	21.3m	11.1 GB	0 /s	0 /s	0

Rows per page: 5

< 1 2 3 4 5 ... 62 >

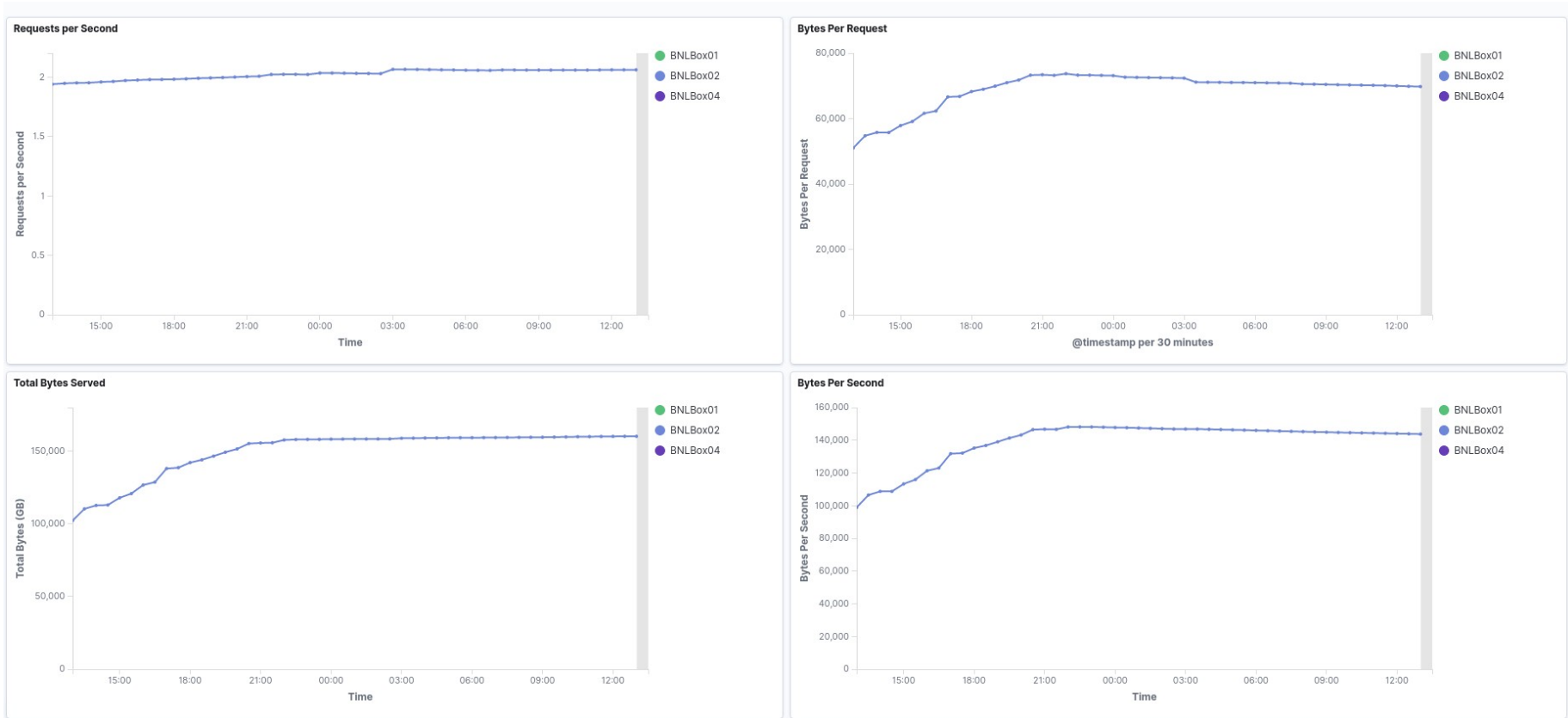
Kibana API console

Console Search Profiler Grok Debugger

History Settings Help

1 GET /_cat/indices?v&human	1 health status index	uuid	pri rep docs.count docs.deleted store.size pri.store.size
2 GET /_cat/indices/metricbeat-7.6.0-2020.05.19-0000047v	2 green open nextcloud-2020.06.29	pojoLXUBRdG7UrcApZ05mA	1 1 5418 5791 10.4mb 5.2mb
3 GET /_cat/indices/metricbeat-7.6.0	3 green open nextcloud-2020.06.23	FgJb-7YeSmirBzaAVg5fmg	1 1 14051 1183 14.2mb 7.4mb
4 GET /_cat/indices/metricbeat-7.6.0	4 green open nextcloud-2020.06.24	RyaNr-XWQc6R6j8-BPLXZg	1 1 4759 1249 6.8mb 3.3mb
5 POST /analyze	5 green open nextcloud-2020.06.21	K5cWu2xSyaIaFQUSI2zZQ	1 1 564 0 2.5mb 438.5kb
6 {"text": "You'll love Elasticsearch 7.0."}	6 green open nextcloud-2020.06.22	VaHvSX9sTqg5p 5 NMdiuQ	1 1 4630 6620 9.3mb 4.8mb
7 GET /_cat/indices/gridftp-test	7 green open nextcloud-2020.06.27	GioQmp-TSjSbzJ4smQAKpw	1 1 924 0 1.1mb 616.1kb
8 PUT /gridftp-test/_mapping	8 green open nextcloud-2020.06.28	m7cTaly9Qqm DOuMLRsAkq	1 1 1183 0 3.4mb 710.4kb
9 {"properties": {	9 green open nextcloud-2020.06.25	1xWf_N9YryuFfToPkpuB-Q	1 1 1620 3121 6.1mb 2.1mb
10 "nbytes": {	10 green open filebeat-7.6.1-2020.08.13-000006	9ANacB8rRsaaCZUa2-fu0w	1 1 0 0 566b 283b
11 "type": "long"	11 green open nextcloud-2020.06.26	-RjcyM6hT7a L46E0T8hSQ	1 1 3439 5666 9.3mb 4.5mb
12 }	12 green open apache-ssl-2020.05.29	yeR1QNZATV-vDFh8IfqcbQ	1 1 162880 0 183mb 91.4mb
13 }	13 green open metricbeat-7.6.0-2020.02.19-000001	0Ub-LcBdRA0Z0LwPH9TVKA	1 1 24907963 0 13.2gb 6.6gb
14 }	14 green open apache-ssl-2020.05.27	R314J-UM5L-bWTYK3bsnyw	1 1 138887 0 159.4mb 79.5mb
15 }	15 green open apache-ssl-2020.05.28	cp1MuygFS2wSAnXg-hyrsQ	1 1 164671 0 174.3mb 87.1mb
16 }	16 green open apache-ssl-2020.05.25	3cELQtC4Qh-DrbLQkctHKA	1 1 90030 0 104.7mb 52.3mb
17 }	17 green open nextcloud-2020.06.30	ZRSE0SBSHmuQeKdvggdJk	1 1 6603 6603 12.1mb 6mb
18 }	18 green open apache-ssl-2020.05.26	FHQzsdT0qays7--prjIg	1 1 137512 0 151.4mb 76.6mb
19 }	19 green open apache-ssl-2020.05.23	FF1VgfAhR4WnsSkHj2Kpkg	1 1 75641 0 81.8mb 40.9mb
20 }	20 green open apache-ssl-2020.05.24	AP6XoPomQ3mf2DBGChpHng	1 1 86008 0 99.5mb 49.7mb
21 GET /gridftp-test/_doc/uwShVXQB3A7iEXlyv7B0	21 green open apache-ssl-2020.05.21	WTqJhAERQYmejIXgbaKkmw	1 1 186894 0 194.4mb 97.2mb
22 GET /_cat/aliases?v	22 green open apache-ssl-2020.05.22	aVfHus2ATLyw3GT1Sxu5JQ	1 1 155398 0 180.3mb 90.1mb
23 GET /_ilm/policy	23 green open apache-ssl-2020.05.20	hsaxhfucT20q9Hf tX8L6w	1 1 158574 0 177.2mb 89.2mb
24 GET /_cat/shards	24 green open nextcloud-2020.07.09	XwPt5kh9RIuMdLURXAGLZA	1 1 2662 6666 10.2mb 5.1mb
25 GET /globus/_mapping	25 green open nextcloud-2020.07.08	LAHWYm9TSbm5N6 leLDiRg	1 1 3194 5701 7.4mb 4.5mb
26 GET /_cat/shards	26 green open nextcloud-2020.07.03	WUOpkBT0R5mQCcGGfXQ1AA	1 1 1305 0 4.6mb 899.7kb
27 GET /_cat/shards	27 green open nextcloud-2020.07.02	ELapii0VQJqC3EcRxt2hQ	1 1 22448 5557 21.3mb 11.2mb
28 GET /_cat/shards	28 green open nextcloud-2020.07.01	blo958tLRAiB2zfAohjEnA	1 1 12594 0 13.9mb 5.6mb
29 GET /_cat/shards	29 green open nextcloud-2020.07.07	2-58fyIIQewfkQdXvuND2Q	1 1 4295 2020 8.3mb 3mb
30 GET /_cat/shards	30 green open nextcloud-2020.07.06	nKghTDRoQQGTJMTJHdYs9g	1 1 6747 1 6.1mb 3mb
31 GET /_cat/shards	31 green open nextcloud-2020.07.05	RPb7s9YfQFQ0kKwszd6KA	1 1 3152 2185 5.9mb 2.5mb
32 GET /_cat/shards	32 green open nextcloud-2020.07.04	3kbGT-ufTo2rbtHGkgWrtw	1 1 2549 2751 6.4mb 2.5mb
33 GET /_cat/shards	33 green open .monitoring-es-7-2020.09.11	1Ph7 LD-RE6KB0amTBB8g	1 1 3016666 590710 3.1gb 1.5gb
	34 green open apache-ssl-2020.06.08	Ax85Sa1lRxuE9v4IEijNOQ	1 1 100825 0 116.4mb 58.2mb
	35 green open connection-2020.08.20	qgipFNSqTC-ABho9k1oDsA	1 1 23 0 142.7kb 63kb
	36 green open apache-ssl-2020.06.09	q0ykZsUKTCqgqwnob6eQBA	1 1 122355 0 135.5mb 67.7mb
	37 green open .monitoring-es-7-2020.09.13	KRA8V10RNqH0BJaeuX0Eg	1 1 3058818 260852 3.1gb 1.5gb
	38 green open apache-ssl-2020.06.06	VuME0uGcHMEVfif50g	1 1 110143 0 121.0mb 60.0mb

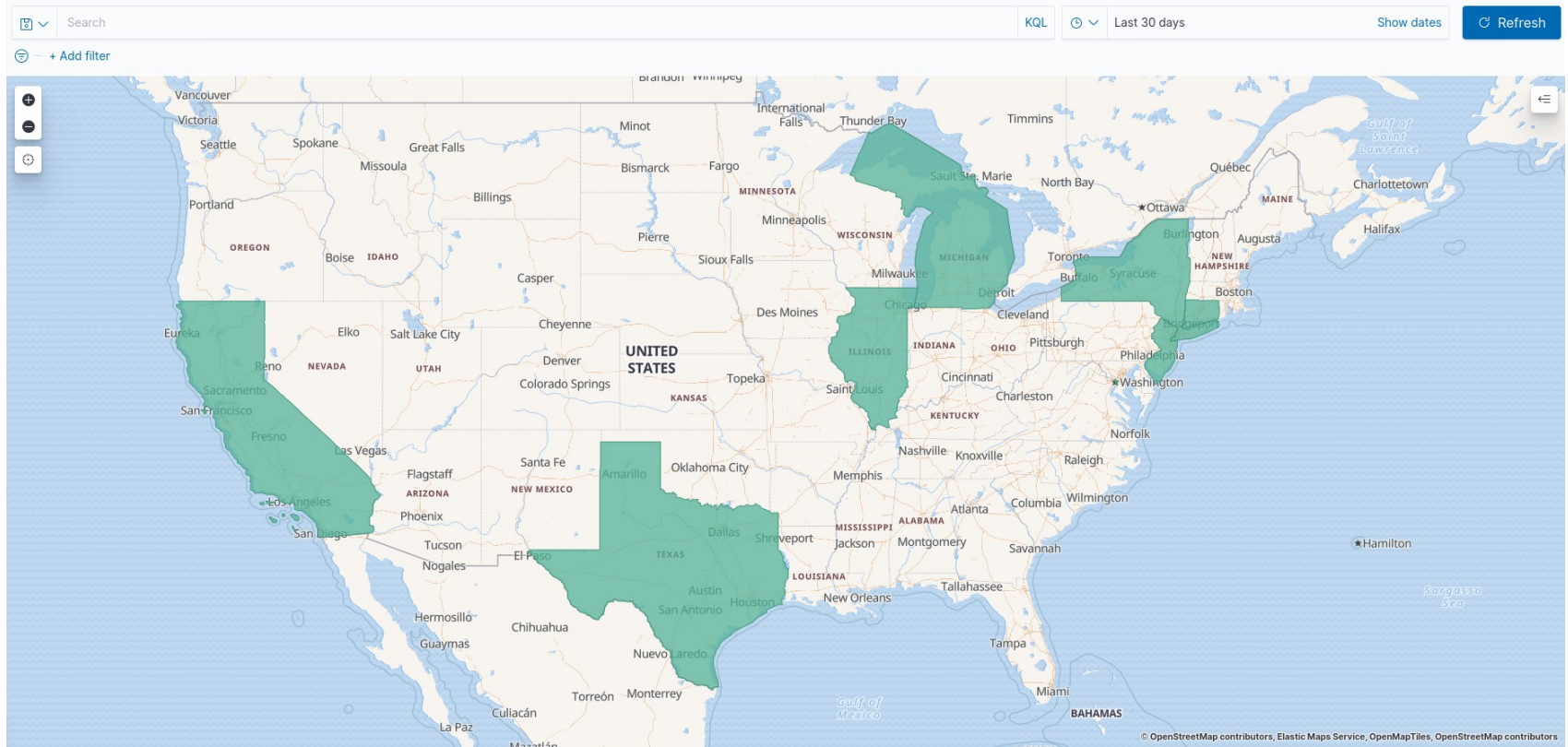
BNLBbox apache dashboard



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



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



**globus01 transfer world country destinations
(todo: include geo_point heat map layer)**

