


Grab ES Migration: From EC2 to AWS ES to ECE

Ben Wu Weibin
Grab, Database Engineer



战略级赞助商  HUAWEI

钻石级赞助商  普翔

白金级赞助商  华夏博格
— SINO BIG DATA —

 神州数码
Digital China

金牌级赞助商  iDataAPI
为大数据应用赋能

合作伙伴  开源中国
oschina.net

 掘金

 谱时

 IT大咖说
知识共享平台

 otpub
成为IT学习平台

 Broadview
www.broadview.com.cn

 百格活动
bagevent.com

 MAXHUB
高效会议平台

Grab ES Migration

From EC2 to AWS ES to ECE

Working in #1 Transport Startup



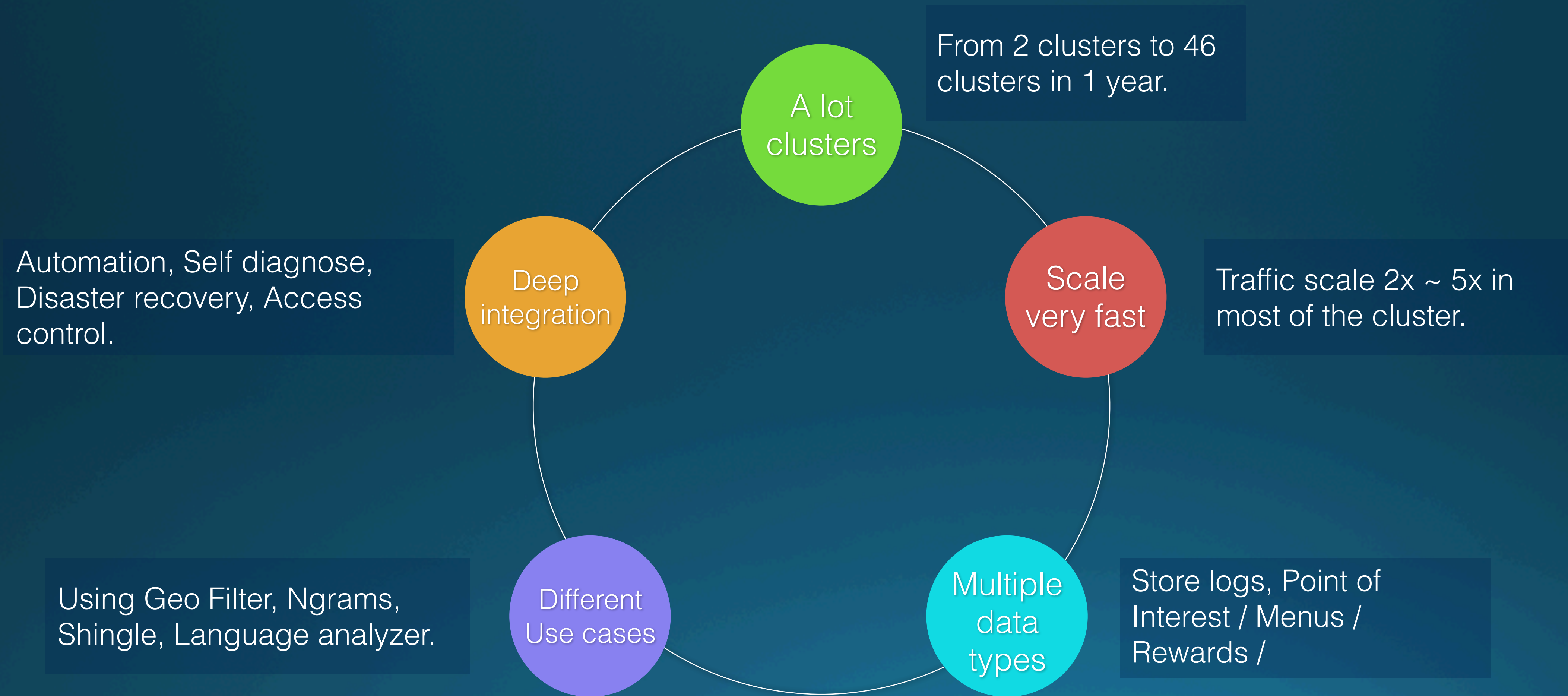
Weibin Wu — Ben

Who is Ben

Ben is working in Grab as a Database Engineer. When he joined Grab, he was expected to manage all the Elasticsearch cluster. Till today, the clusters number went from 2 to 30+. During the scale of Grab, our Elasticsearch cluster explored 3 different platforms. No one wants to migrate so many times for Elasticsearch. This story is about how the decision was made, and what's our experience with different platforms.



Elasticsearch in Grab



Our Problem

- Applying new EC2 need days, weeks to process.
- Changing instance type need a manual rolling upgrade. (Access needed)
- Not able to handle hardware problem automatically. (On-call nightmare)

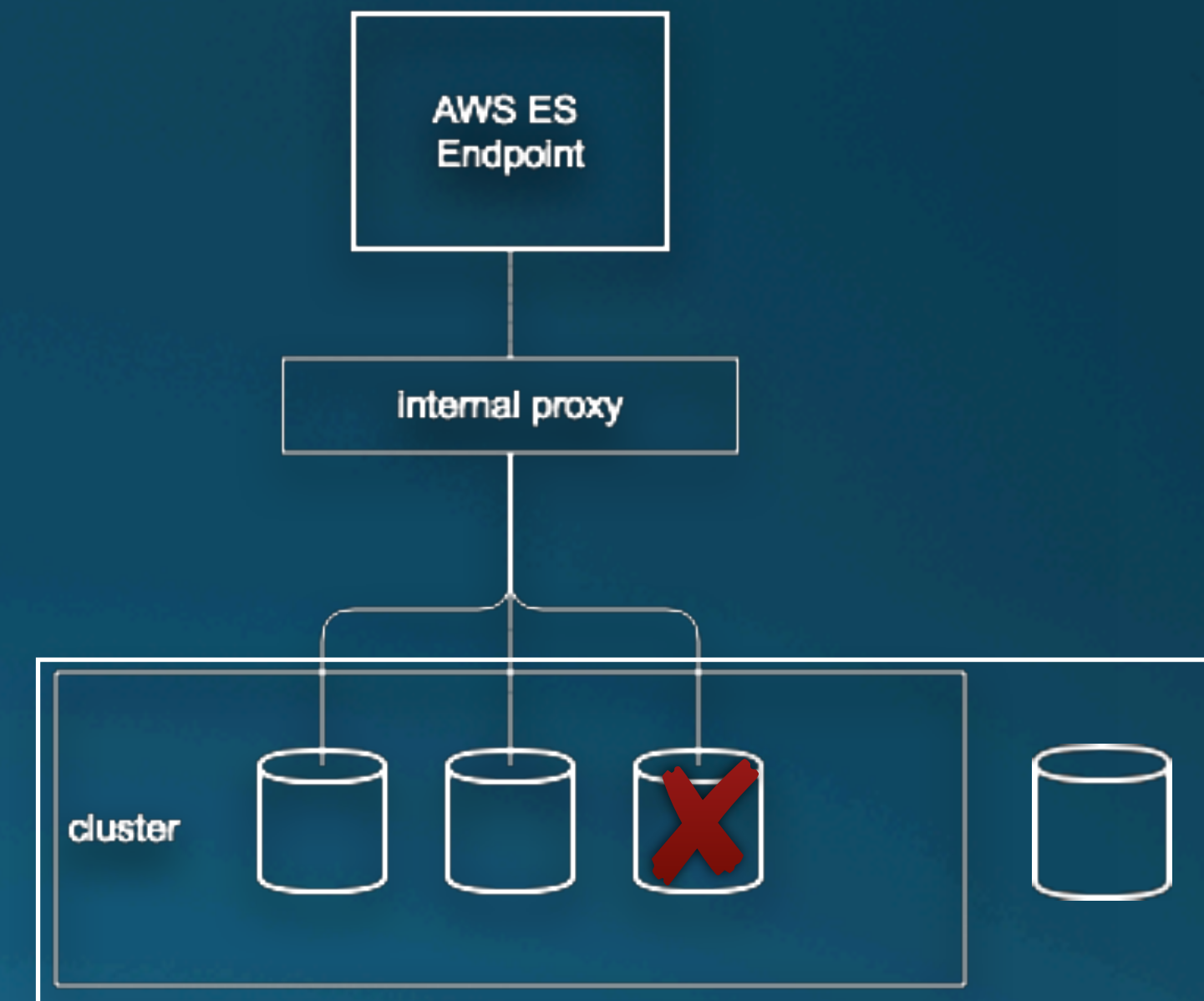
First Migration

From EC2 to AWS ES

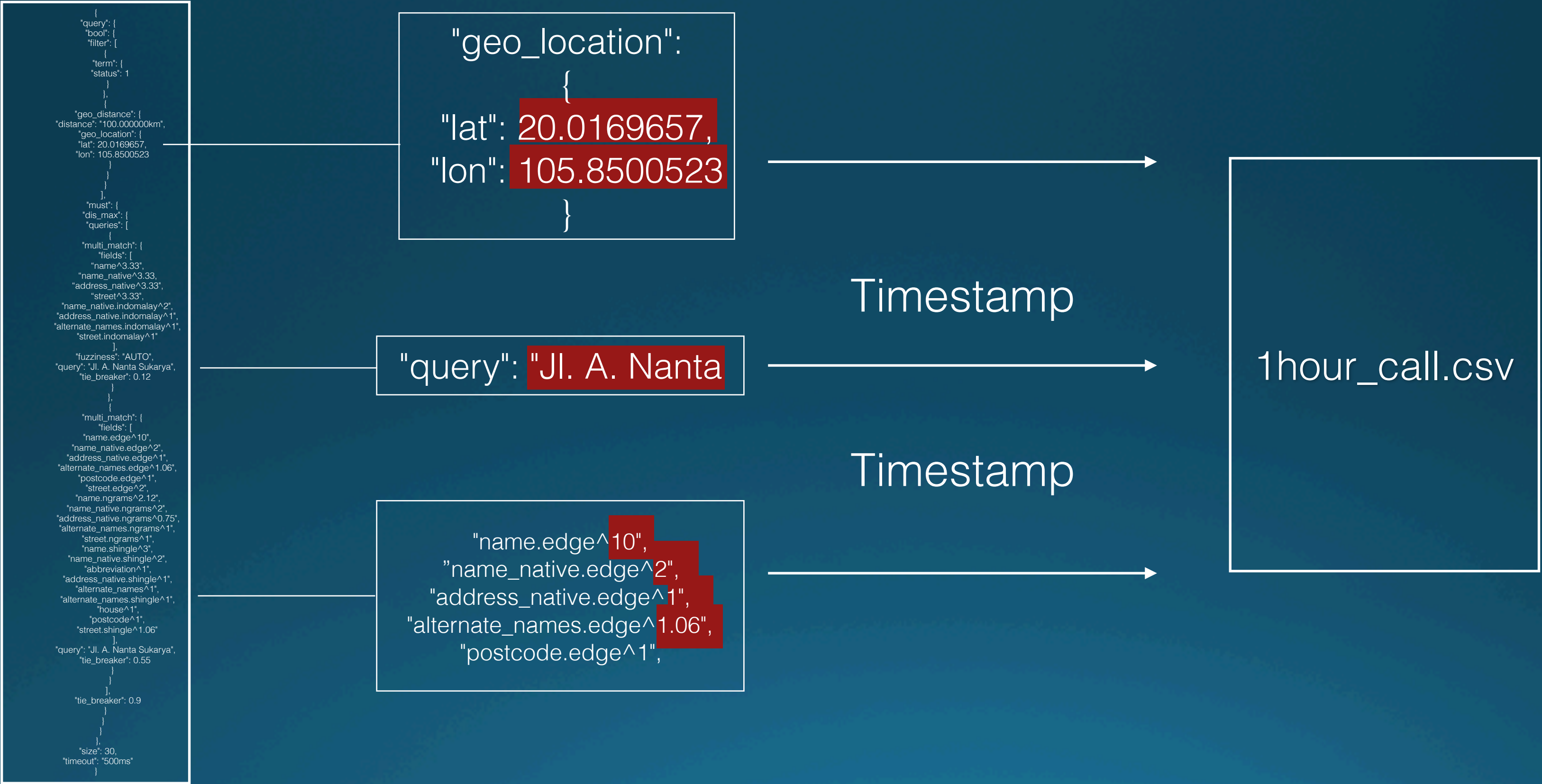
Control our cluster

Why AWS Elasticsearch

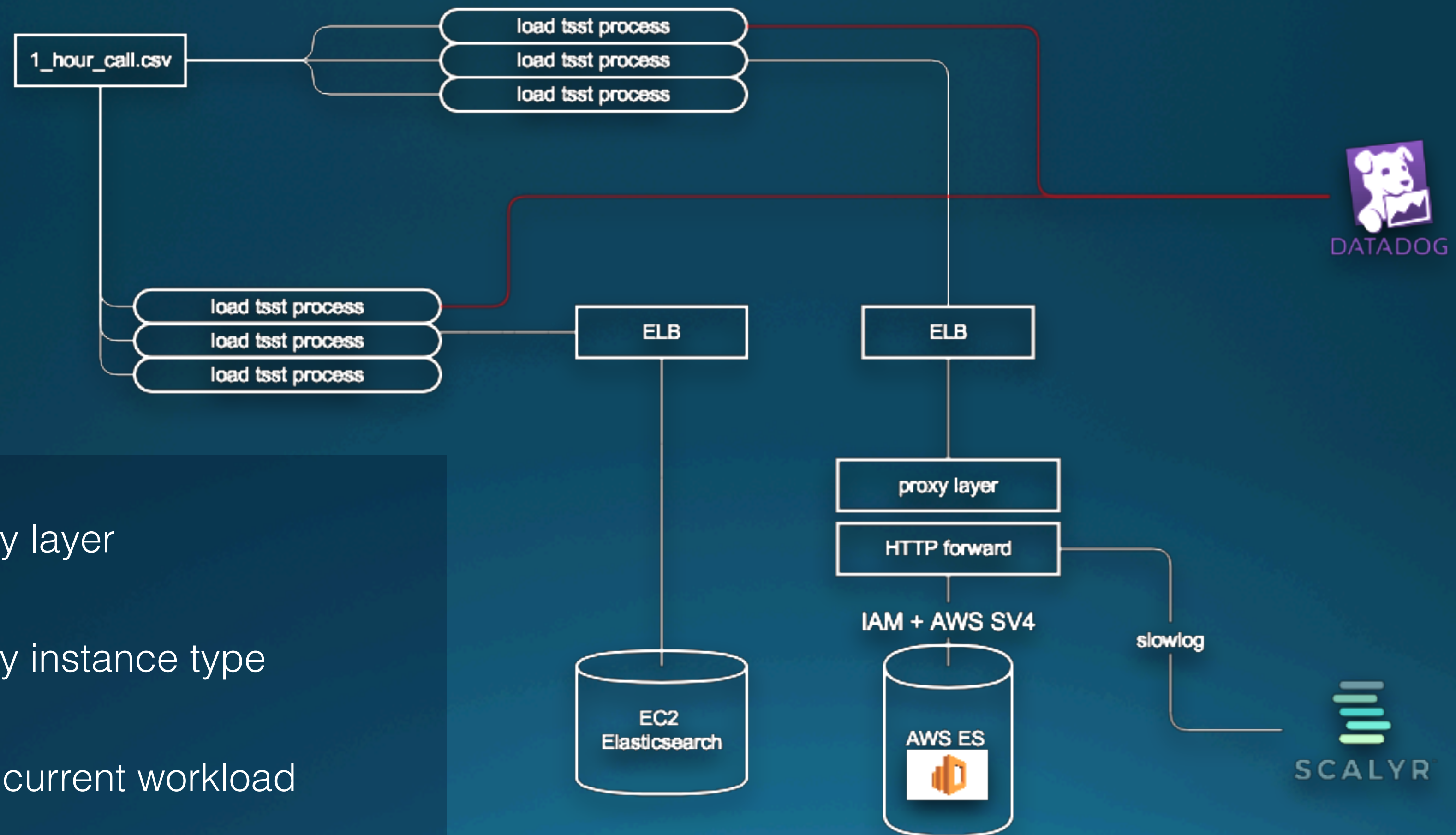
- Scale cluster up/out in around 30 min.
- AWS managed the infrastructure
- Auto node replacement



Migration Load Test



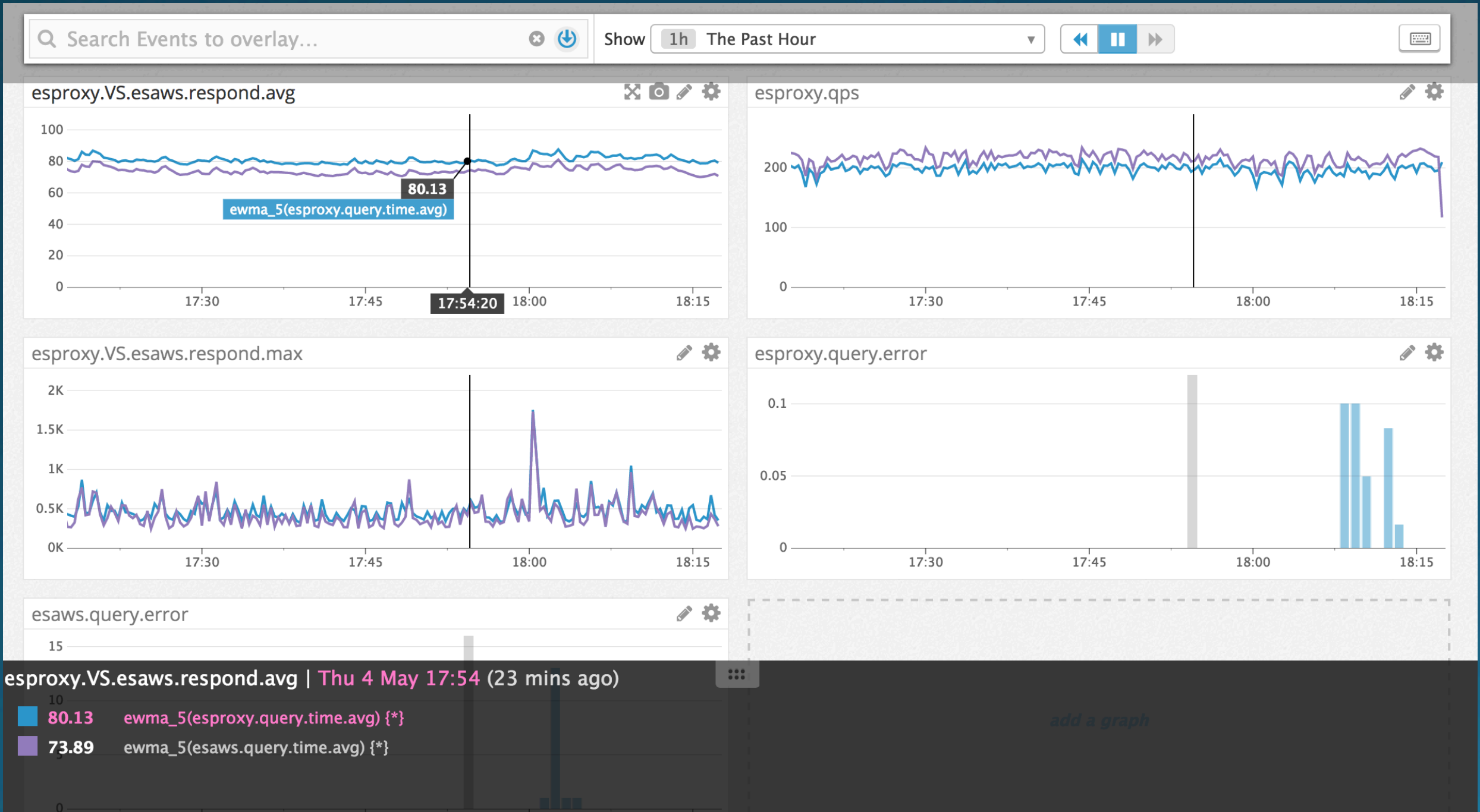
Migration Load Test



- Verify impact of proxy layer
- Verify impact of proxy instance type
- Verify CPU usage of current workload

Load Test Result

Infras	Load Test Processes	Proxy Instance Type	Avg Search Time	Proxy Cost
EC2 ES VS Proxy + AWS ES	16	c4.large	73 / 80 ms	10%
	35	c4.large	102 / 110 ms	10%
	35	c4.2xlarge	106 / 110 ms	5%



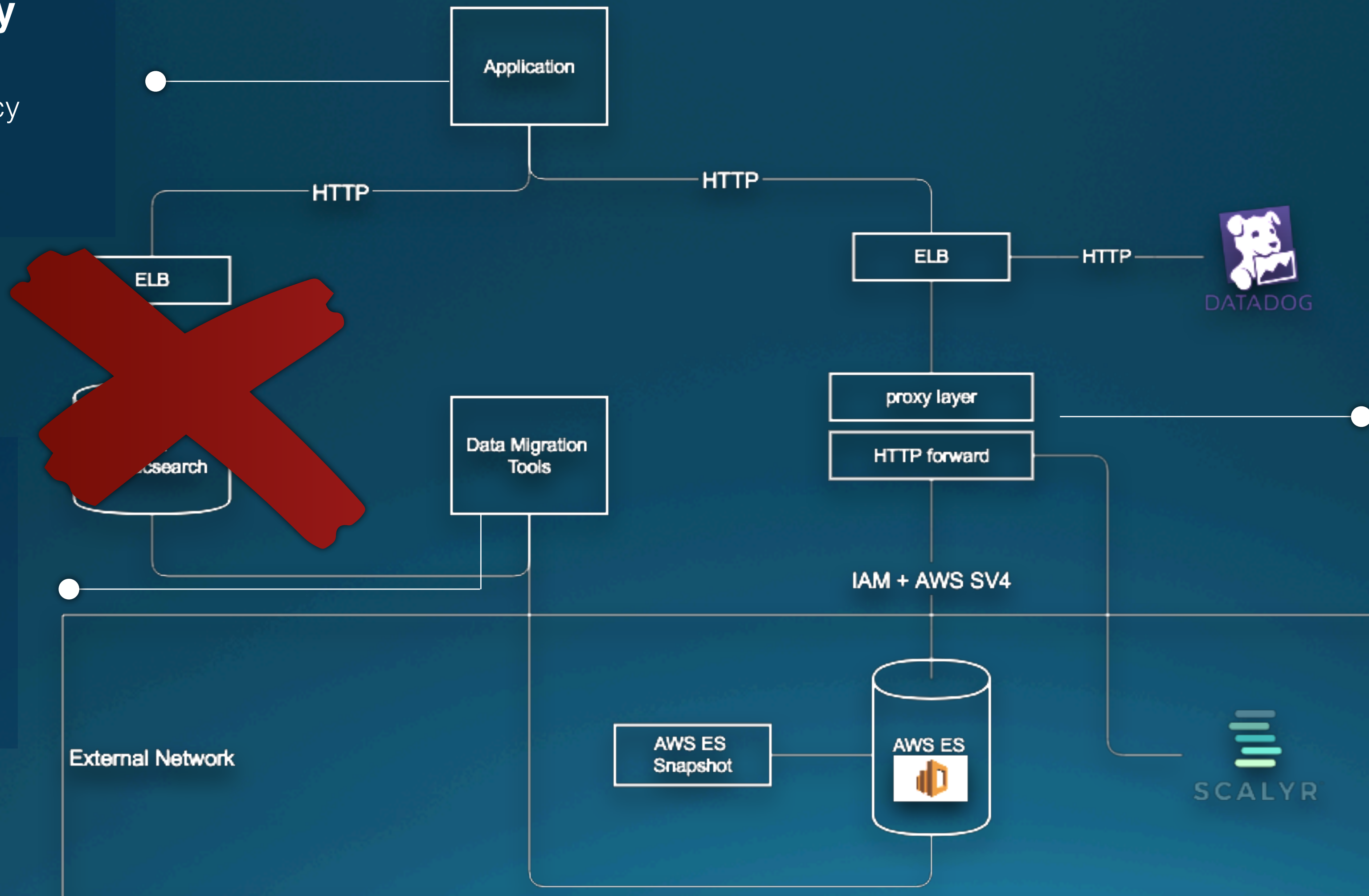
Migration Steps

Canary Deploy

Ensure query Latency

Migrate Data

Support multi platform and version

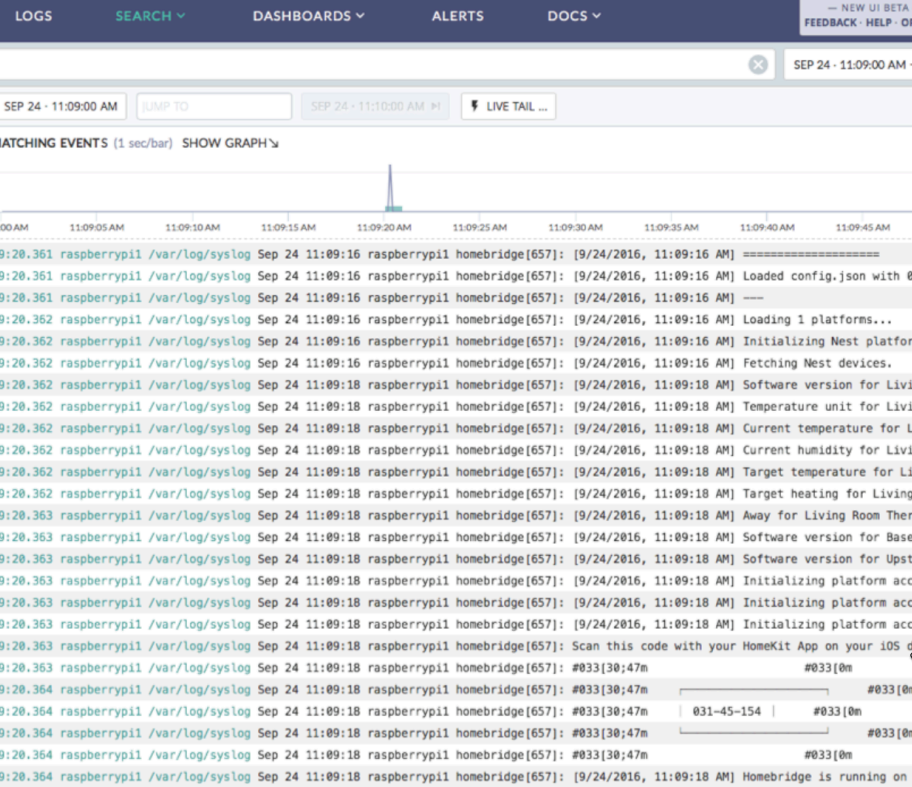
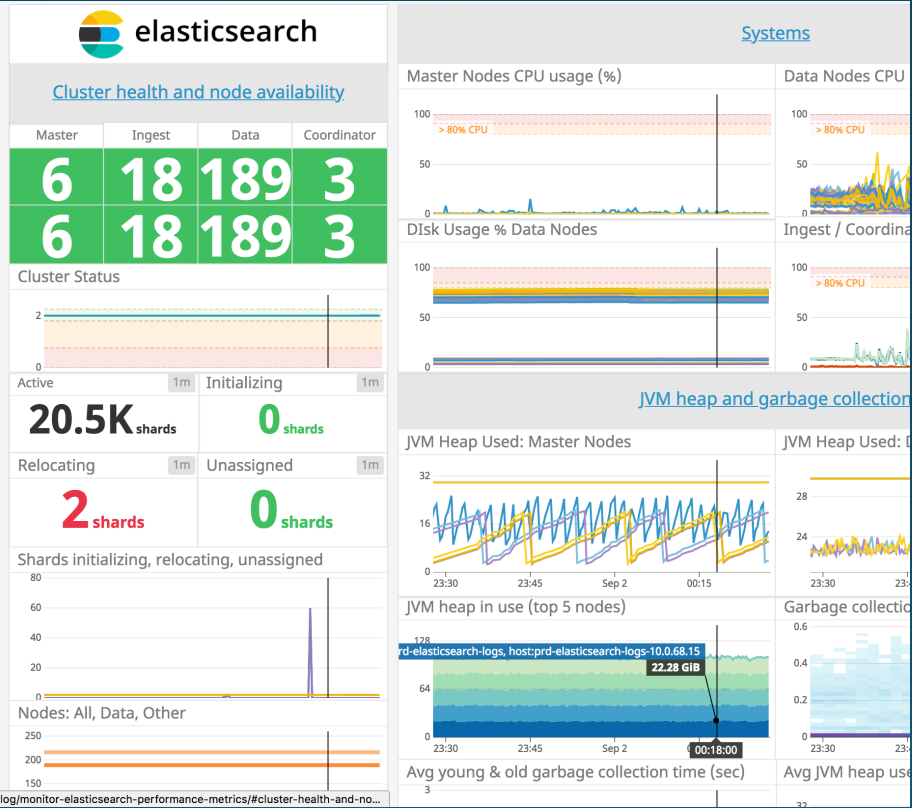
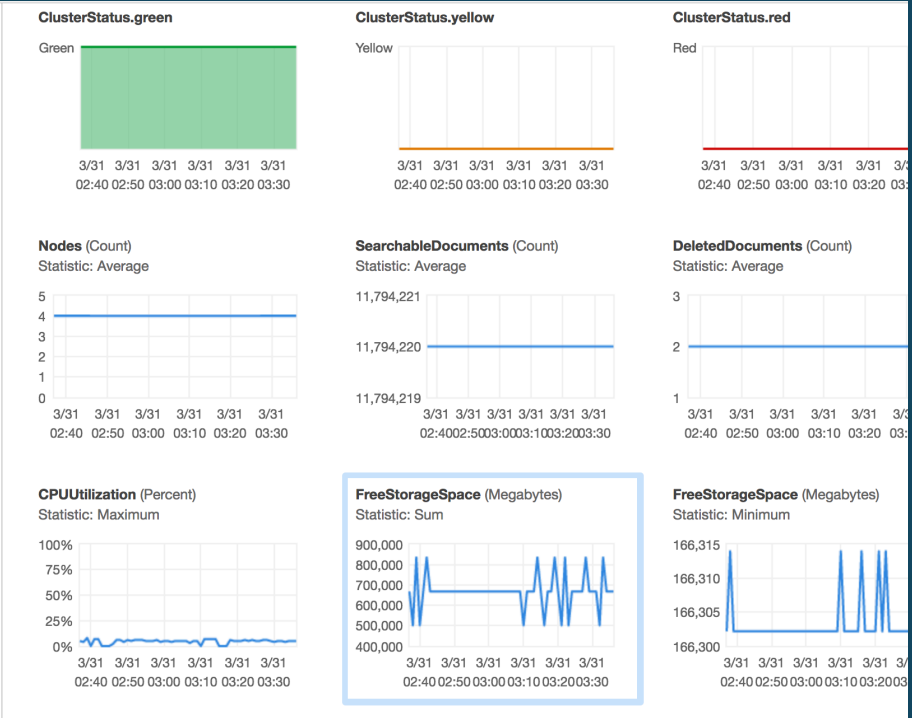


Golang Proxy

Ensure query Latency

Monitoring For AWS ES

AWS ES



CloudWatch

Slack alert

Network Monitoring

Share Monitoring with AWS

Datadog

Self defined metrics

Thread pool monitoring in sec.

Slack alert integration

Scalyr

All the query logs

Customized parser

When we moved fast

5 Months

- QPS double
- Data Volume doubled
- GEO search radius increase



3 Shards 1 Replica 3 Nodes



5 Shards 1 Replica 5 Nodes



5 Shards 3 Replica 10 Nodes



Things Change

Everything has a cost

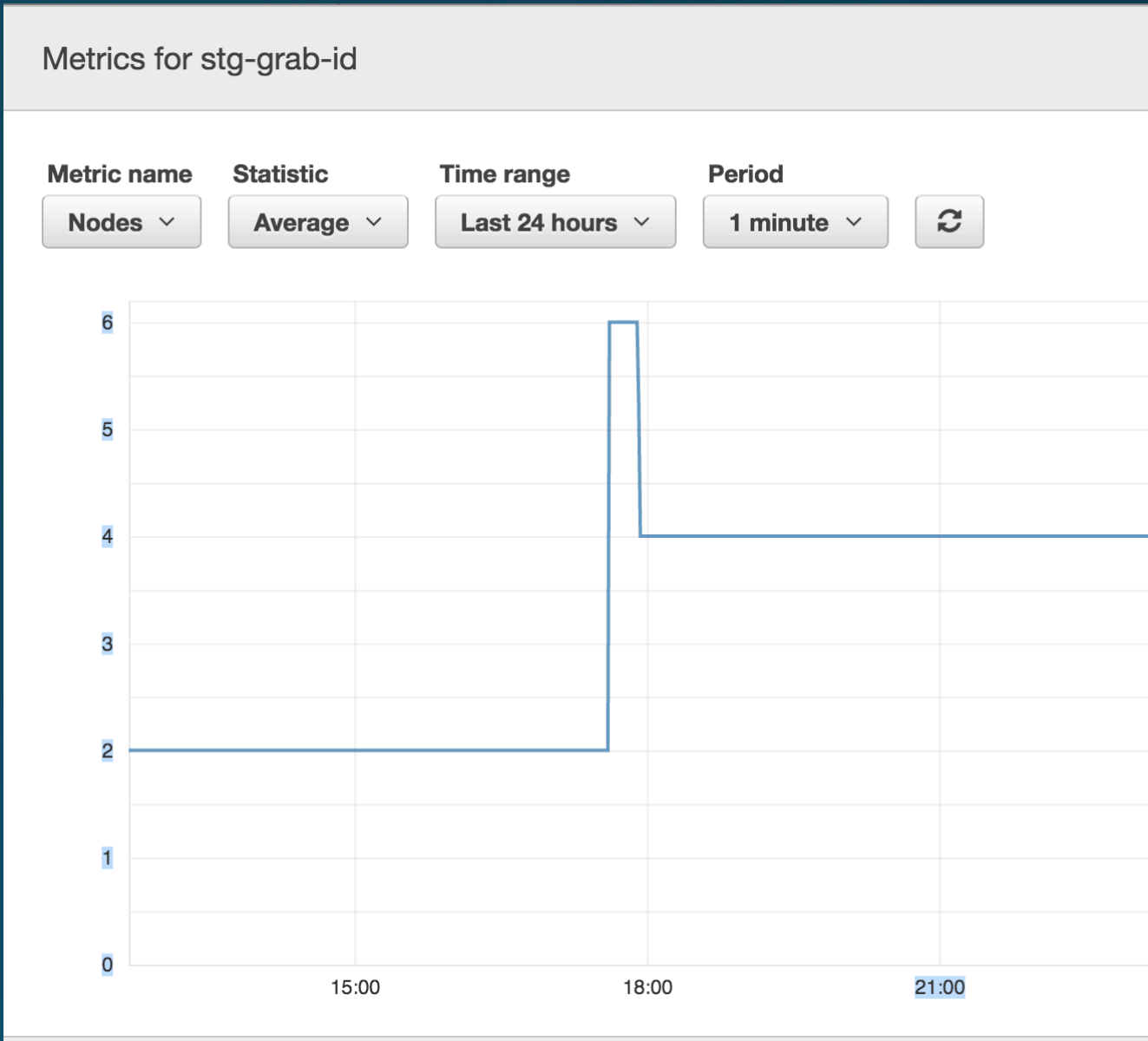
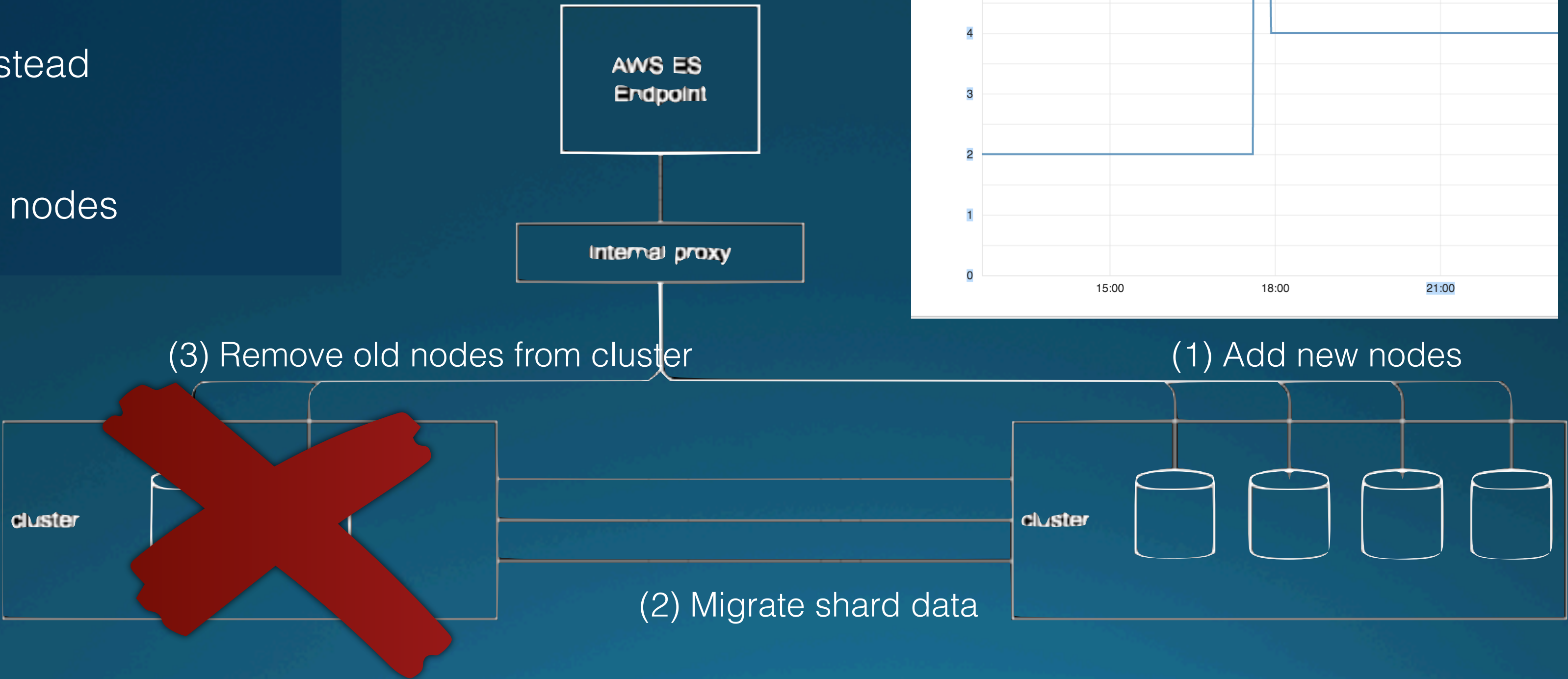
- Scalability has cost.
- We have to adapt to AWS security control.
- Managed service means less customization.
- Black box operation. Need to rely on AWS support.
- API call limitation from EC2 to AWS ES.



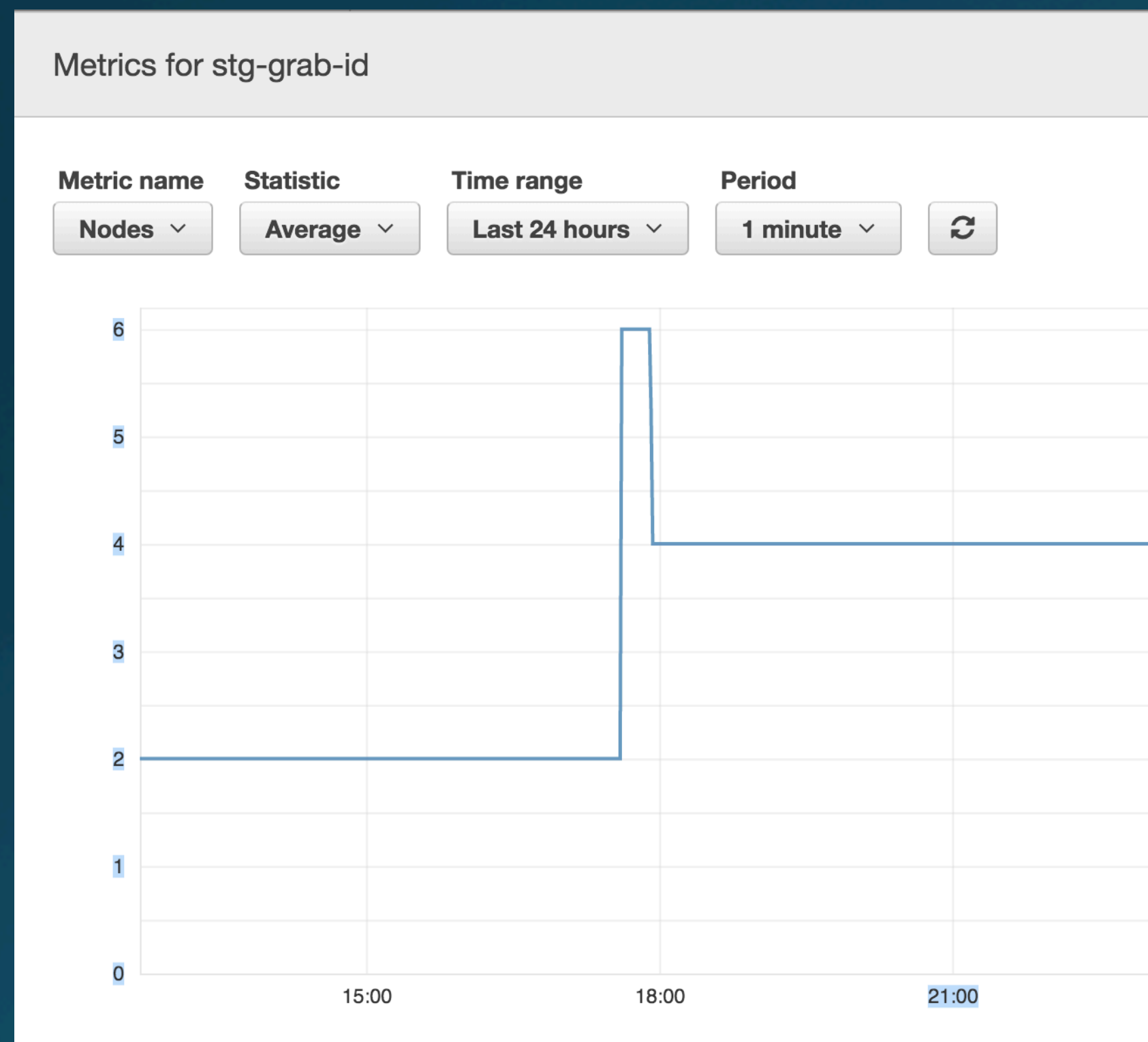
AWS ES Scalability Experience

Adding a new node

Migrate to new nodes instead
Example:
Add 2 nodes
2 nodes -> 6 nodes -> 4 nodes



What to trigger

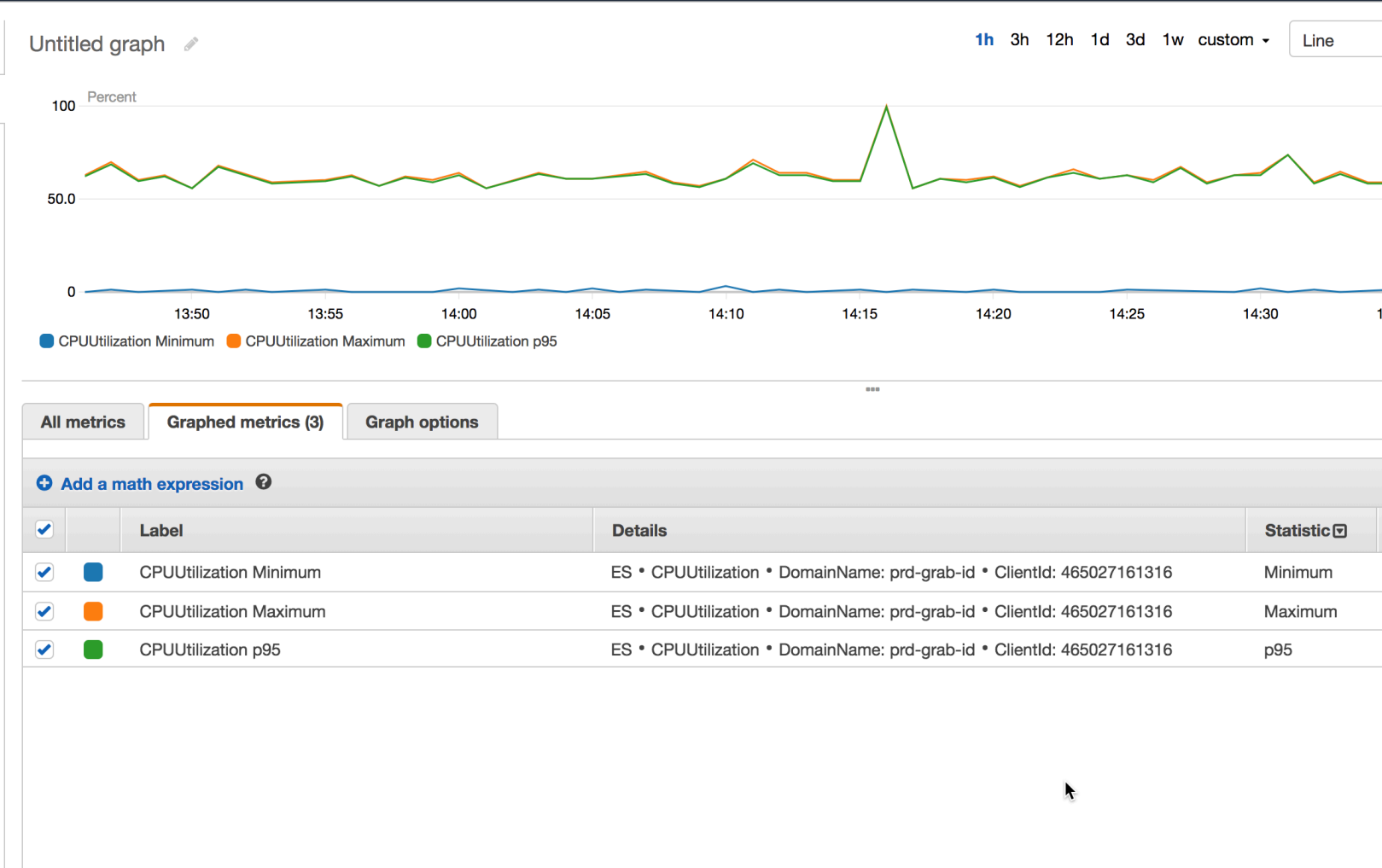


When upgrading instance type.

When adding new nodes.

When adding user access to the cluster.

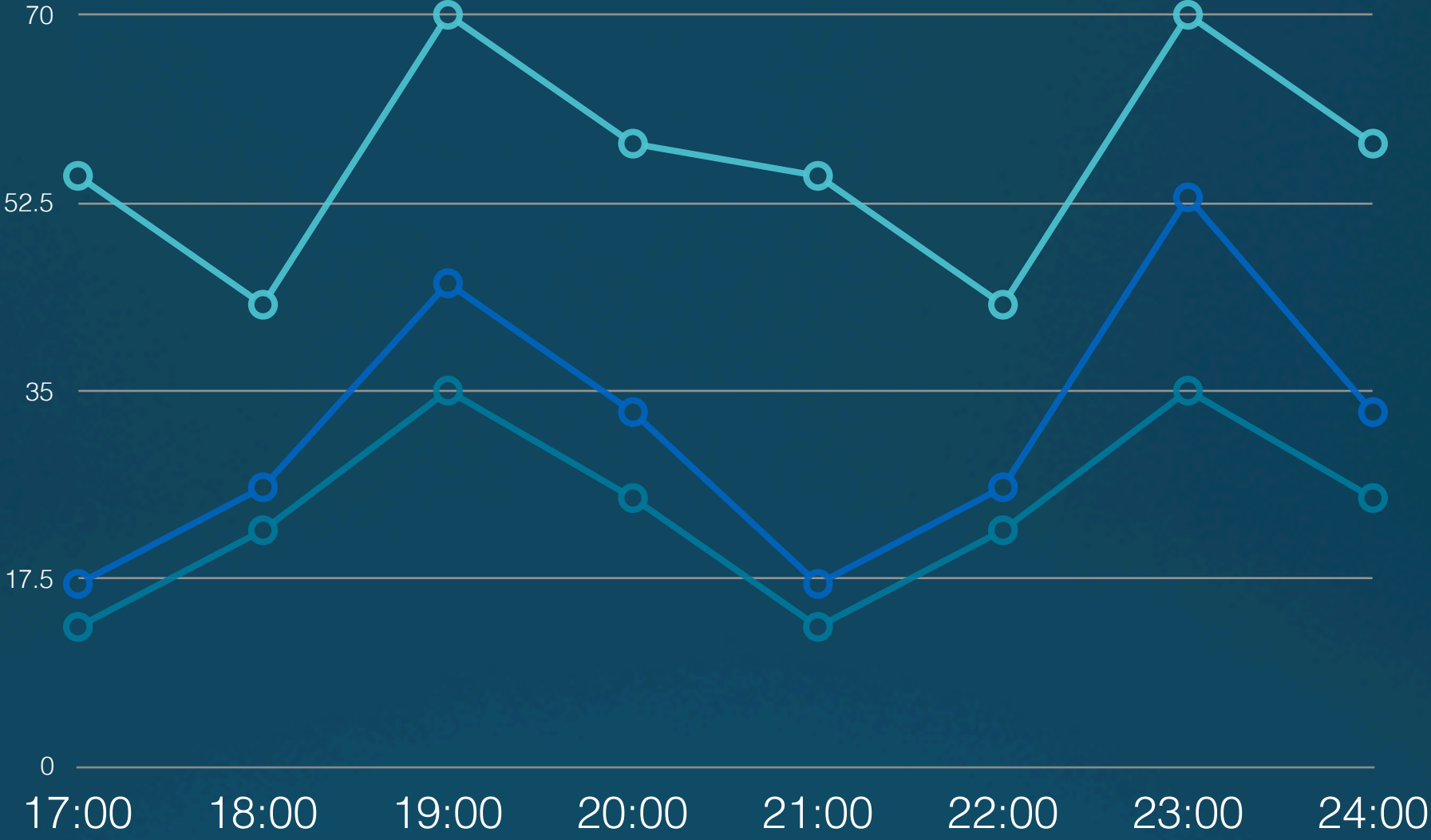
Why not CloudWatch



Streams for accesslog > Events for i-969d76bc

18 : 15 : 02 Local (GMT+05:30) →

Event Data	
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:01 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:01 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:01 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:01 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:02 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:02 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:02 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:02 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:03 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:03 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:03 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:03 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:03 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"
TC+5:30	▶ 122.166.211.122 - - [01/Aug/2014:12:45:03 +0000] "GET /cloudthat HTTP/1.1" 404 286 "-"



Max: max cpu of all the nodes

P99: %99 node lower than this CPU

P95: %95 node lower than this CPU

CloudWatch Logs

- No parser to parse the log.
- No aggregation on logs.
- No visualisation on logs.

We need more than scalability

- We need more powerful monitoring.
- Better version control.
- Security control / User control.
- Uptime become first priority

The Second Migration

From AWS ES to Elastic Cloud

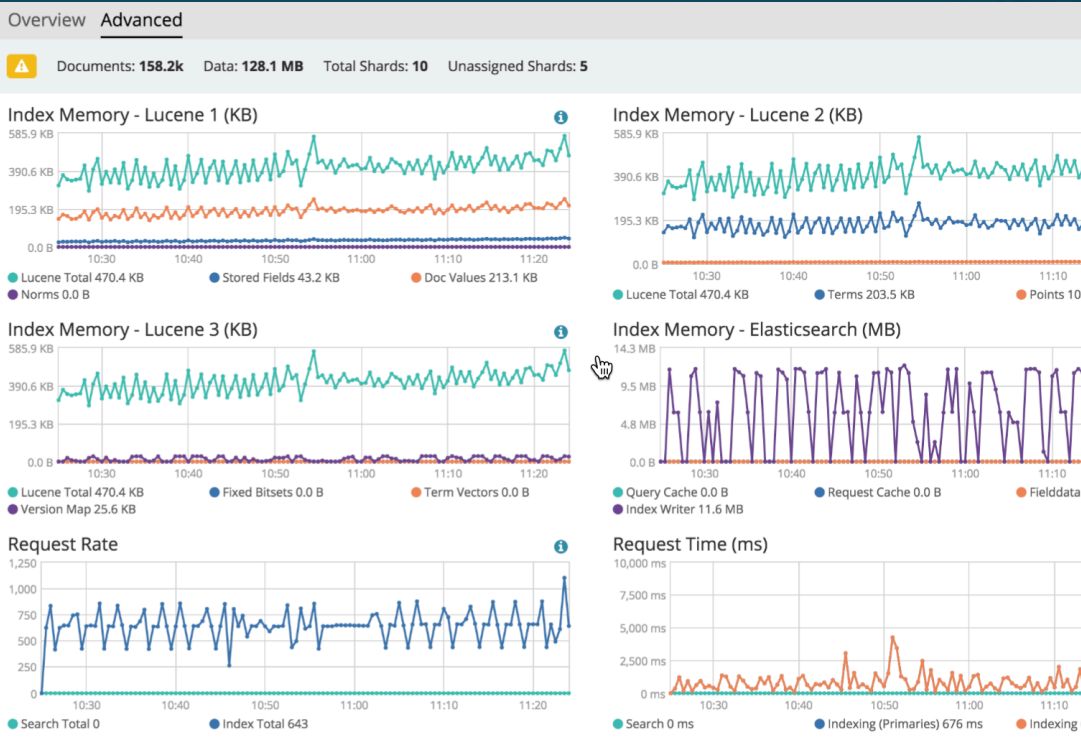
Stability is the first.

Elastic Cloud



AWS Marketplace

Billing is convenient.
Using AWS instance type



ES Monitoring

Index/Search rate, latency
Thread_pool/GC/Per node Resources
Index segment, memory

Cluster:

f4944a775dd36e9361c488291df5f285

Status:

Green

Nodes:

5

Indices:

10

Memory: 24GB / 124GB

Total Shards: 38

Unassigned Shards: 0

Documents: 1,291,936

Data: 2GB

Uptime: 11 days

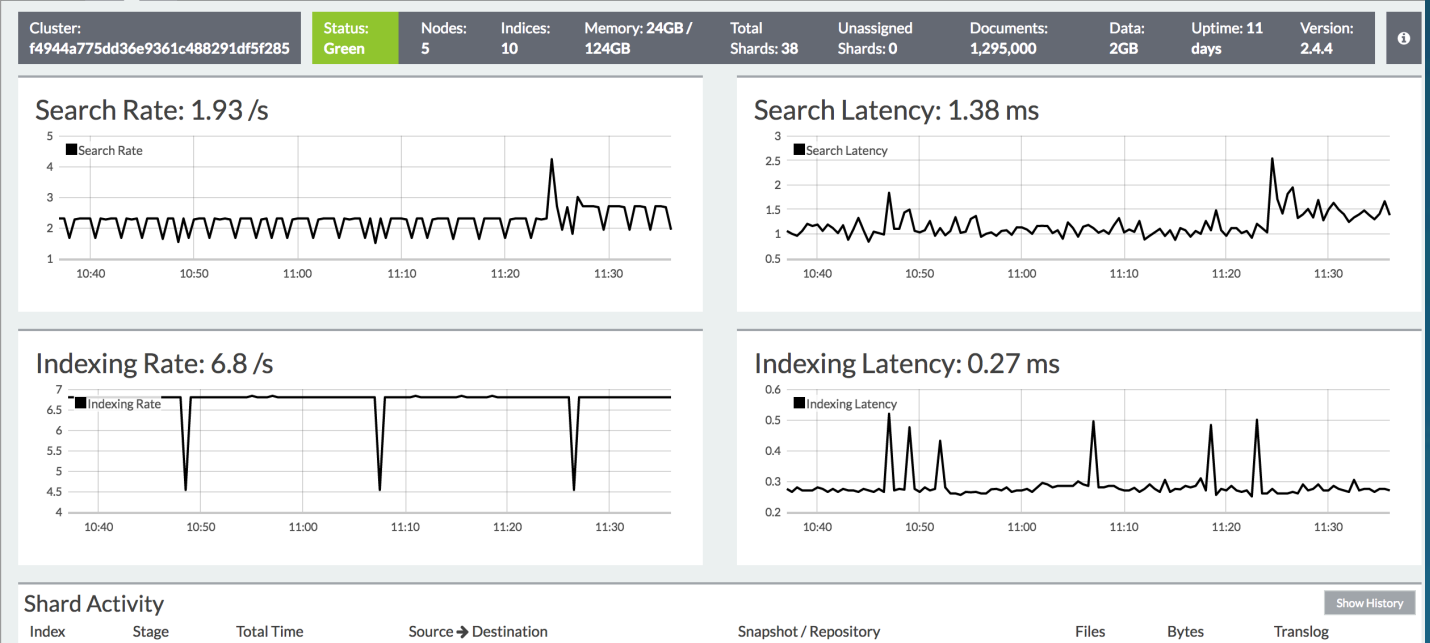
Version: 2.4.4

Nodes

Filter Nodes

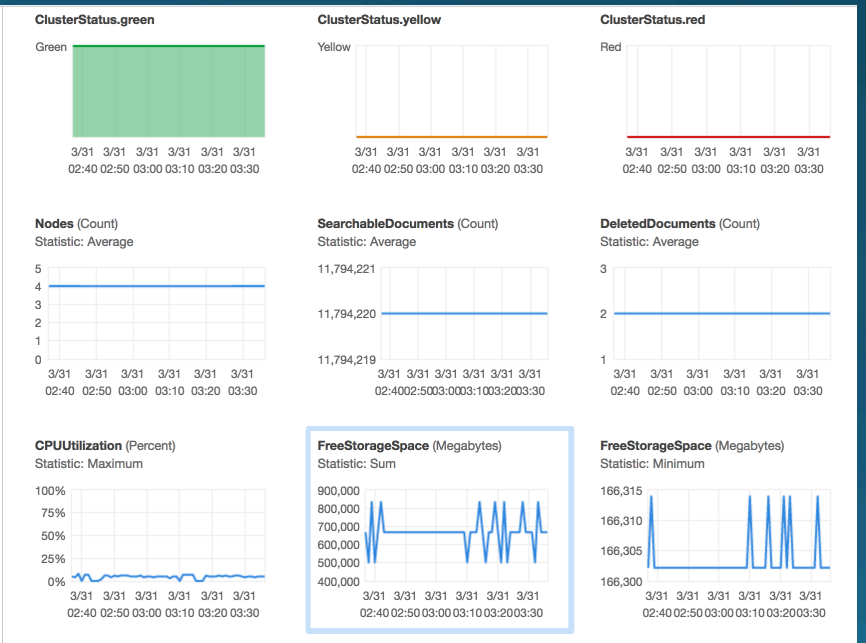
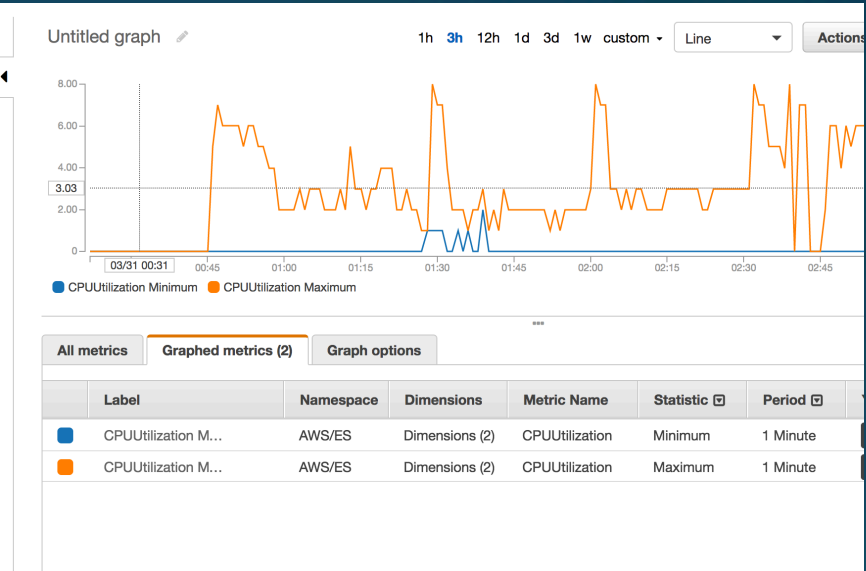
5 of 5

Name	CPU Usage	JVM Memory	Load Average	Disk Free Space	Shards	Status
<div><div><div>instance-0000000001</div><div>10.164.154.76:19268</div></div></div>	0.33%↓ <div><div>1.67% max</div><div>0.33% min</div></div>	19.67%↓ <div><div>21.5% max</div><div>19% min</div></div>	11.79↓ <div><div>11.91 max</div><div>5.03 min</div></div>	1.6TB↑ <div><div>1.6TB max</div><div>1.6TB min</div></div>	10	Online
<div><div><div>instance-0000000002</div><div>10.174.26.2:19116</div></div></div>	0%↓ <div><div>0.67% max</div><div>0.67% min</div></div>	7%↓ <div><div>9% max</div><div>6% min</div></div>	0↓ <div><div>0.22 max</div><div>0 min</div></div>	1.6TB↓ <div><div>1.6TB max</div><div>1.6TB min</div></div>	9	Online
<div><div><div>instance-0000000003</div><div>10.166.138.147:19997</div></div></div>	0%↓ <div><div>3% max</div><div>0.33% min</div></div>	42%↓ <div><div>43% max</div><div>41% min</div></div>	0.29↑ <div><div>0.95 max</div><div>0.05 min</div></div>	1.6TB↑ <div><div>1.6TB max</div><div>1.6TB min</div></div>	10	Online
<div><div><div>instance-0000000004</div><div>10.167.141.45:19350</div></div></div>	0%↓ <div><div>0% max</div><div>0% min</div></div>	5%↓ <div><div>6% max</div><div>3% min</div></div>	0.67↑ <div><div>1.26 max</div><div>0.26 min</div></div>	1.6TB↓ <div><div>1.6TB max</div><div>1.6TB min</div></div>	9	Online
<div><div><div>tiebreaker-0000000000</div><div>172.29.230.123:19112</div></div></div>	0%↓ <div><div>0% max</div><div>0% min</div></div>	13.33%↑ <div><div>25% max</div><div>3% min</div></div>	0.19↓ <div><div>0.43 max</div><div>0.04 min</div></div>	26.0GB↓ <div><div>26.0GB max</div><div>26.0GB min</div></div>	0	Online



AWS ES

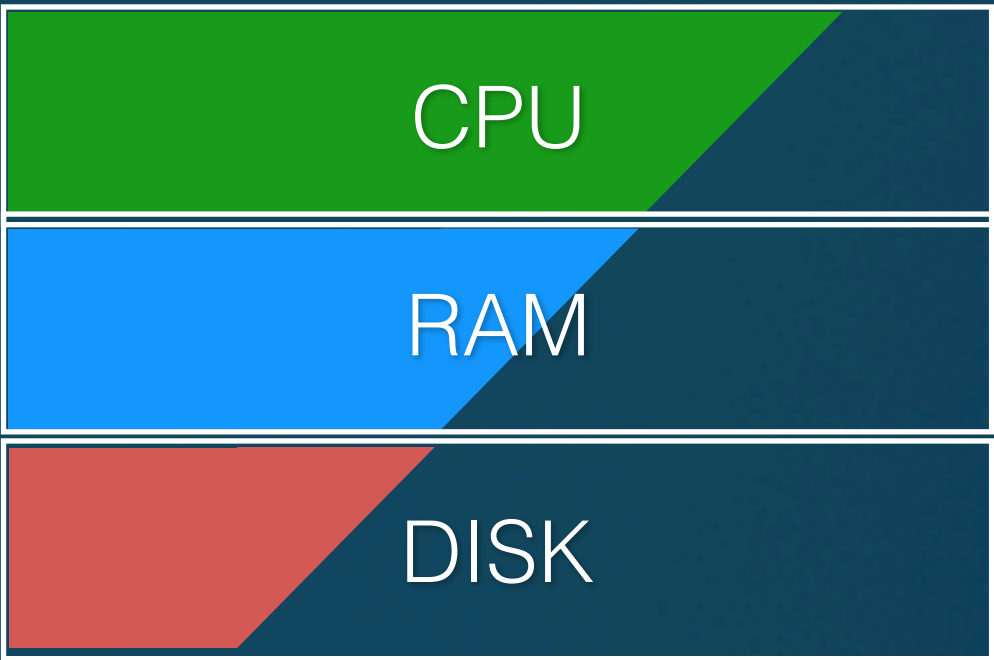
Elastic Cloud



i3 AWS instance 1:32

Point of Interest

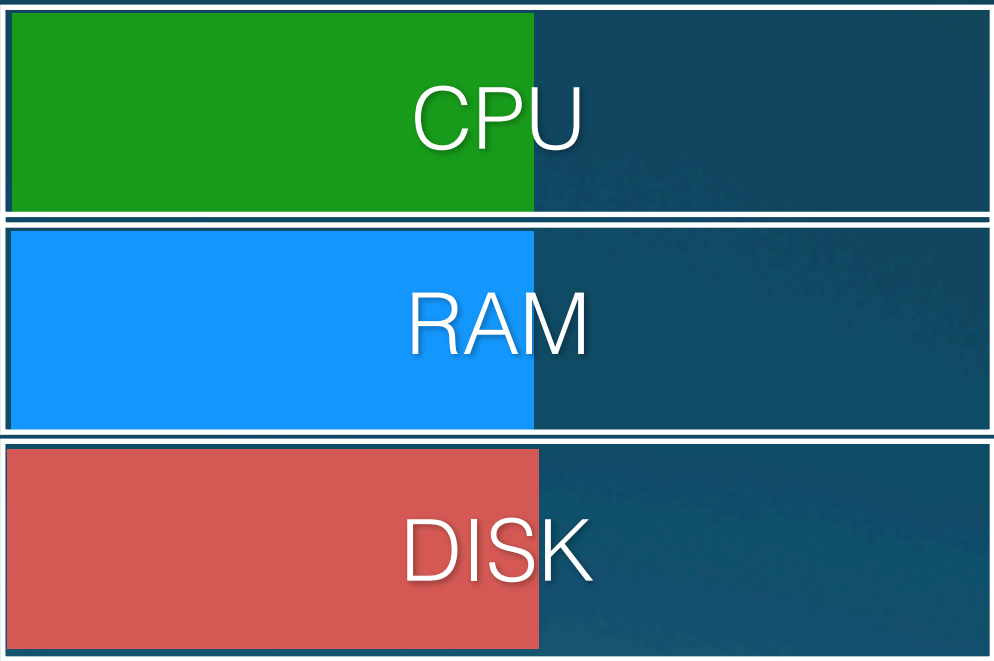
Searching POI with given lat/lon.
Multimatch/ boost / tie breaker
Sort by geo location



Waste Disk

Audit Log Service

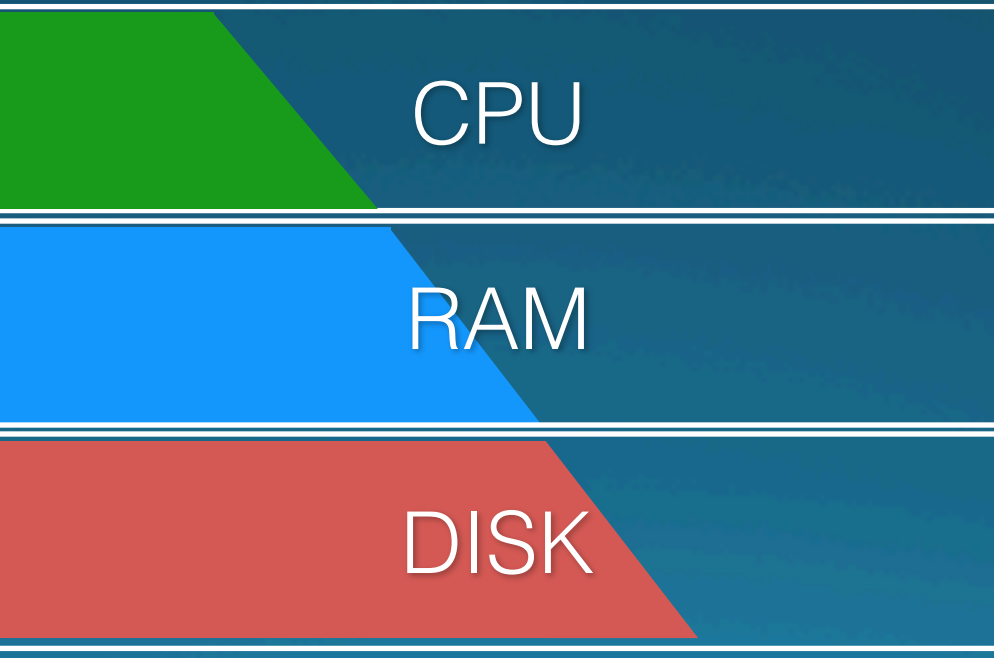
Large amount of daily data.
Log retention is short
Archive periodically.



Perfect

Action Trace log

Medium size of daily logs.
Require years of data.



Waste CPU

The Third Migration

To Elastic Cloud Enterprise

Looking for manageable and flexible

What we expected

Customized Resources

Fully use our CPU Mem Disk, minimize the cost
Using AWS instance type
Scale in/out resource on demand.

High Availability

Auto recover nodes when it dies.
High availability topology.
Scaling out/in, up/down in a proper way.

Self Managed Cluster

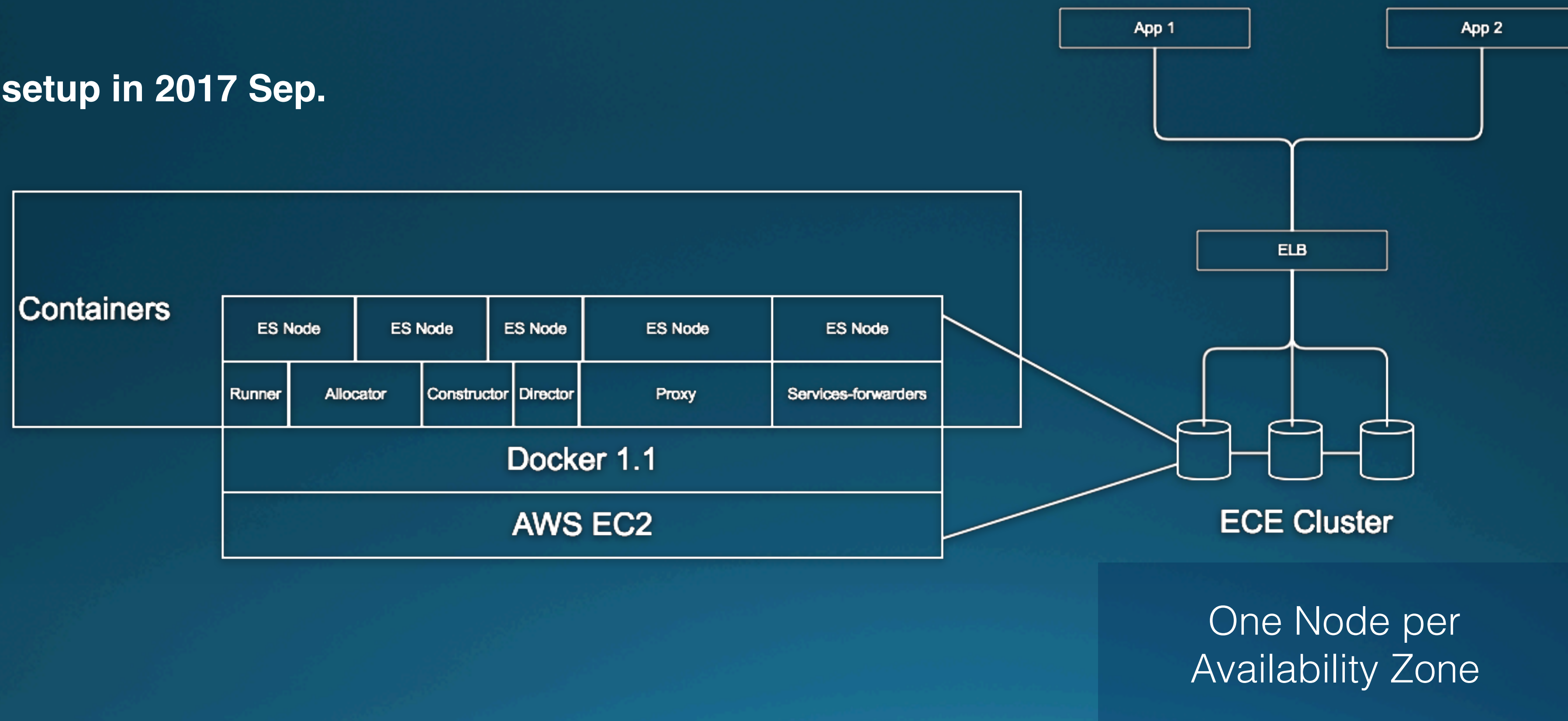
Remove sysops from the deploy workflow.
Review resource usage by Dbops.
Full Elasticsearch feature

Easy Integration

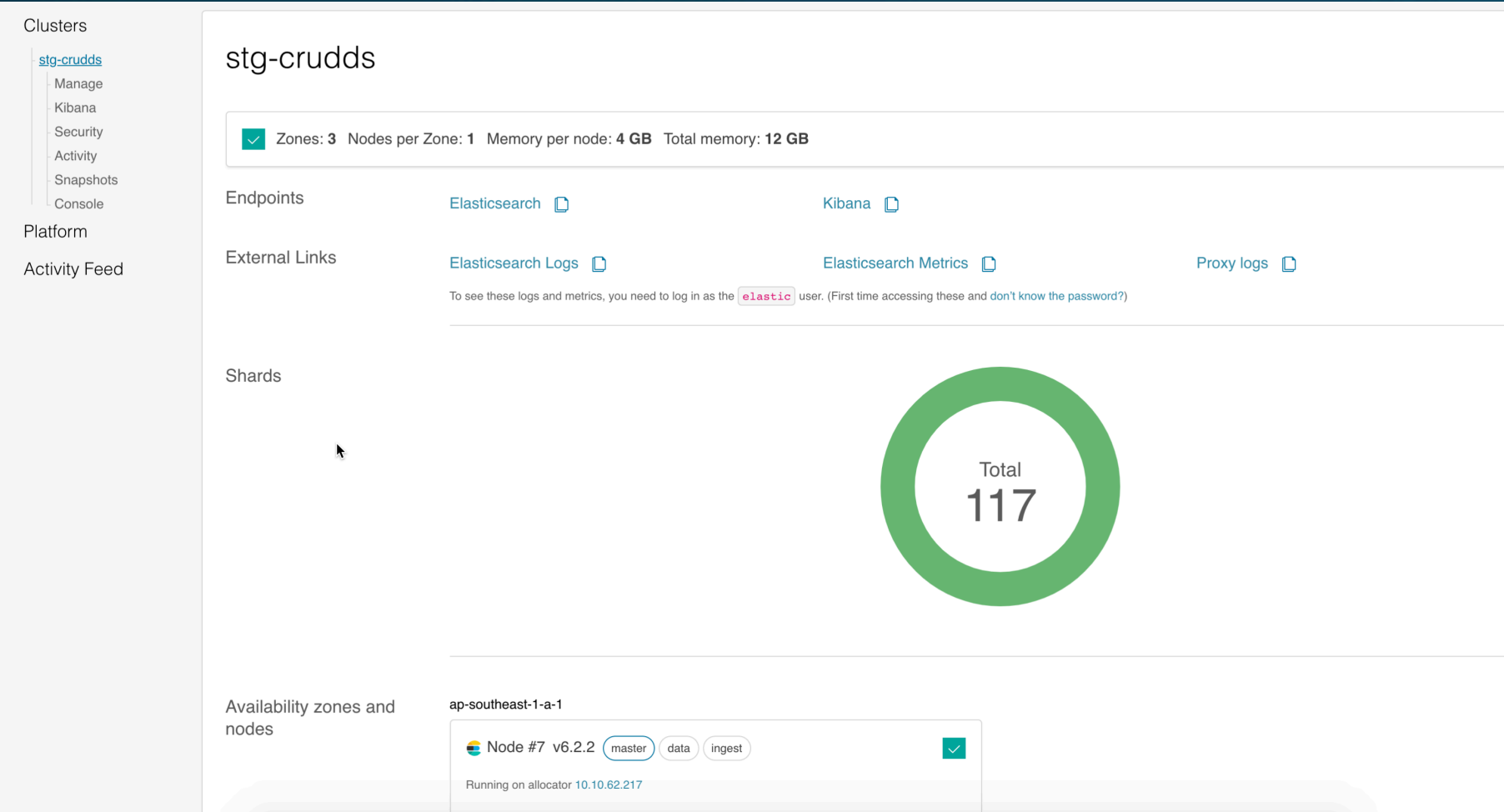
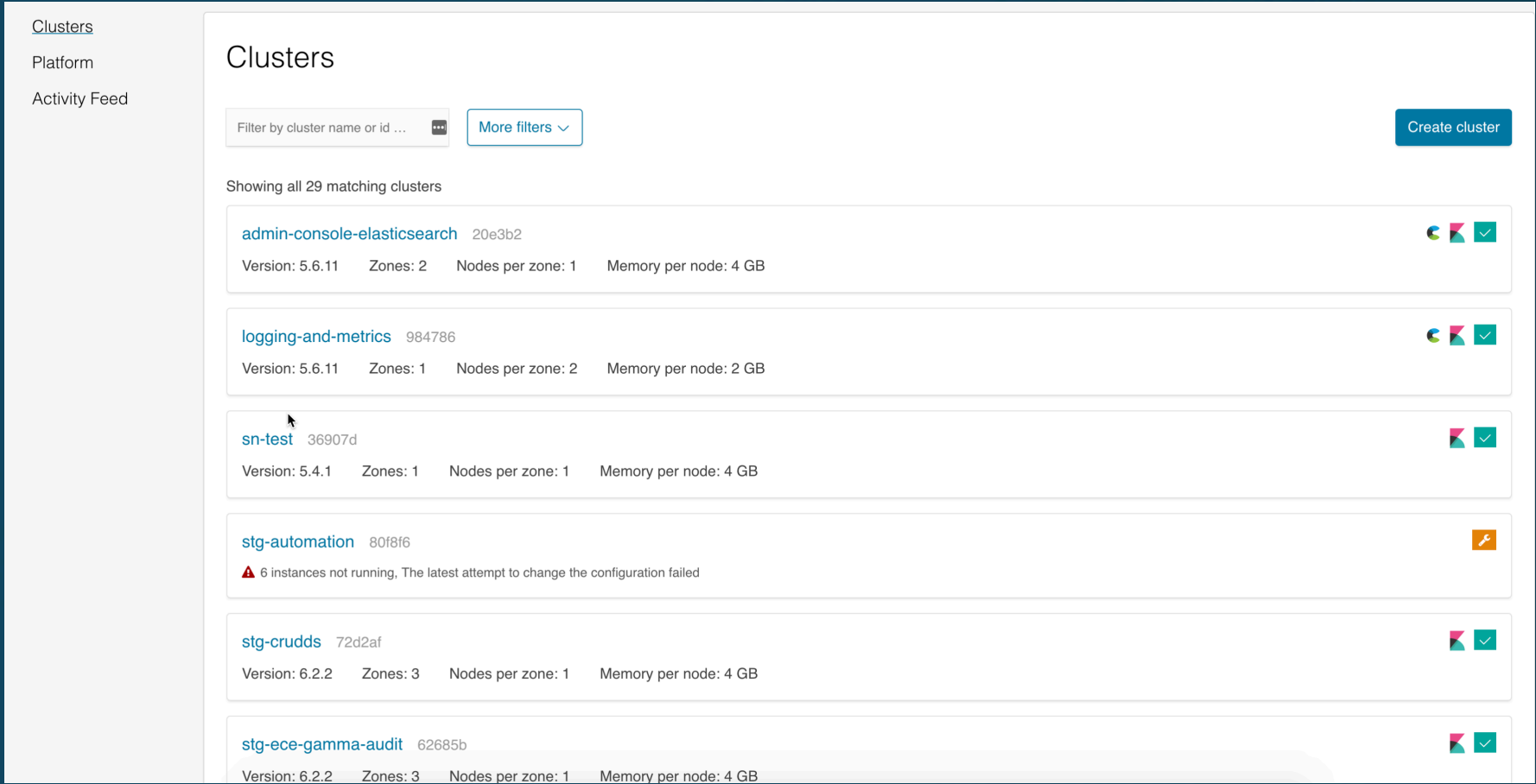
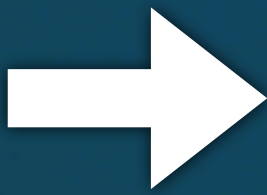
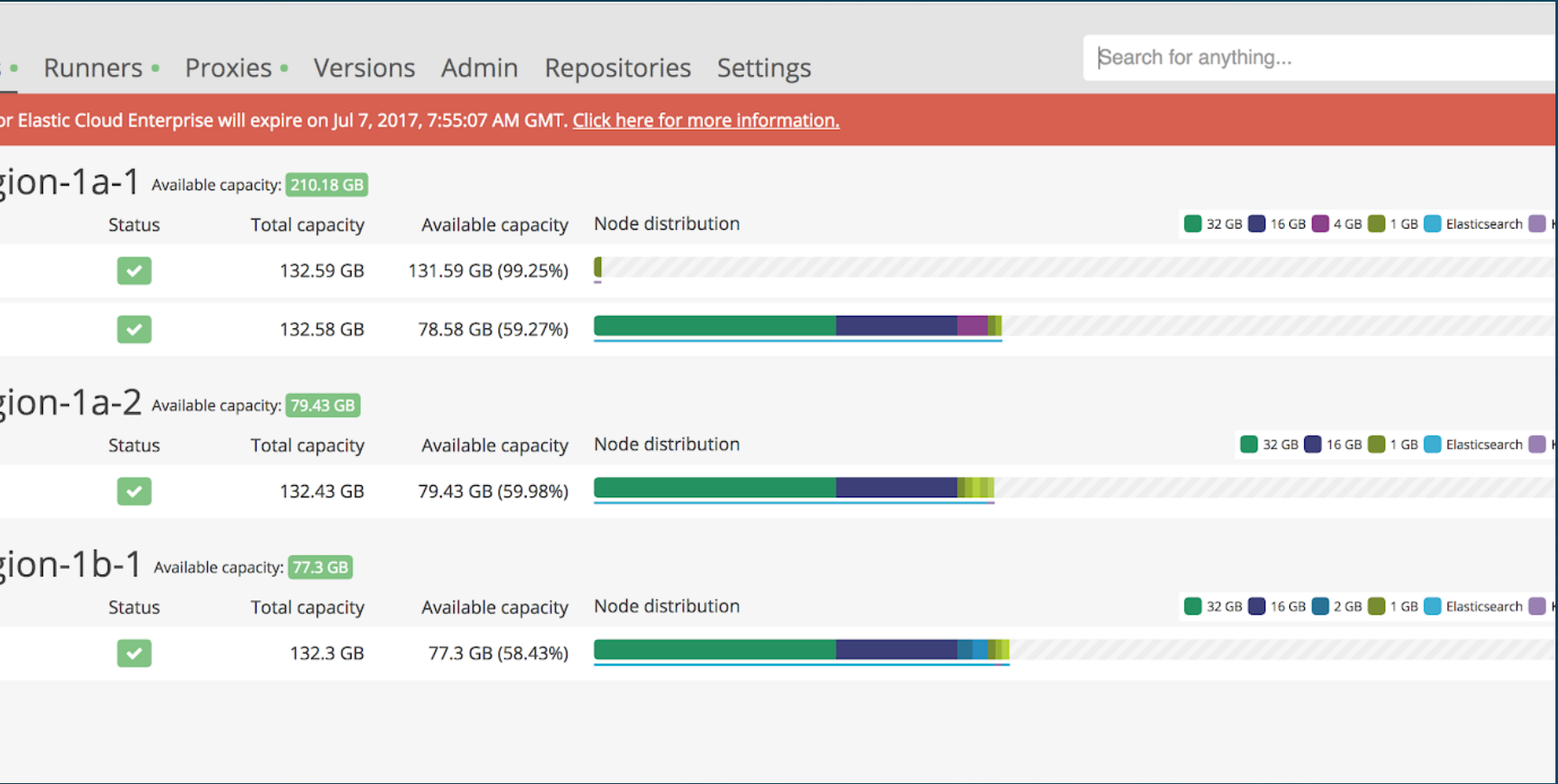
Provide easy integration with other Grab service.
Good enough API call for automation.
User friendly interface.

Easy Integration

The initial setup in 2017 Sep.

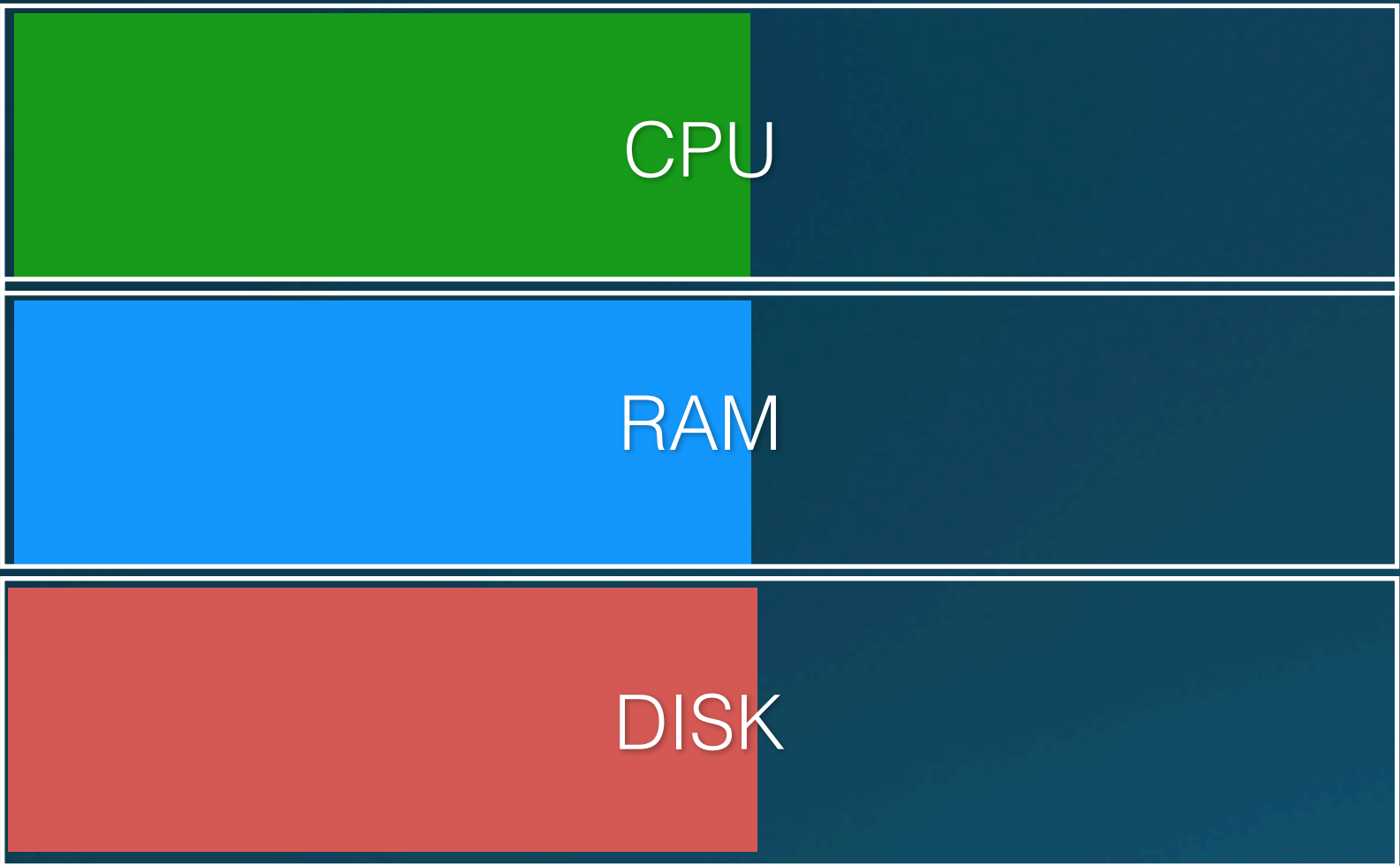


Self Managed Cluster



Status	Step Name	Started	Completed	Time to Complete
> ✓	Plan successful	Nov 2, 2018, 3:03:20 AM	Nov 2, 2018, 3:03:20 AM	0 ms
> ✓	Performing clean-up	Nov 2, 2018, 3:03:17 AM	Nov 2, 2018, 3:03:20 AM	3 seconds
> ✓	Applying X-Pack monitoring configuration	Nov 2, 2018, 3:03:16 AM	Nov 2, 2018, 3:03:17 AM	291 ms
> ✓	Applying X-Pack license to cluster	Nov 2, 2018, 3:03:16 AM	Nov 2, 2018, 3:03:16 AM	313 ms
> ✓	Ensuring snapshot repository exists	Nov 2, 2018, 3:03:15 AM	Nov 2, 2018, 3:03:16 AM	1 seconds
> ✓	Fetching information about current state of cluster nodes	Nov 2, 2018, 3:03:15 AM	Nov 2, 2018, 3:03:15 AM	12 ms
> ✓	Resizing nodes	Nov 2, 2018, 3:03:15 AM	Nov 2, 2018, 3:03:15 AM	9 ms
> ✓	Deleting cluster nodes	Nov 2, 2018, 3:03:15 AM	Nov 2, 2018, 3:03:15 AM	15 ms
> ✓	Waiting until nodes are deallocated	Nov 2, 2018, 3:03:10 AM	Nov 2, 2018, 3:03:15 AM	5 seconds
> ✓	Deallocating cluster nodes	Nov 2, 2018, 3:03:10 AM	Nov 2, 2018, 3:03:10 AM	15 ms
> ✓	Verifying cluster is not in split-brain state	Nov 2, 2018, 3:03:10 AM	Nov 2, 2018, 3:03:10 AM	9 ms
> ✓	Waiting until master nodes are elected	Nov 2, 2018, 3:03:10 AM	Nov 2, 2018, 3:03:10 AM	12 ms
> ✓	Waiting until nodes are stopped	Nov 2, 2018, 3:02:29 AM	Nov 2, 2018, 3:03:10 AM	41 seconds
> ✓	Stopping nodes	Nov 2, 2018, 3:02:29 AM	Nov 2, 2018, 3:02:29 AM	13 ms
> ✓	Setting maintenance mode	Nov 2, 2018, 3:02:24 AM	Nov 2, 2018, 3:02:24 AM	13 ms
> ✓	Setting Elasticsearch cluster quorum size	Nov 2, 2018, 3:02:24 AM	Nov 2, 2018, 3:02:24 AM	485 ms
> ✓	Migrating shard data	Nov 2, 2018, 3:02:23 AM	Nov 2, 2018, 3:02:24 AM	552 ms
> ✓	apply-transient-cluster-settings	Nov 2, 2018, 3:02:23 AM	Nov 2, 2018, 3:02:23 AM	2 ms

Customized Resource



A node

8vCPU

32GB (32GB JVM)

1 TB data

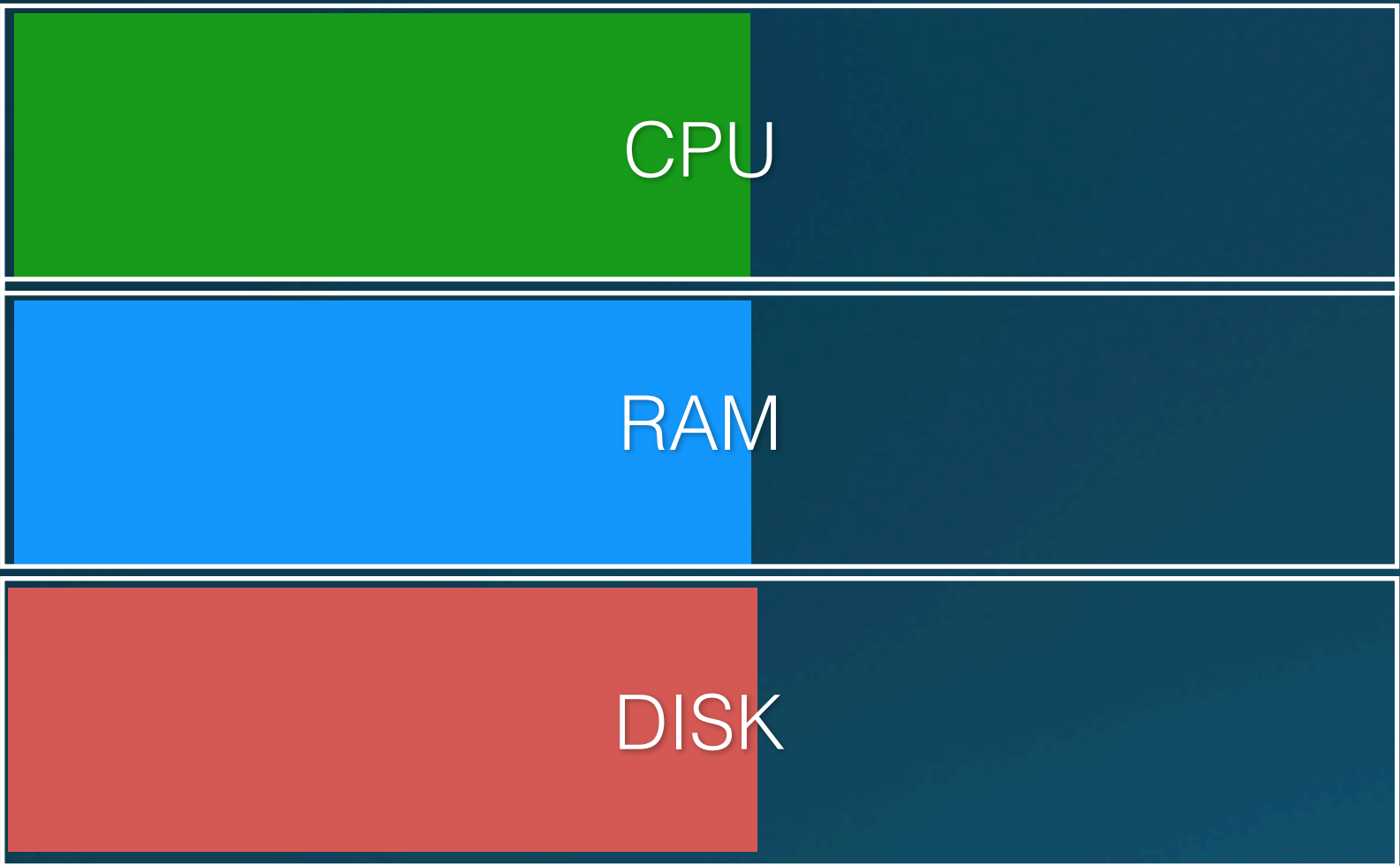
M4.16xlarge

64vCPU

256GB

8 TB disk

Customized Resource



A node

16 vCPU

64GB(32GB JVM)

4TB

c5.18xlarge

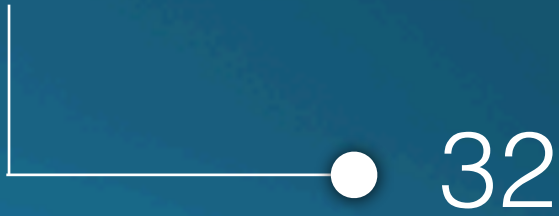
72vCPU

144GB

8 TB disk

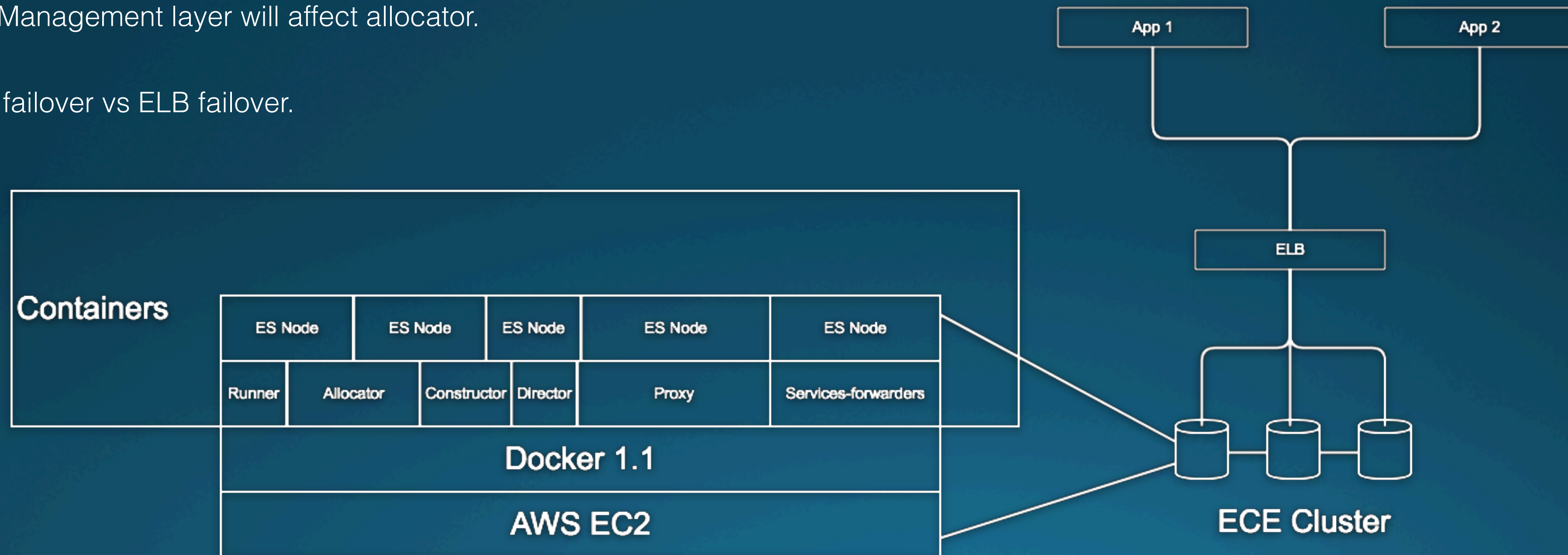
```
Math.min(16,Math.max(2,(16*instanceCapacity*1.0/1024/64).toInt))
```

Max 16vCpu per 64GB RAM

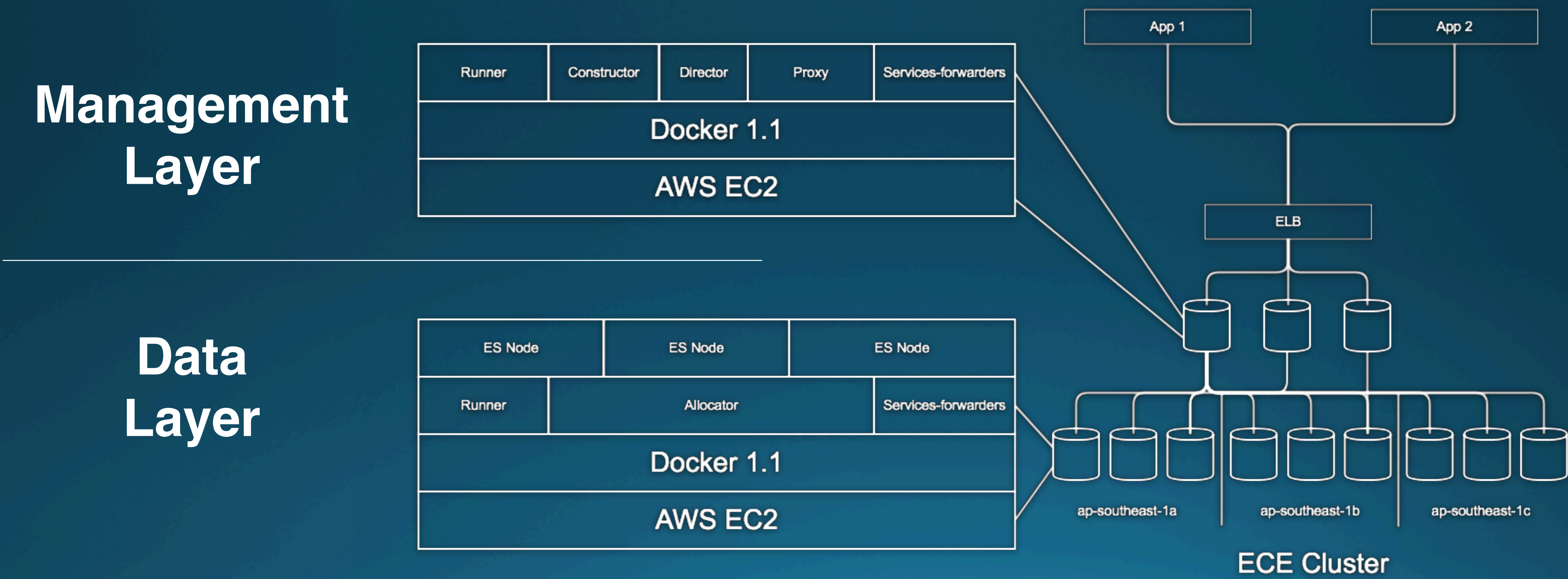


High Availability ?

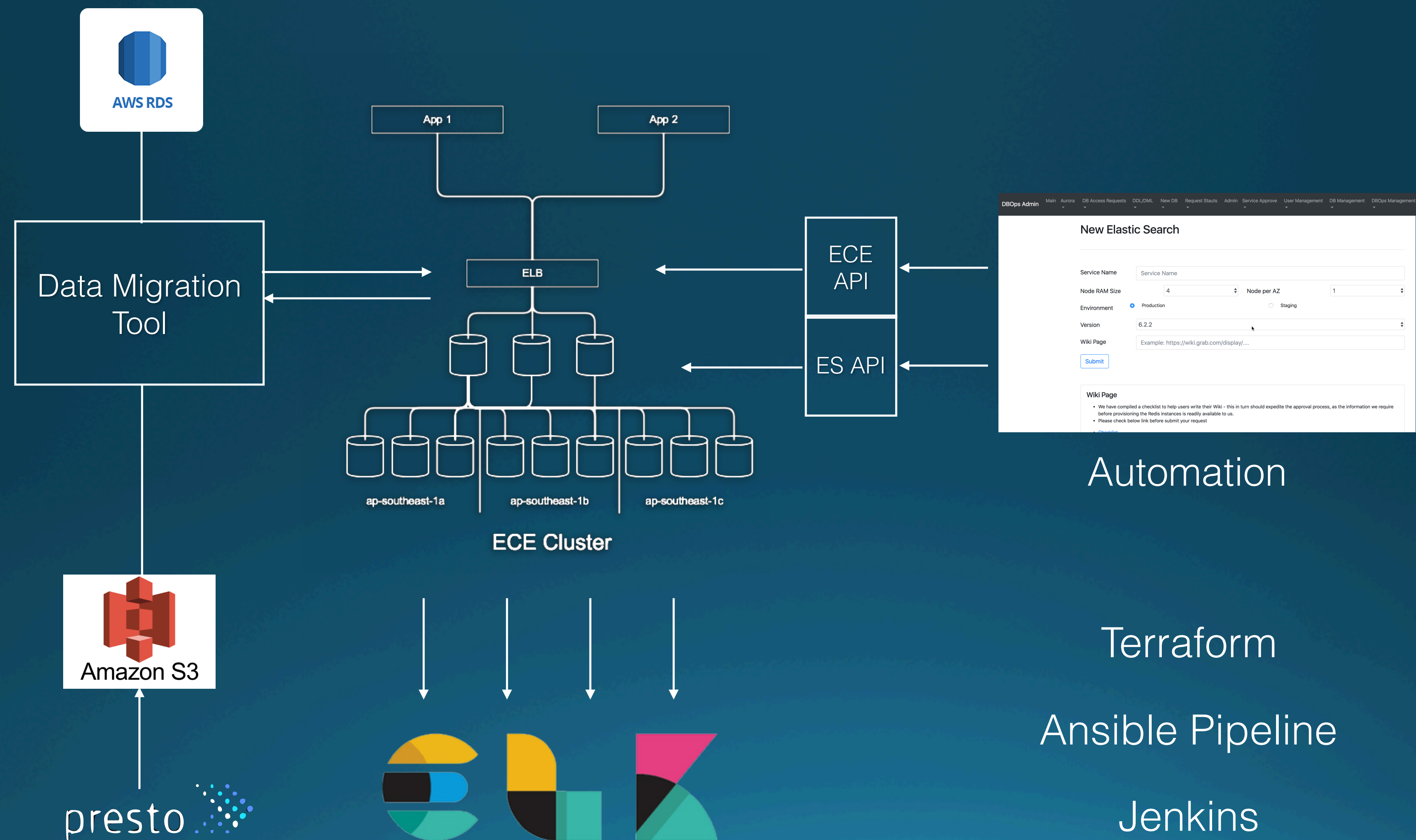
- Proxy and ES node container share the same Node. Ref <https://www.elastic.co/guide/en/cloud-enterprise/current/ece-monitoring-ece-proxies.html>
- Change in Management layer will affect allocator.
- Zookeeper failover vs ELB failover.



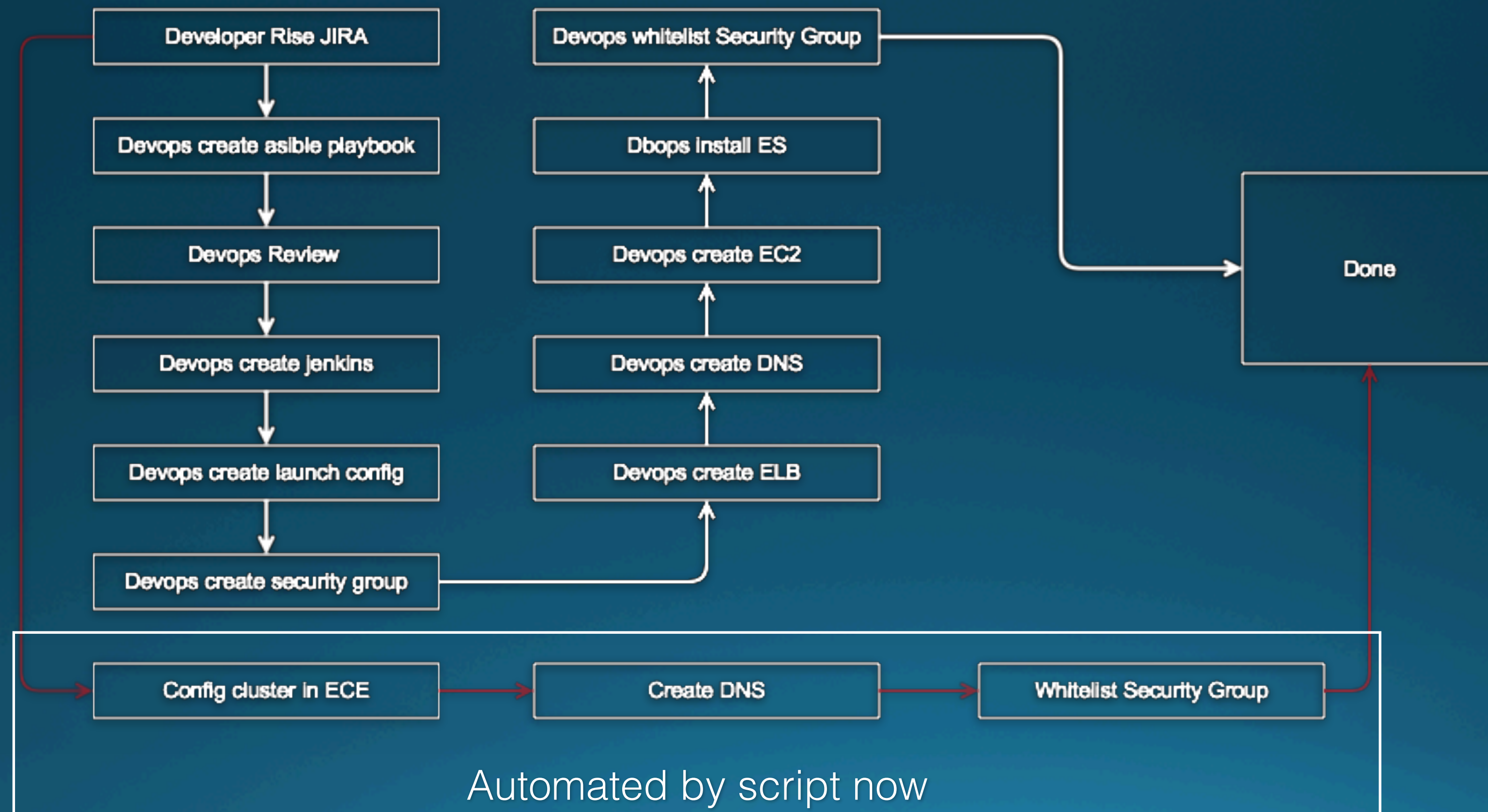
Scale to Medium



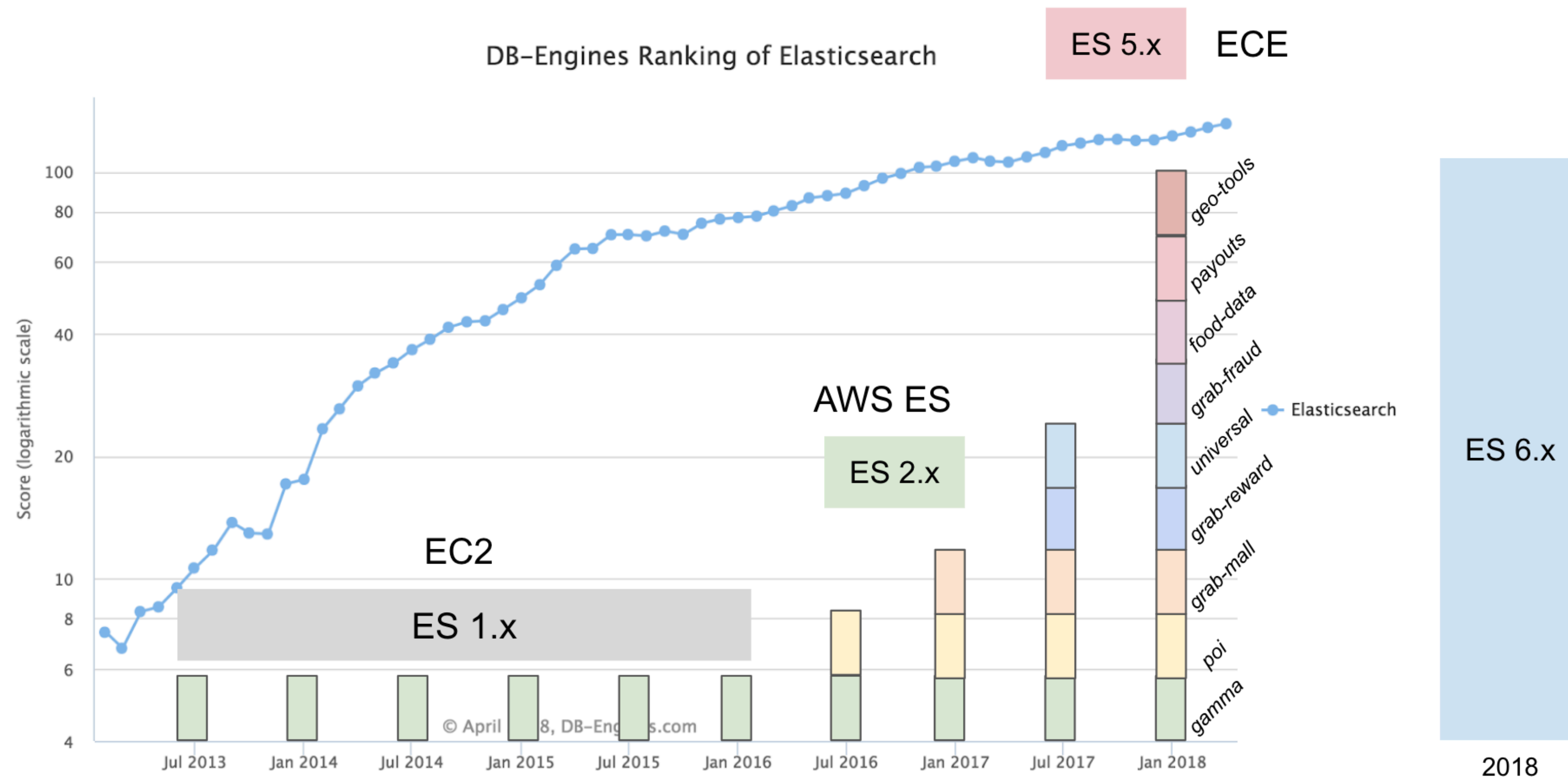
Integrate with Grab



What are all these for ?



Goal: Help Our Developers



Thank You



elastic 中文社区

专业、垂直、纯粹的 Elastic 开源技术交流社区

<https://elasticsearch.cn/>