

Kibana Setup Document

Introduction:

Kibana is an open source data visualization plugin for Elastic-search. It provides visualization capabilities on top of the content indexed on an Elastic-search cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data

Hardware Requirement

RAM	4 GB
Operating System	Ubuntu 18.04(64 bit),16.04, and Centos 7
CPU	2 core
DISK	SSD

Step1: Installation Dependency

(java 8 is required)

Java is required for the Elastic stack deployment. Elasticsearch requires Java 8. It is recommended to use the Oracle JDK 1.8

Install java using following command

```
sudo apt install openjdk-8-jre
```

Install apt repository

```
sudo apt-get install apt-transport-https
```

Install apache2 or Nginx server for ssl setup

```
sudo apt-get install apache2 or sudo apt-get install nginx
```

Install common packages

```
sudo apt-get install curl zip wget
```

Install mysql driver and server

```
sudo apt-get install mysql-client mysql-server
```

Step2: Install and configure Elastic-search

Elasticsearch is an open source search engine based on Lucene, developed in java. It provides a distributed and multi tenant full-text search engine with an HTTP Dashboard web-interface (Kibana) and JSON documents scheme. Elasticsearch is a scalable search engine that can be used to search for all types of documents, including log file. Elasticsearch is the heart of the 'Elastic Stack' or ELK Stack.

In this step, we will install and configure Elasticsearch. Install Elasticsearch from the elastic repository and configure it to run on the localhost IP.

1. Before installing Elasticsearch, add the elastic repository key to the serve

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

2. Save the repository definition to /etc/apt/sources.list.d/elastic-7.x.list:

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```

3. Install the Elasticsearch Debian package with:

```
sudo apt-get update && sudo apt-get install elasticsearch
```

4. Once Elasticsearch is finished installing, use your preferred text editor to edit Elasticsearch's main configuration file, elasticsearch.yml.
Here, we'll use nano:

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

Note: Elasticsearch's configuration file is in YAML format, which means that indentation is very important! Be sure that you do not add any extra spaces as you edit this file.

Elasticsearch listens for traffic from everywhere on port 9200.

Edit `elasticsearch.yml` file which is located at `/etc/elasticsearch/elasticsearch.yml`

```
network.host: localhost or IP
```

Save and close `elasticsearch.yml` by pressing CTRL+X, followed by Y and then ENTER if you're using nano. Then, start the Elasticsearch service with `systemctl`.

```
sudo systemctl start elasticsearch
```

Next, run the following command to enable Elasticsearch to start up every time your server boots:

```
sudo systemctl enable elasticsearch
```

You can test whether your Elasticsearch service is running by sending an HTTP request:

```
curl -X GET "localhost:9200" or curl -X GET "your_server_ip:9200"
```

Step3 : Installing and Configuring the Kibana Dashboard

Kibana is a data visualization interface for Elasticsearch. Kibana provides a pretty dashboard (web interfaces), it allows you to manage and visualize all data from Elasticsearch on your own. It's not just beautiful, but also powerful.

In this step, we will install and configure Kibana

Install Kibana with this apt command:

```
sudo apt-get install -y kibana
```

Now edit the `kibana.yml` configuration file.

```
nano /etc/kibana/kibana.yml
```

Uncomment the server.port, server.hos and elasticsearch.url lines.

```
server.port: 5601
server.host: "localhost"
elasticsearch.url: "http://localhost:9200" or ip:9200
```

Save the file and exit vim.

```
sudo systemctl enable kibana
sudo systemctl start kibana
```

Kibana will run on port 5601 as node application.

Step4: Installing and Configuring the Logstash

In this step, we will install and configure Logstash to centralize server logs from client sources with filebeat, then filter and transform all data (Syslog) and transport it to the stash (Elasticsearch).

Install logstash using following command

```
sudo apt-get install -y logstash
```

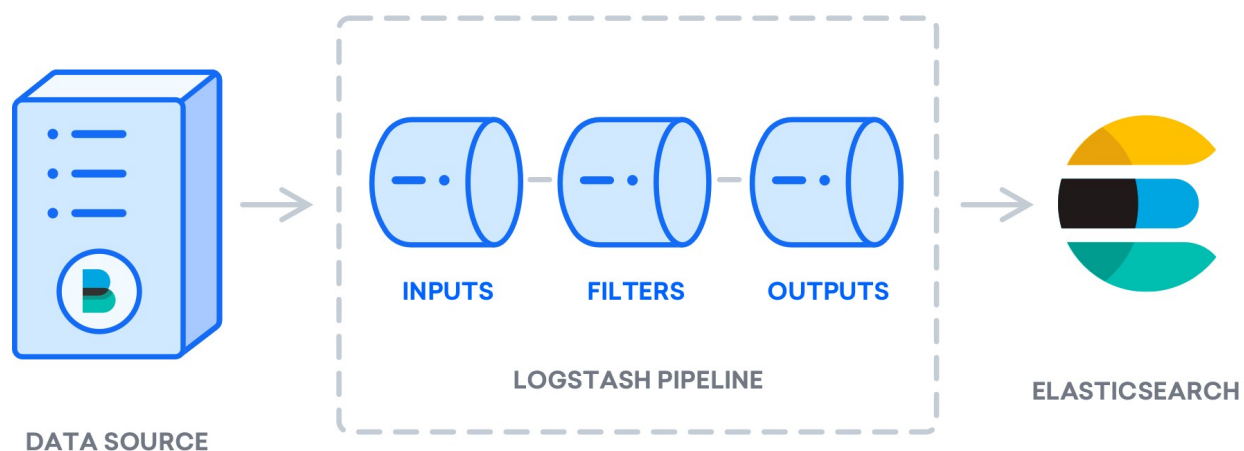
Configure /etc/logstash/logstash.yml file

Open this file and find http.host

update with your IP

```
http.host : 127.0.0.1 # server_ip
```

Pictorial representation of logstash



Logstash configuration files are in the JSON-format, and reside in **/etc/logstash/conf.d**. The configuration consists of three sections:

- input.conf
- filter.conf
- output.conf

configuration file looks like as followed

Input configuration:

```
input {  
  beats {  
    port => 5044  
  }  
}
```

Filter Configuration:

This filter is used to parse incoming system logs to make them structured and usable by the predefined Kibana dashboards

```
filter {

  if [fileset][module] == "system" {
    if [fileset][name] == "auth" {
      grok {
        match => { "message" => ["%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\\[%{POSINT:[system][auth][pid]}\\)?: %{DATA:[system][auth][ssh][event]} %{DATA:[system][auth][ssh][method]} for (invalid user)? %{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]} port %{NUMBER:[system][auth][ssh][port]} ssh2(: %{GREEDYDATA:[system][auth][ssh][signature]})?",
          "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\\[%{POSINT:[system][auth][pid]}\\)?: %{DATA:[system][auth][ssh][event]} user %{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]}",
          "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\\[%{POSINT:[system][auth][pid]}\\)?: Did not receive identification string from %{IPORHOST:[system][auth][ssh][dropped_ip]}",
          "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sudo(?:\\[%{POSINT:[system][auth][pid]}\\)?: \\s*%{DATA:[system][auth][user]} :( %{DATA:[system][auth][sudo][error]} ;)? TTY=%{DATA:[system][auth][sudo][tty]} ; PWD=%{DATA:[system][auth][sudo][pwd]} ; USER=%{DATA:[system][auth][sudo][user]} ; COMMAND=%{GREEDYDATA:[system][auth][sudo][command]}",
          "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} groupadd(?:\\[%{POSINT:[system][auth][pid]}\\)?: new group: name=%{DATA:system.auth.groupadd.name}, GID=%{NUMBER:system.auth.groupadd.gid}",
          "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} useradd(?:\\[%{POSINT:[system][auth][pid]}\\)?: new user: name=%{DATA:[system][auth][user][add][name]}, UID=%{NUMBER:[system][auth][user][add][uid]}, GID=%{NUMBER:[system][auth][user][add][gid]}, home=%{DATA:[system][auth][user][add][home]}, shell=%{DATA:[system][auth][user][add][shell]}$",
          "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} %{DATA:[system][auth][program]}(?:\\[%{POSINT:[system][auth][pid]}\\)?: %{GREEDYMULTILINE:[system][auth][message]}" ] }

        pattern_definitions => {
          "GREEDYMULTILINE"=> "(.\\n)*"
        }
      }
    }
  }
}
```

```

    remove_field => "message"
  }
  date {
    match => [ "[system][auth][timestamp]", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
  }
  geoip {
    source => "[system][auth][ssh][ip]"
    target => "[system][auth][ssh][geoip]"
  }
}
else if [fileset][name] == "syslog" {
  grok {
    match => { "message" => ["%{SYSLOGTIMESTAMP:[system][syslog][timestamp]} %
{SYSLOGHOST:[system][syslog][hostname]} %{DATA:[system][syslog][program]}(?:\%
{POSINT:[system][syslog][pid]}\)??: %{GREEDYMULTILINE:[system][syslog][message]}"] }
    pattern_definitions => { "GREEDYMULTILINE" => "(.|\n)*" }
    remove_field => "message"
  }
  date {
    match => [ "[system][syslog][timestamp]", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
  }
}
}
}
}

```

Output Configuration:

Output configuration. Essentially, this output configures Logstash to store the Beats data in Elasticsearch, which is running at localhost:9200, in an index named after the Beat used

- output.conf

```

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%
{+YYYY.MM.dd}"
  }
}

```

Test Logstash configuration with this command:

```
sudo -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t
```

output will be Configuration OK

If your configuration test is successful, start and enable Logstash to put the configuration changes into effect:

```
sudo systemctl start logstash  
sudo systemctl enable logstash
```

Step4: Installing and Configuring Filebeat

The Elastic Stack uses several lightweight data shippers called Beats to collect data from various sources and transport them to Logstash or Elasticsearch

Filebeat: collects and ships log files.

Install Filebeat using following command:

```
sudo apt install filebeat
```

Configure Filebeat to connect to Logstash. Open the Filebeat configuration file:

```
sudo nano /etc/filebeat/filebeat.yml
```

comment out the following lines by preceding them with a #;

```
...  
#output.elasticsearch:  
# Array of hosts to connect to.  
#hosts: ["localhost:9200"]  
...
```


Uncomment the lines `output.logstash:` and `hosts: ["localhost:5044"]` by removing the `#`

```
output.logstash:  
  hosts: ["localhost:5044"]
```

Let's enable it:

```
sudo filebeat modules enable system
```

You can see a list of enabled and disabled module:

```
sudo filebeat modules enable system
```

To load the template, use the following command:

```
sudo filebeat setup --template -E output.logstash.enabled=false -E  
'output.elasticsearch.hosts=["localhost:9200"]'
```

Output

Loaded index template

Now you can start and enable Filebeat:

```
sudo systemctl start filebeat  
sudo systemctl enable filebeat
```

To verify that Elasticsearch is indeed receiving this data, query the Filebeat index with this command:

```
curl -XGET 'http://localhost:9200/filebeat-*/_search?pretty'
```

Step5 : Sync and configure Mysql with logstash

Create mysql.conf file inside etc/logstash/mysql.conf

Insert following code:

Note : download location : <https://dev.mysql.com/downloads/connector/j/>

/usr/share/java/mysql-connector-java-8.0.16.jar → this represent path of jdbc file

jdbc:mysql://localhost:3306/kibana → here kibana is database name

```
input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-java-8.0.16.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/kibana"
    jdbc_user => "root"
    jdbc_password => "password"
    statement => "SELECT * from kibana"
    jdbc_validate_connection => true
    tracking_column => "id"
    use_column_value => true
    tracking_column_type => "numeric"
    clean_run => true
    jdbc_pool_timeout => 10
    jdbc_paging_enabled => true
    jdbc_page_size => 10000
  }
}
output{
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "kibana_dashboard" // name of index
    user => "elastic"
    password => "password"
  }
}
stdout { codec => rubydebug { metadata => true } }
# stdout { codec => dots }
}
```

Finally, you can run logstash command which will pull all mysql data and create index for you to create custom kibana dashboard.

Command to run logstash :

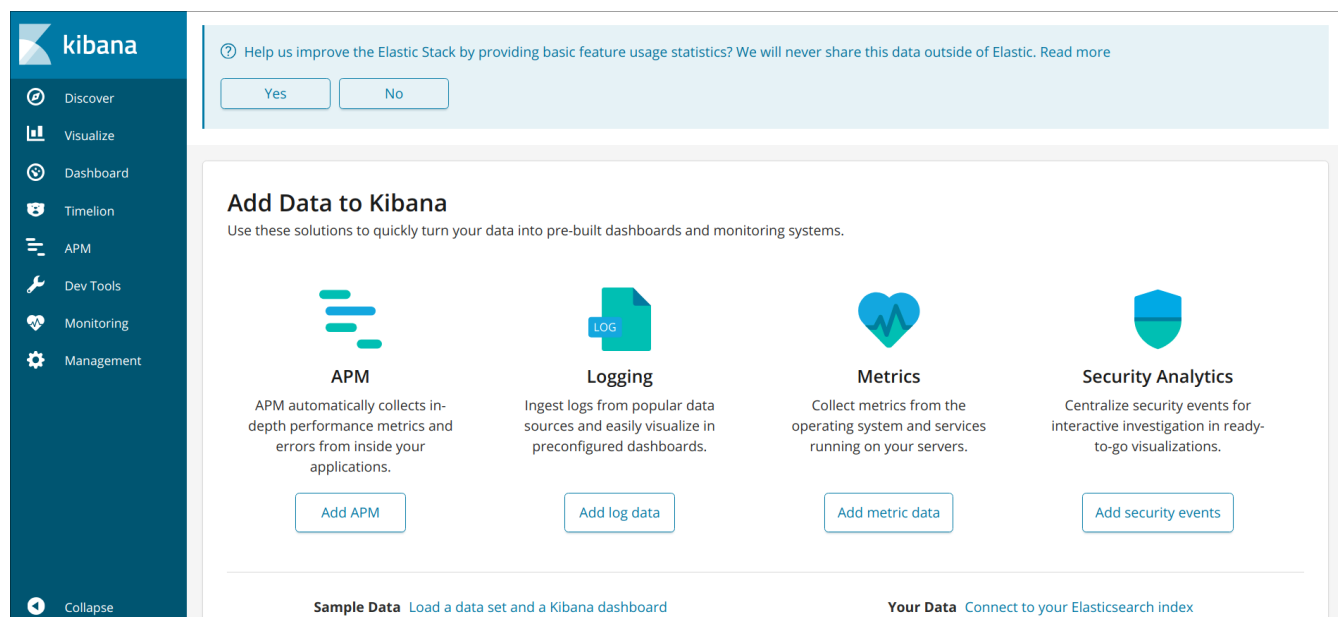
```
/usr/share/logstash/bin/logstash -f /etc/logstash/mysql.conf
```

Above command will create log file of mysql data.

Kibana is all set now. You can browse kibana dashboard using url :


localhost:5601 or server_ip:5601

Exploring Kibana Dashboards



Go to management menu → index patter → create index pattern

Following page will be appear

 **Elasticsearch**

Index Management

Index Lifecycle Policies


Rollup Jobs

Cross Cluster Replication

Remote Clusters

License Management

8.0 Upgrade Assistant

 **Kibana**

[Index Patterns](#)

Saved Objects

Spaces

Reporting

Advanced Settings

Management / Create index pattern

Create index pattern

★ kibana_sample_data_logs

filebeat-*

kibana*

kibana_dashboard

mykibana

Create index pattern

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

☐ Include system indices

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 \, /, ?, ", <, >, |.

> Next step

Your index pattern can match any of your **10 indices**, below.

filebeat-7.1.1-2019.06.07

filebeat-7.1.1-2019.06.10

filebeat-7.1.1-2019.06.11

filebeat-7.1.1-2019.06.12

filebeat-7.1.1-2019.06.13

filebeat-7.1.1-2019.06.14

filebeat-7.1.1-2019.06.17

kibana_dashboard

kibana_sample_data_logs

mykibana

Rows per page: 10 ▾

Search your index and click on next

Configure filter and click on create Index Pattern

Management / Create index pattern

Elasticsearch

- Index Management
- Index Lifecycle Policies
- Rollup Jobs
- Cross Cluster Replication
- Remote Clusters
- License Management
- 8.0 Upgrade Assistant

Kibana

- Index Patterns
- Saved Objects
- Spaces
- Reporting
- Advanced Settings

Create index...

- ★ kibana_sample_...
- filebeat-*
- kibana*
- kibana_dashboard
- mykibana

Create index pattern

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

☐ Include system indices

Step 2 of 2: Configure settings

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

Time Filter field name [Refresh](#)

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.

[Show advanced options](#)

[< Back](#) [Create index pattern](#)

Now Click on Visualization menu

Visualize

Search...

+

1-20 of 157

Title ↑	Type
<input type="checkbox"/> [Logs] File Type Scatter Plot	</> Vega
<input type="checkbox"/> [Logs] Goals	📊 Gauge
<input type="checkbox"/> [Logs] Heatmap	🔗 Heat Map
<input type="checkbox"/> [Logs] Host, Visits and Bytes Table	📊 Visual Builder
<input type="checkbox"/> [Logs] Input Controls	⚙️ Controls
<input type="checkbox"/> [Logs] Markdown Instructions	📄 Markdown
<input type="checkbox"/> [Logs] Response Codes Over Time + Annotations	📊 Visual Builder
<input type="checkbox"/> [Logs] Source and Destination Sankey Chart	</> Vega
<input type="checkbox"/> [Logs] Unique Visitors by Country	🌐 Region Map
<input type="checkbox"/> [Logs] Unique Visitors vs. Average Bytes	📊 Area
<input type="checkbox"/> [Logs] Visitors by OS	🥞 Pie
<input type="checkbox"/> Access logs over time [Filebeat Nginx] ECS	📊 Visual Builder
<input type="checkbox"/> Access map [Filebeat IIS] ECS	📍 Coordinate Map
<input type="checkbox"/> Access Map [Filebeat Nginx] [ML] ECS	📍 Coordinate Map
<input type="checkbox"/> Access Map [Filebeat Nginx] ECS	📍 Coordinate Map

Click on + icon to create your visualization

Select Visualization type

The screenshot shows a dashboard interface with a 'Visualize' tab selected. A 'New Visualization' modal is open, displaying a grid of 16 visualization types: Area, Controls, Coordinate Map, Data Table, Gauge, Goal, Heat Map, Horizontal Bar, Line, Markdown, Metric, Pie, Region Map, Tag Cloud, Timellon, and Vega. A red annotation 'select any visualization type which you want to plot' points to the grid. The modal also includes a search filter and a section titled 'Select a visualization type' with the instruction 'Start creating your visualization by selecting a type for that visualization.'

New Visualization

Search Filter

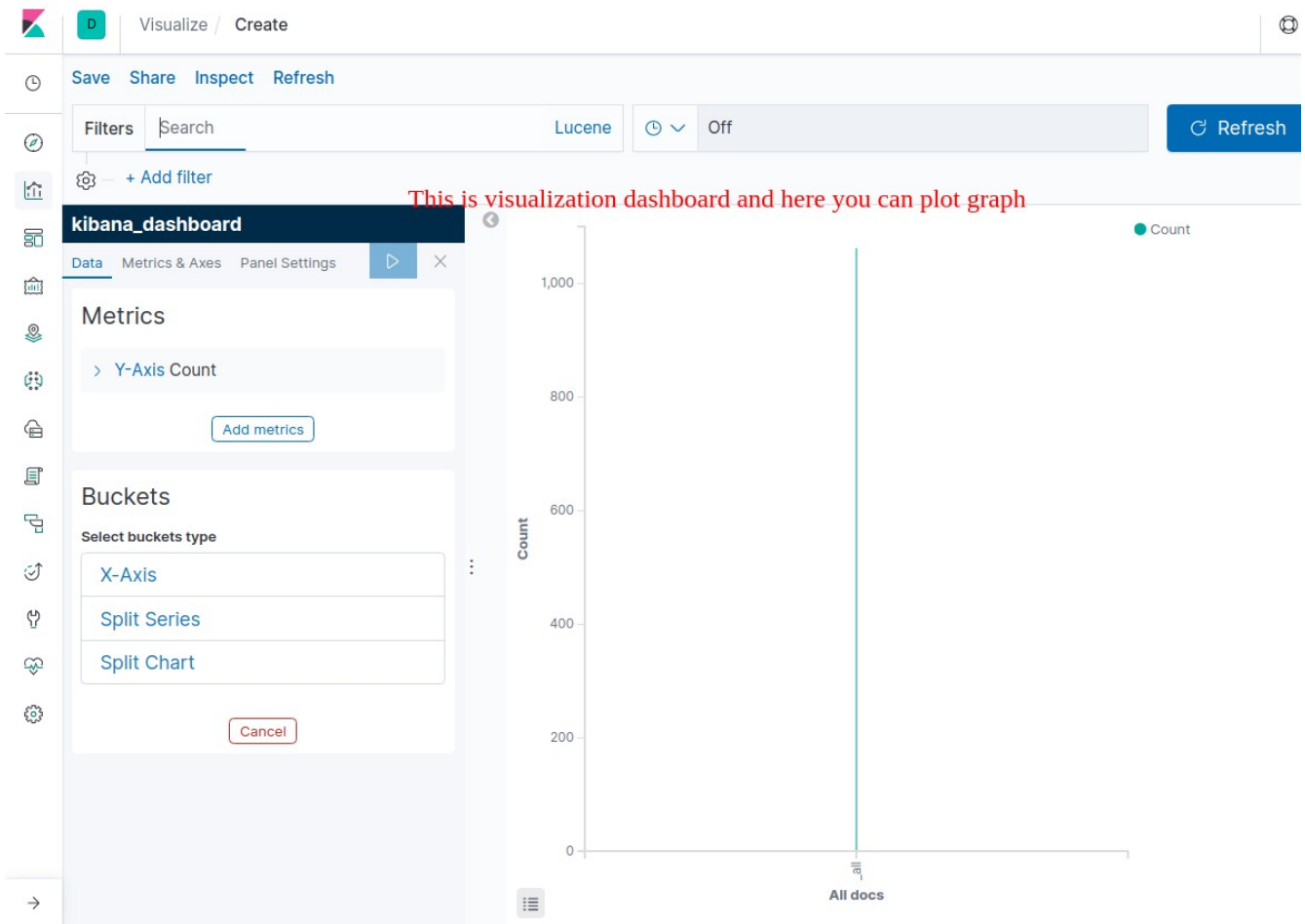
Select a visualization type

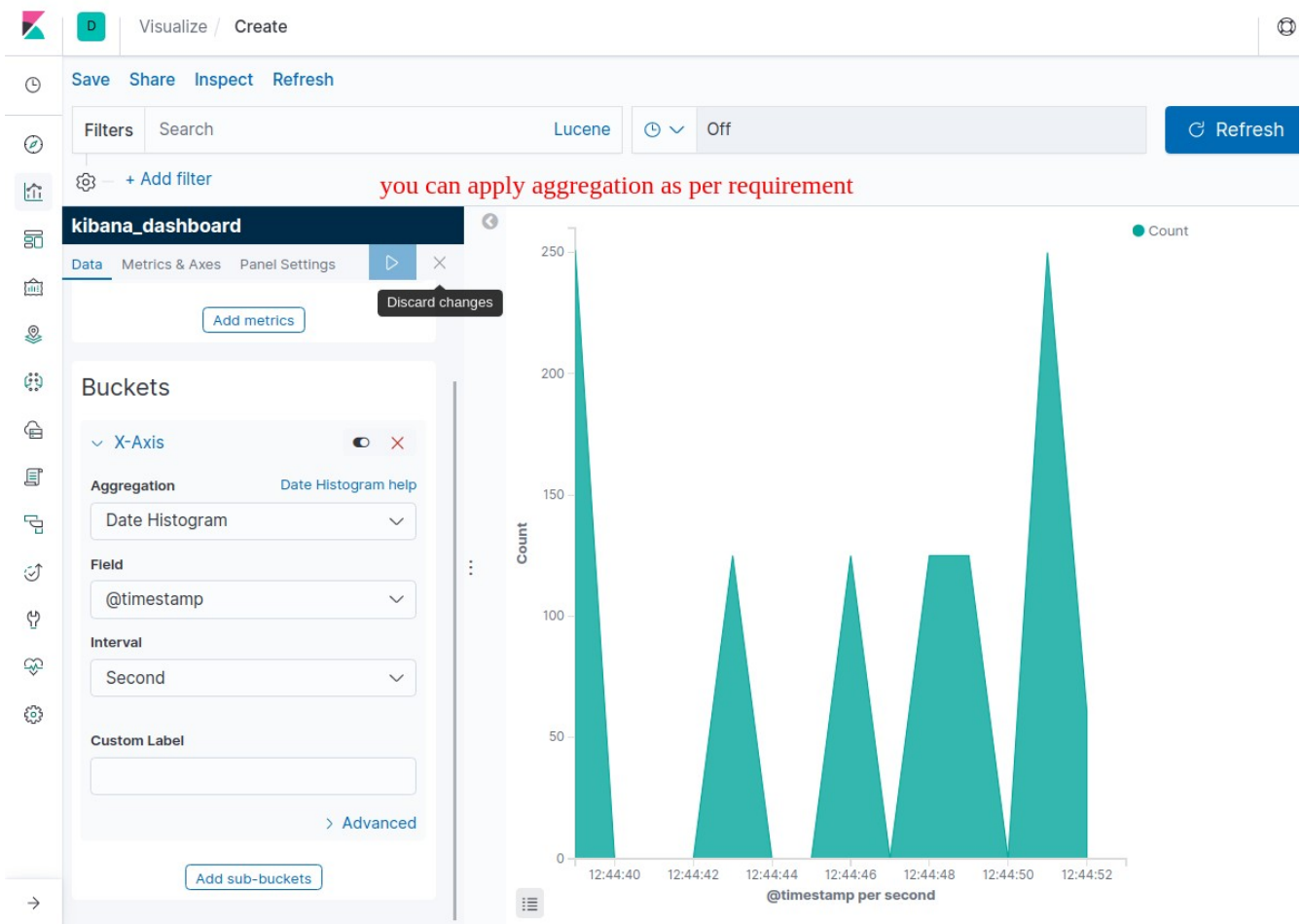
Start creating your visualization by selecting a type for that visualization.

select any visualization type which you want to plot

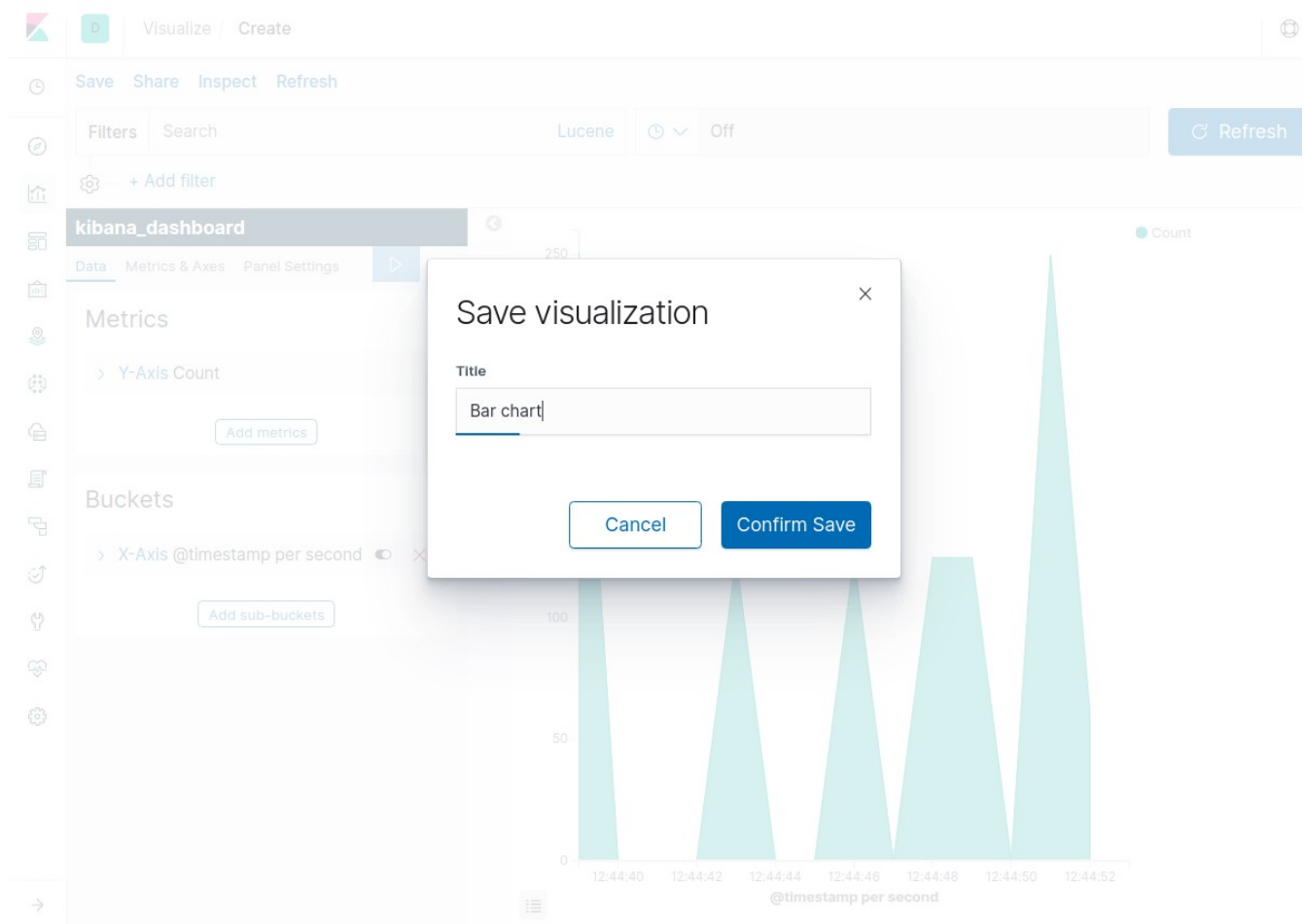
Area, Controls, Coordinate Map, Data Table, Gauge, Goal, Heat Map, Horizontal Bar, Line, Markdown, Metric, Pie, Region Map, Tag Cloud, Timellon, Vega

Click on visualization type and choose your index





Now click now save button to save graph



Once you done. Now you create your dashboard.

Goto dashboard menu then following page will appear

D

Dashboards

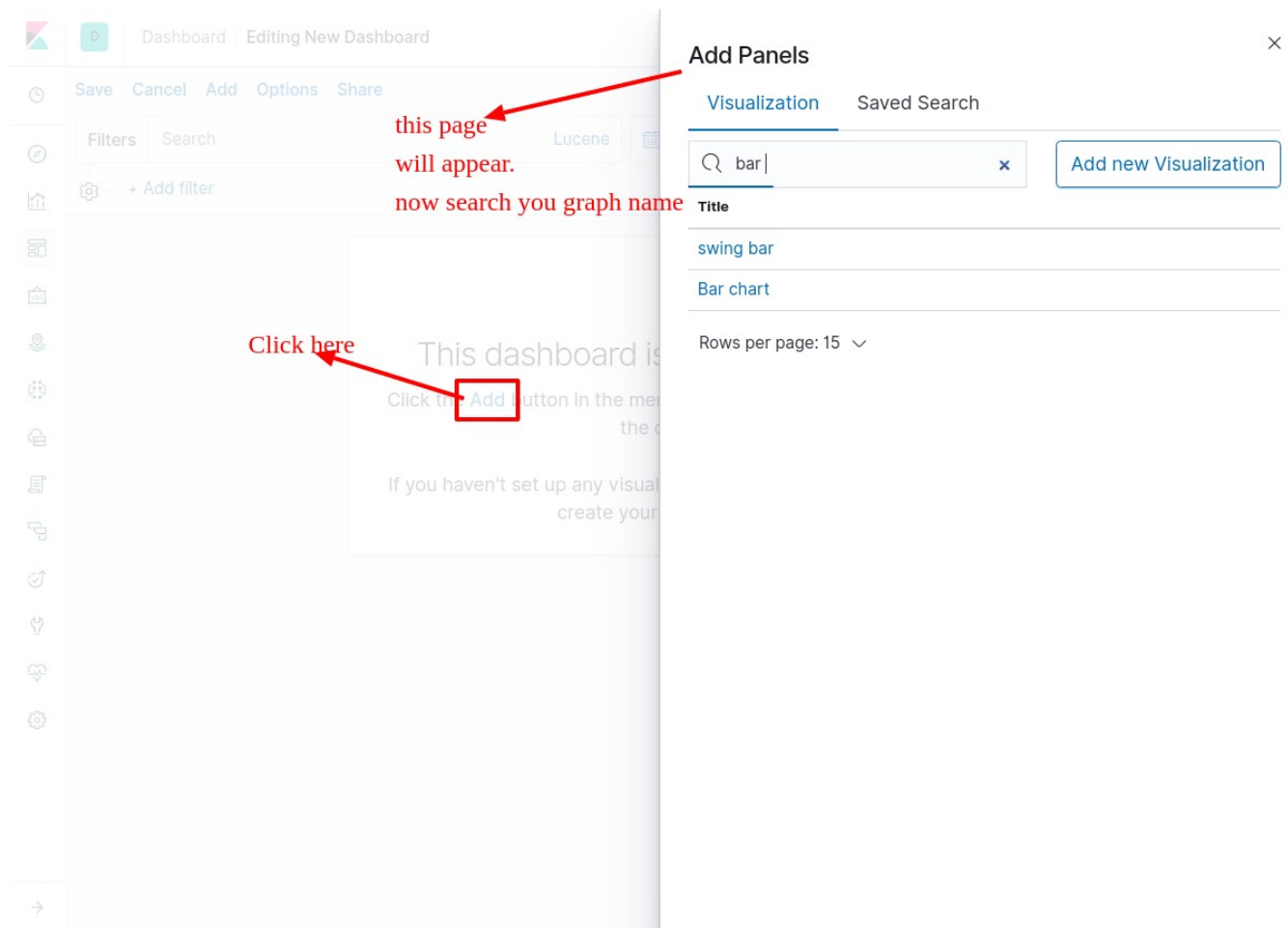
Click here

Create new dashboard

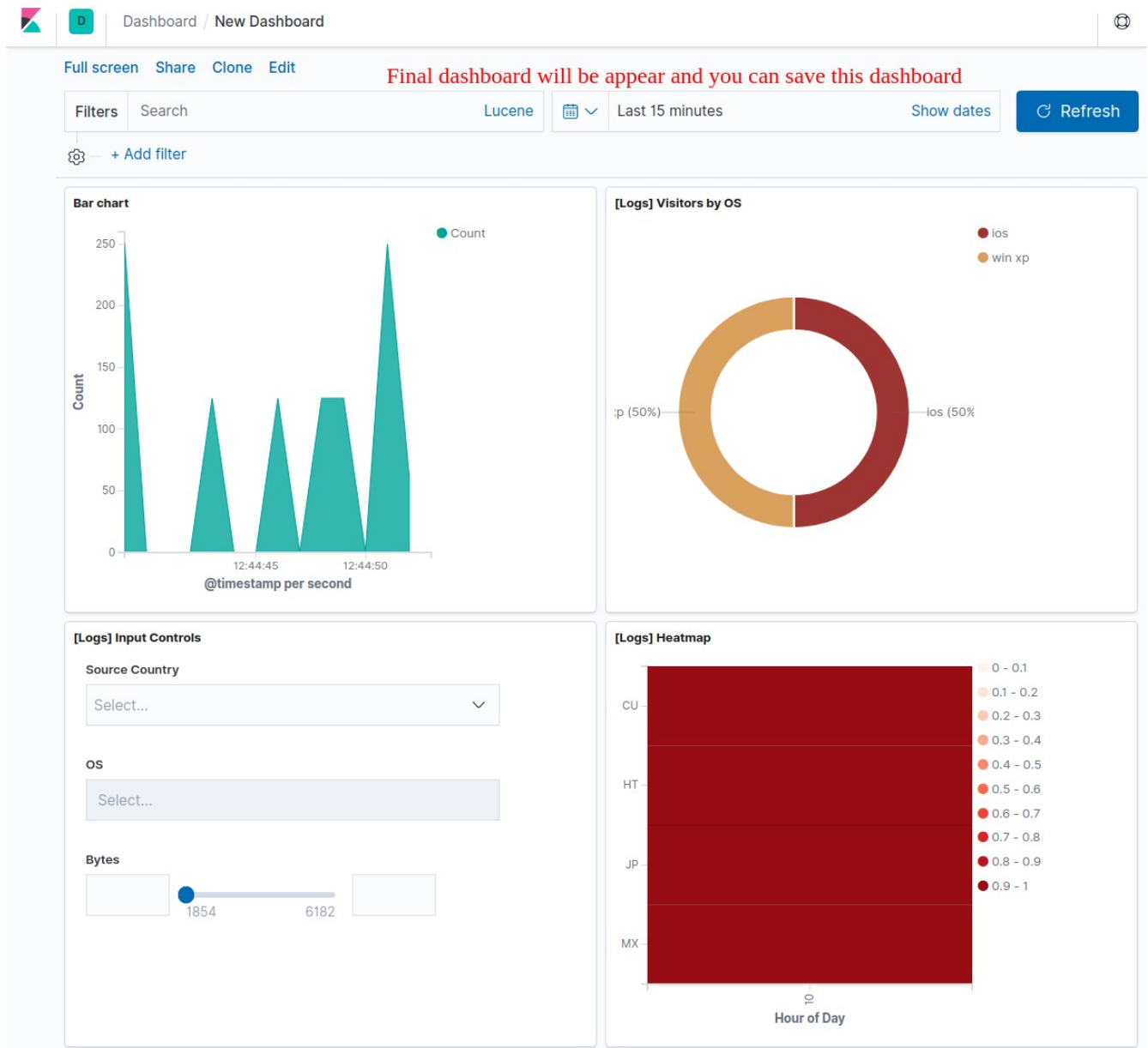
Search...

<input type="checkbox"/> Title ↑	Description	Actions
<input type="checkbox"/> [Filebeat Apache] Access and error logs ECS	Filebeat Apache module dashboard	Edit
<input type="checkbox"/> [Filebeat Auditd] Audit Events ECS	Dashboard for the Auditd Filebeat module	Edit
<input type="checkbox"/> [Filebeat HAProxy] Overview ECS	Filebeat HAProxy module dashboard	Edit
<input type="checkbox"/> [Filebeat Icinga] Debug Log ECS	Filebeat Icinga module dashboard for the debug logs	Edit
<input type="checkbox"/> [Filebeat Icinga] Main Log ECS	Filebeat Icinga module dashboard for the main log files	Edit
<input type="checkbox"/> [Filebeat Icinga] Startup Errors ECS	Filebeat Icinga module dashboard for startup errors	Edit
<input type="checkbox"/> [Filebeat IIS] Access and error logs ECS	Dashboard for the Filebeat IIS module	Edit
<input type="checkbox"/> [Filebeat Iptables] Overview ECS	Overview of the iptables events dashboard.	Edit
<input type="checkbox"/> [Filebeat Iptables] Ubiquiti Firewall Overview ECS	Overview of the Ubiquiti Firewall iptables events dashboard.	Edit
<input type="checkbox"/> [Filebeat Kafka] Overview ECS	Filebeat Kafka module dashboard	Edit
<input type="checkbox"/> [Filebeat MySQL] Overview ECS	Overview dashboard for the Filebeat	Edit

Click on create new dashboard



As soon as you start selecting visualization, Your dashboard will create and this will look like the following image.



Thank you !

Kundan roy