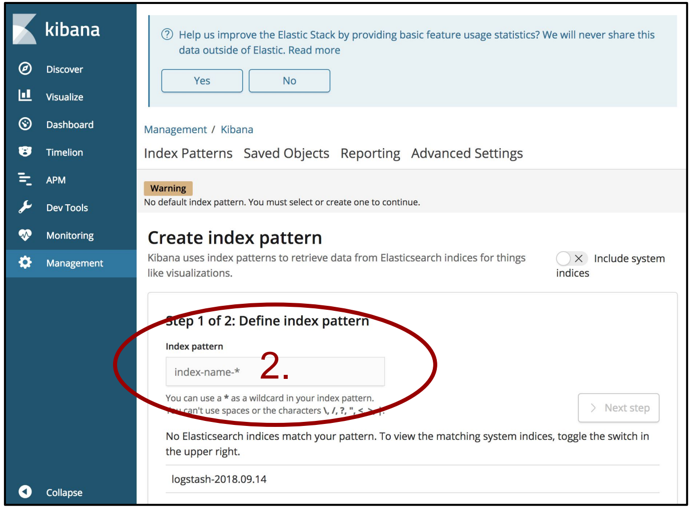
start analyse data with kibana

Peerapong Thongpubet

Step.1 Create Index Pattern

1. Open “Management” Menu
2. Input index pattern: logstash*
3. Click “next step”
4. Select time fileter name “@timestamp”
5. Click create pattern



Help us improve the Elastic Stack by providing basic feature usage statistics? We will never share this data outside of Elastic. Read more

No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

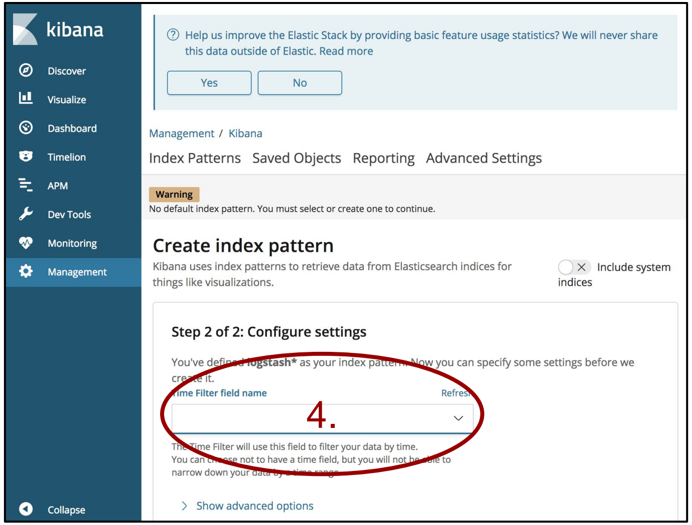
Step 1 of 2: Define index pattern

Index pattern: index-name-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

No Elasticsearch indices match your pattern. To view the matching system indices, toggle the switch in the upper right.

logstash-2018.09.14



Help us improve the Elastic Stack by providing basic feature usage statistics? We will never share this data outside of Elastic. Read more

No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Step 2 of 2: Configure settings

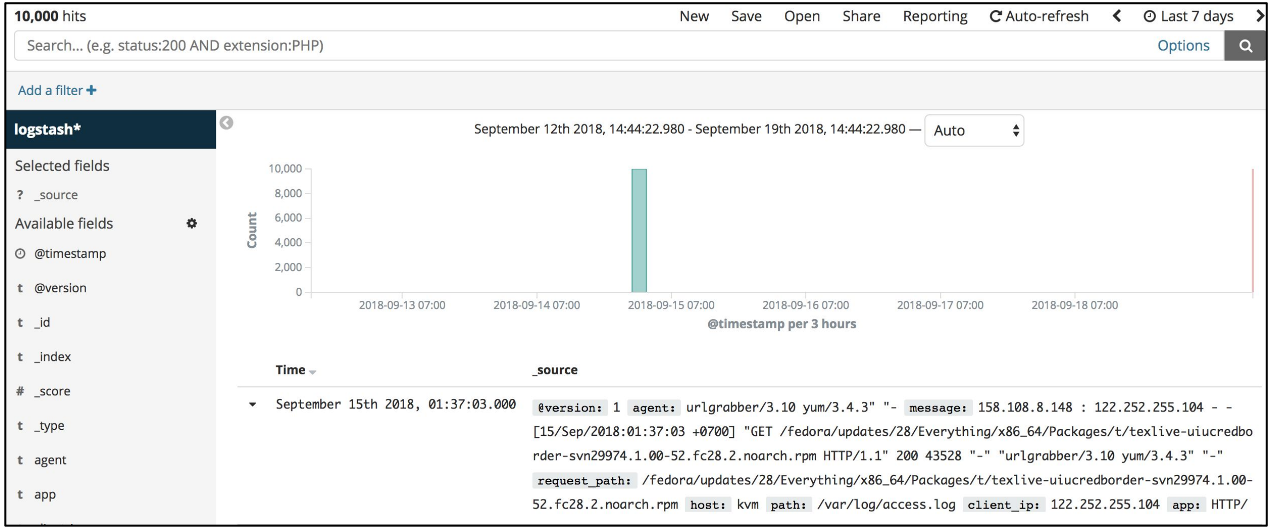
You've defined **logstash*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

Step.2 Display Document with “Discovery Menu”

1. Open “Discovery” Menu
2. Select time (Top right) : Last 7 day



10,000 hits

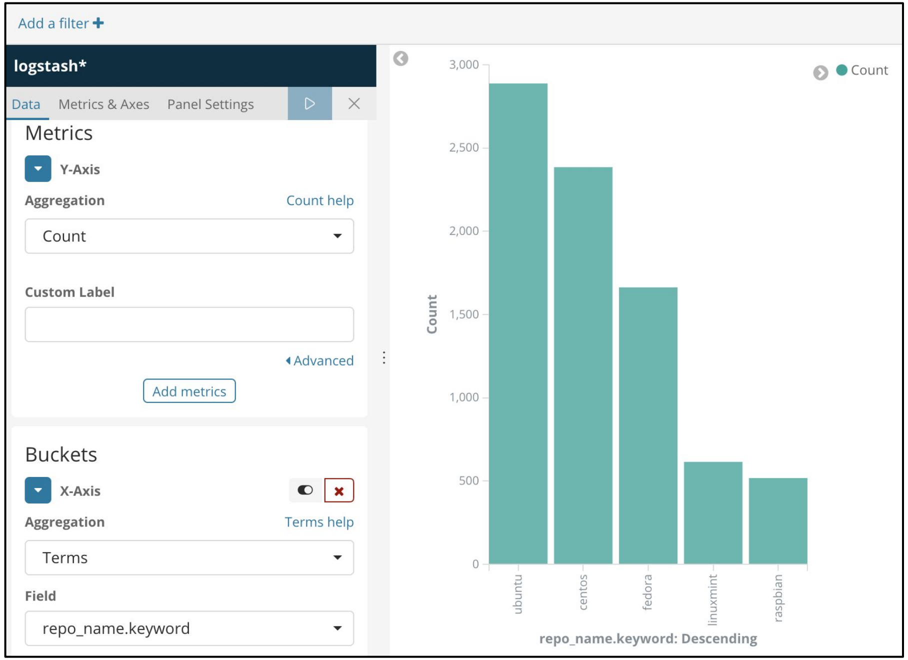
Last 7 days

September 12th 2018, 14:44:22.980 - September 19th 2018, 14:44:22.980 —

```
September 15th 2018, 01:37:03.000   @version: 1 agent: urlgrabber/3.10 yum/3.4.3" "- message: 158.108.8.148 : 122.252.255.104 - -
                                    [15/Sep/2018:01:37:03 +0700] "GET /fedora/updates/28/Everything/x86_64/Packages/t/texlive-uiucredbo
                                    rder-svn29974.1.00-52.fc28.2.noarch.rpm HTTP/1.1" 200 43528 "-" "urlgrabber/3.10 yum/3.4.3" "-"
                                    request_path: /fedora/updates/28/Everything/x86_64/Packages/t/texlive-uiucredborder-svn29974.1.00-
                                    52.fc28.2.noarch.rpm host: kvm path: /var/log/access.log client_ip: 122.252.255.104 app: HTTP/
```

Step.3 Create Vitualize (Virtical Bar)

1. Click “**Vitualize**” Menu
2. Create a vitualization (**Click button +**)
3. Select visualization type “**Virtical Bar**”
4. Select index “**Logstash***”
5. Select “Y-Axis” Value “**count**”
6. Select buckets type “X-Axis”
7. Select aggregation “**Term**”
8. Select feild “**repo_name.keyword**”
9. Save vitualization



logstash*

Metrics

Y-Axis

Aggregation: Count

Custom Label:

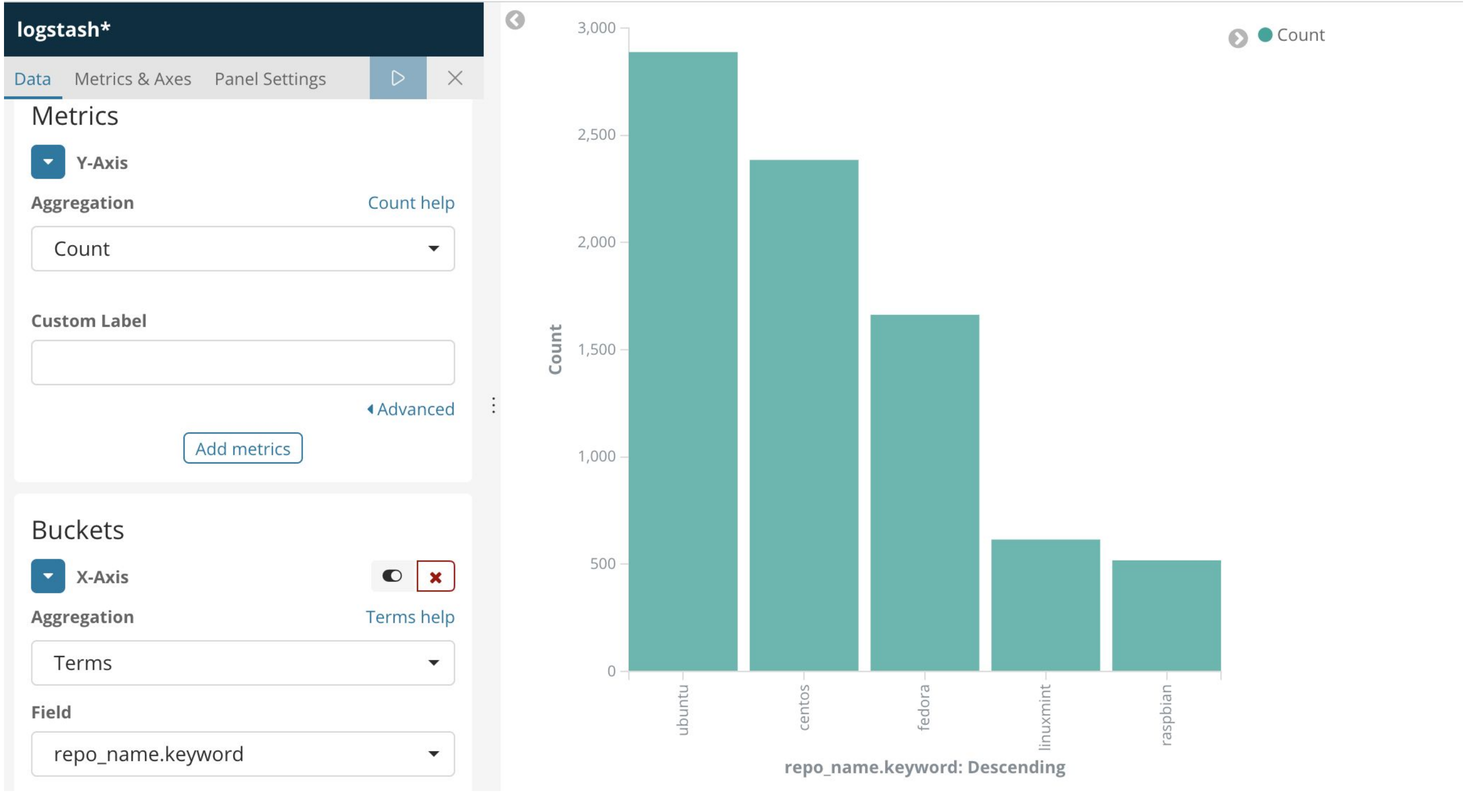
Buckets

X-Axis

Aggregation: Terms

Field: repo_name.keyword

Example “Virtical Bar”



logstash*

Metrics

Y-Axis

Aggregation: Count

Custom Label:

Buckets

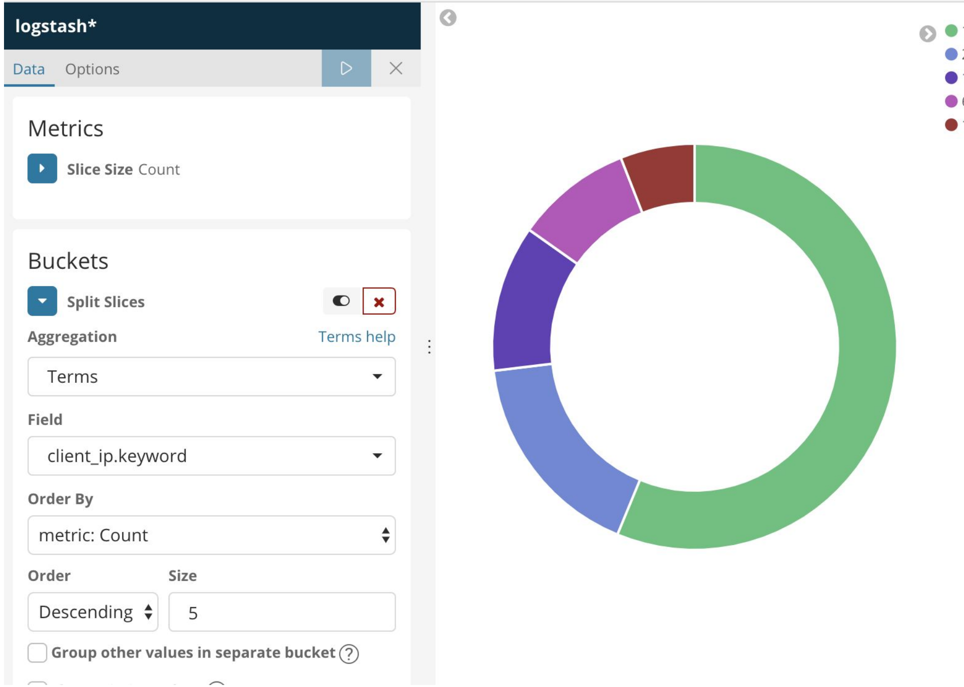
X-Axis

Aggregation: Terms

Field: repo_name.keyword

Step.3 Create Vitualize (Pie)

1. Click “Vitualize” Menu
2. Create a vitualization (Click button +)
3. Select visualization type “**Pie**”
4. Select index “Logstash*”
5. Select “Slice Size” Value “**count**”
6. Select buckets type “Split Slices”
7. Select aggregation “**Term**”
8. Select feild “repo_name.keyword”
9. Save vitualization



logstash*

Metrics

Slice Size Count

Buckets

Split Slices

Aggregation: Terms

Field: client_ip.keyword

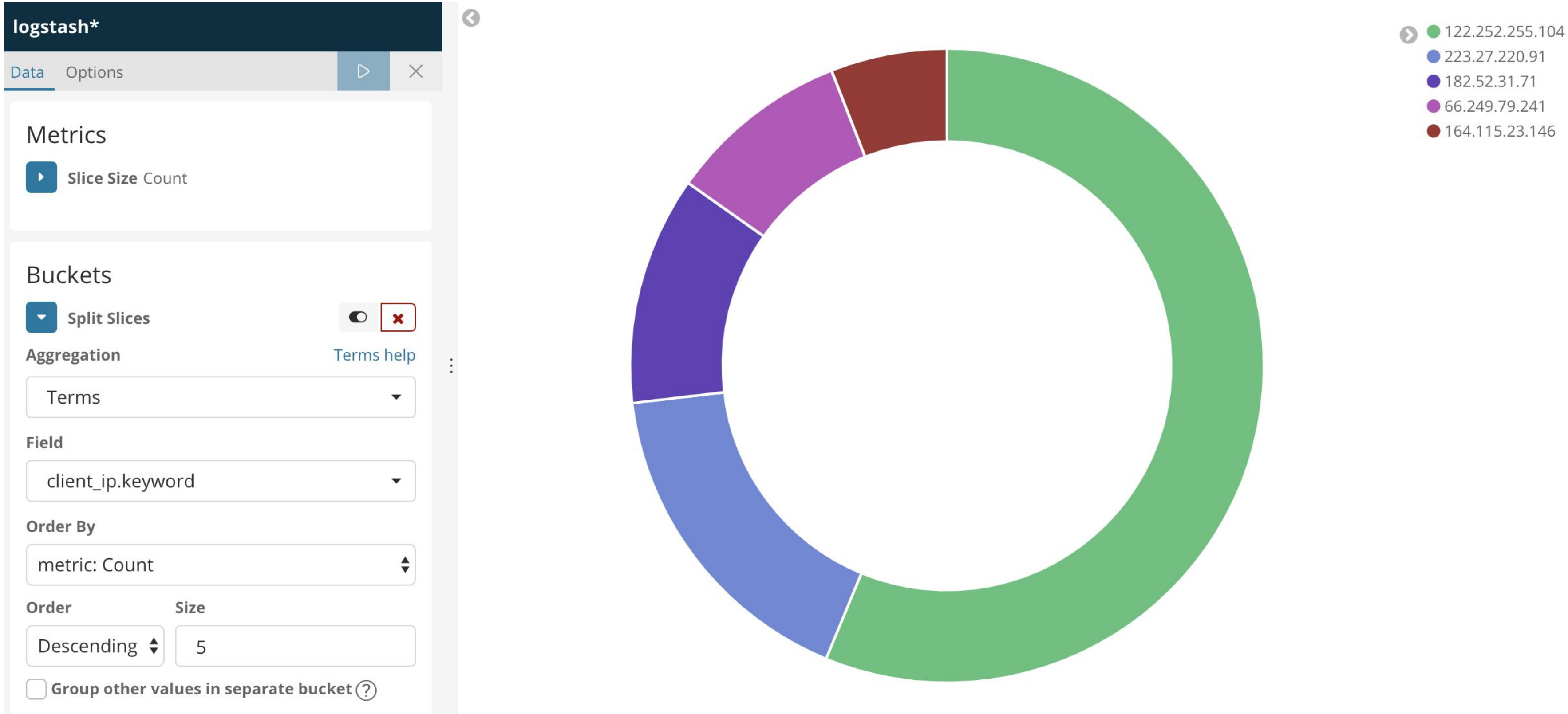
Order By: metric: Count

Order: Descending

Size: 5

□ Group other values in separate bucket

Eample “Pie”



Metrics

Slice Size Count

Buckets

Split Slices

Aggregation: Terms

Field: client_ip.keyword

Order By: metric: Count

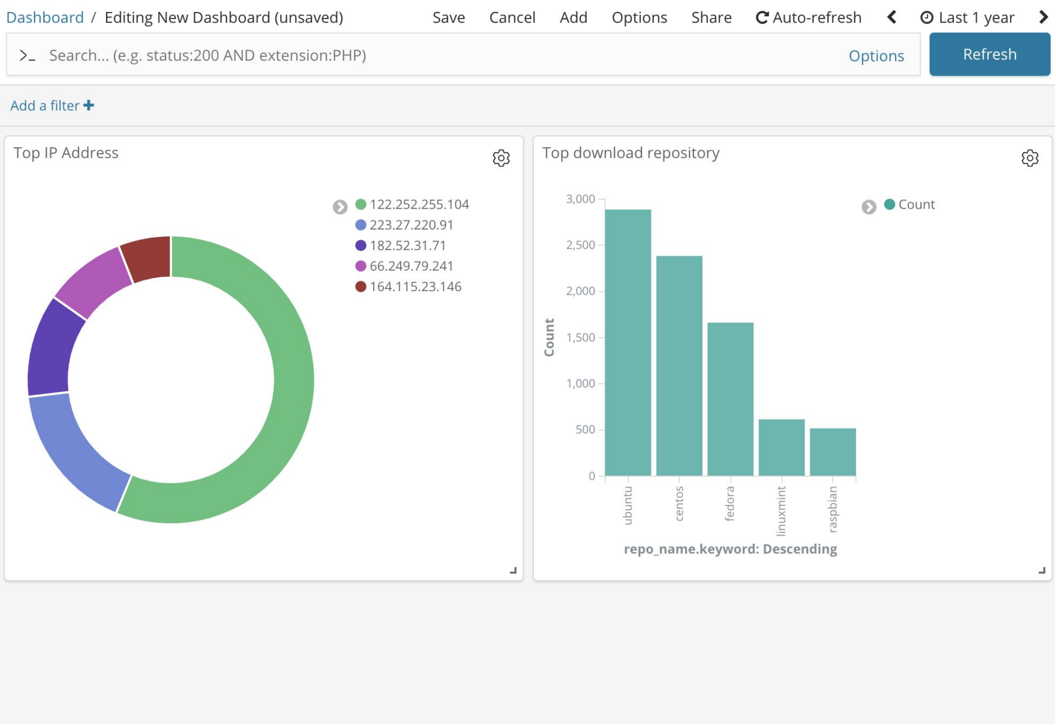
Order: Descending

Size: 5

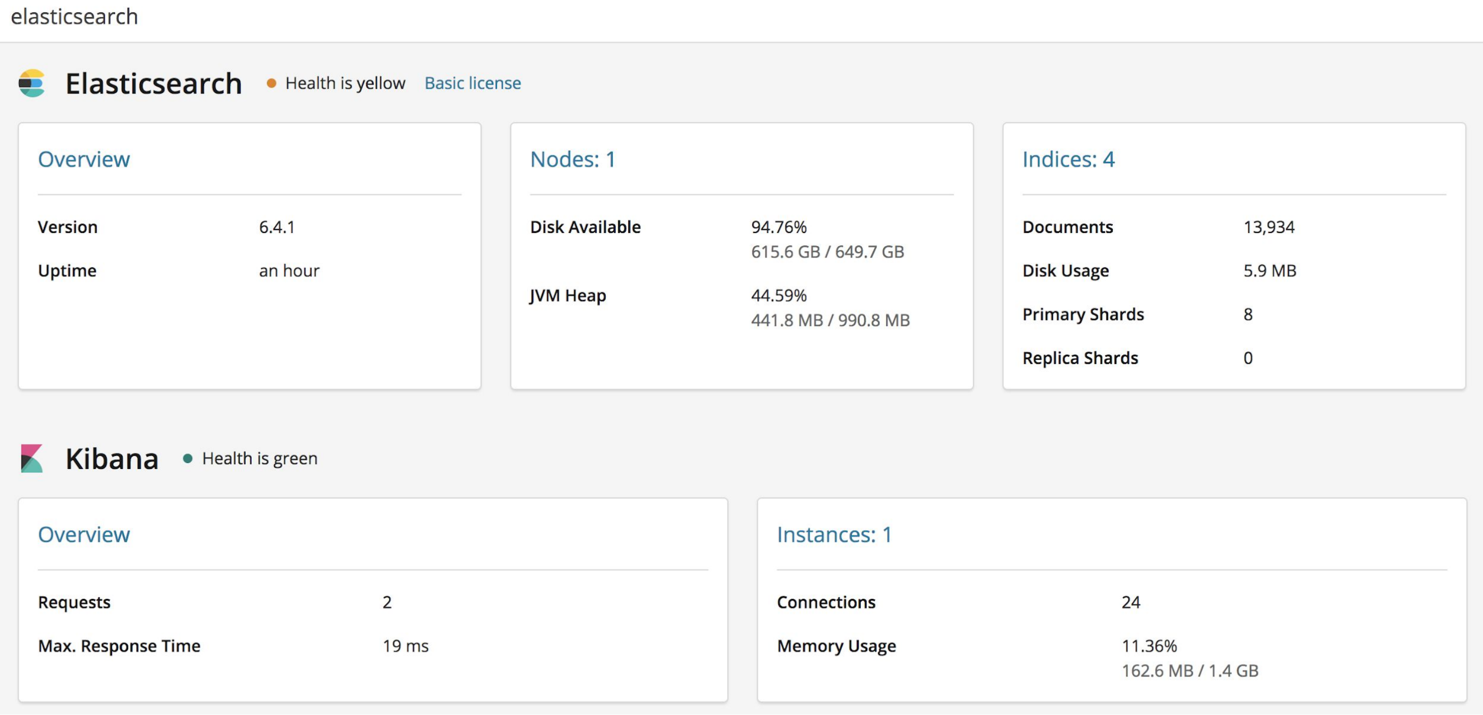
□ Group other values in separate bucket

Step.4 Create Dashboard

1. Click “Dashboard” Menu
2. Click “Create new dashboard”
3. Click “**Add**” button
4. Select vitualization
5. Save Dashboard



Monitoring



elasticsearch

Elasticsearch

● Health is yellow Basic license

Overview

Version	6.4.1
Uptime	an hour

Nodes: 1

Disk Available	94.76% 615.6 GB / 649.7 GB
JVM Heap	44.59% 441.8 MB / 990.8 MB

Indices: 4

Documents	13,934
Disk Usage	5.9 MB
Primary Shards	8
Replica Shards	0

Kibana

● Health is green

Overview

Requests	2
Max. Response Time	19 ms

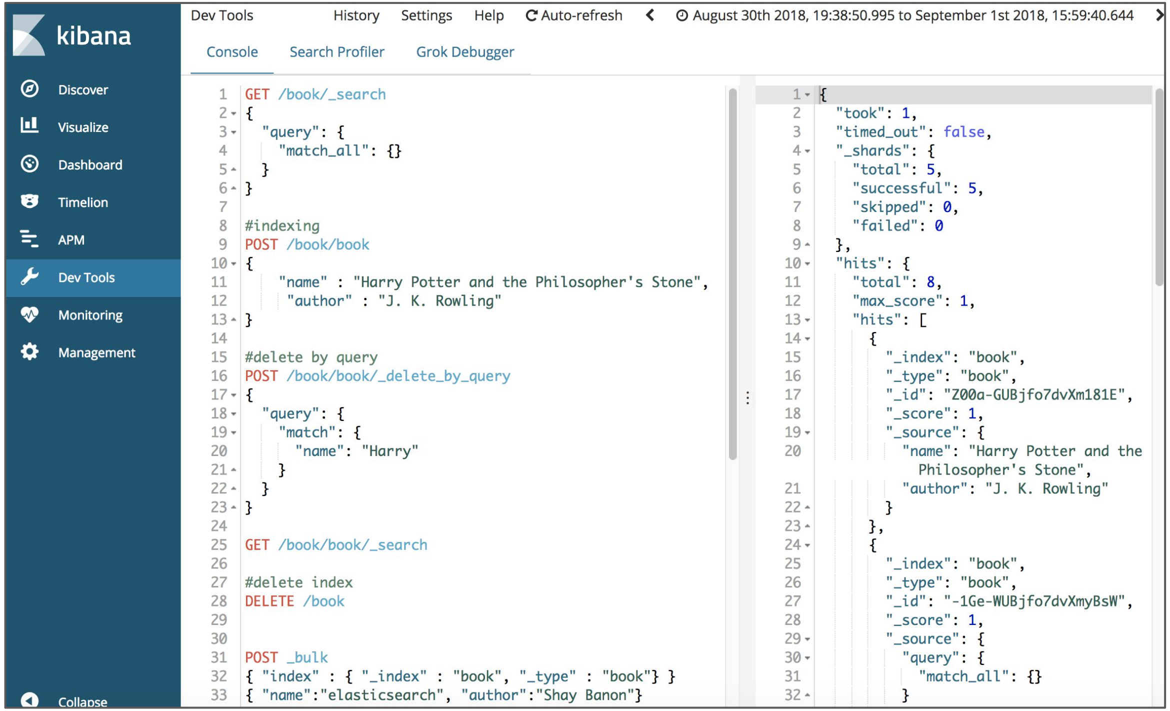
Instances: 1

Connections	24
Memory Usage	11.36% 162.6 MB / 1.4 GB

Dev tools

Provide tools for dev

- Console
- Grok Debugger



```
GET /book/_search
{
  "query": {
    "match_all": {}
  }
}

#indexing
POST /book/book
{
    "name" : "Harry Potter and the Philosopher's Stone",
     "author" : "J. K. Rowling"
}

#delete by query
POST /book/book/_delete_by_query
{
  "query": {
    "match": {
      "name": "Harry"
    }
  }
}

GET /book/book/_search

#delete index
DELETE /book


POST _bulk
{ "index" : { "_index" : "book", "_type" : "book"} }
{ "name":"elasticsearch", "author":"Shay Banon"}
```

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 8,
    "max_score": 1,
    "hits": [
      {
        "_index": "book",
        "_type": "book",
        "_id": "Z00a-GUBjfo7dvXm181E",
        "_score": 1,
        "_source": {
          "name": "Harry Potter and the
            Philosopher's Stone",
          "author": "J. K. Rowling"
        }
      },
      {
        "_index": "book",
        "_type": "book",
        "_id": "-1Ge-WUBjfo7dvXmyBsW",
        "_score": 1,
        "_source": {
          "query": {
            "match_all": {}
          }
```