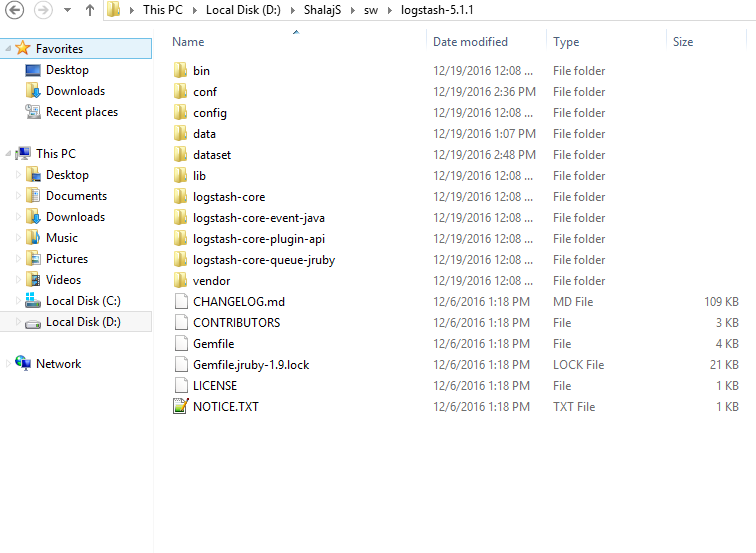
Download logstash from link below

<https://www.elastic.co/downloads/logstash>

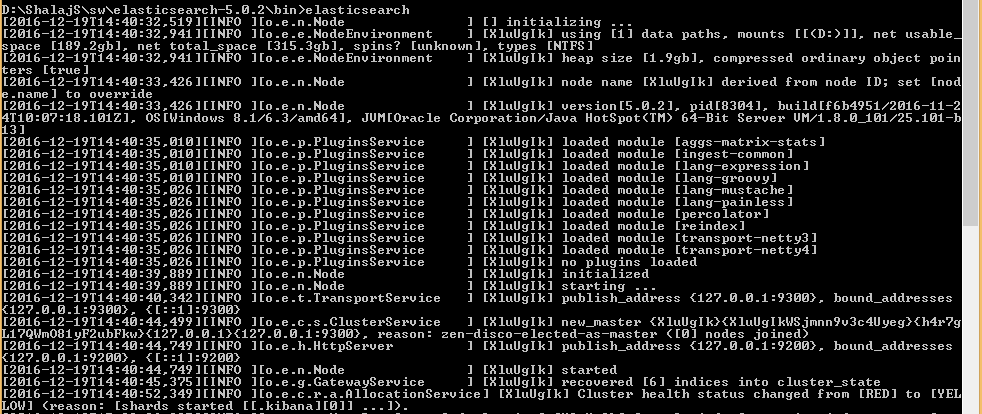
Unzipped logstash file



In case of windows set logstash path till bin directory in path variable

Here we will read a file from system and parse it using filter and later will store it in elastic search

First start elasticsearch



Now create a sample.log file with below content and save it in some location

|  |
| --- |
| 55.3.244.1 GET /index.html 15824 0.043  55.3.244.2 POST /content.html 12345 0.054  55.3.244.3 GET /main.html 23414 0.065  55.3.244.4 PUT /about.html 34527 0.074  55.3.244.4 PUT /about.html 34527 0.074 |

Now create a logstash.conf file with below content and save it some location

|  |
| --- |
| input {  file {  path => "D:\ShalajS\sw\logstash-5.1.1\dataset\sample1.log"  start\_position => beginning  sincedb\_path => "/dev/null"  ignore\_older => 0  }  }  filter {  grok {  match => { "message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}" }  }  }  output {    #stdout { codec => json }  elasticsearch {  hosts => ["localhost:9200"]  document\_type => "sample"  index => "sample\_index"  }    } |

Under input section path is the location of file

By default logstash can’t read one file multiple time so we need to specify sincedb\_path property for testing purpose

Under filter section we are parsing log file using grok filter and creating different variables like client, method, request, bytes and duration

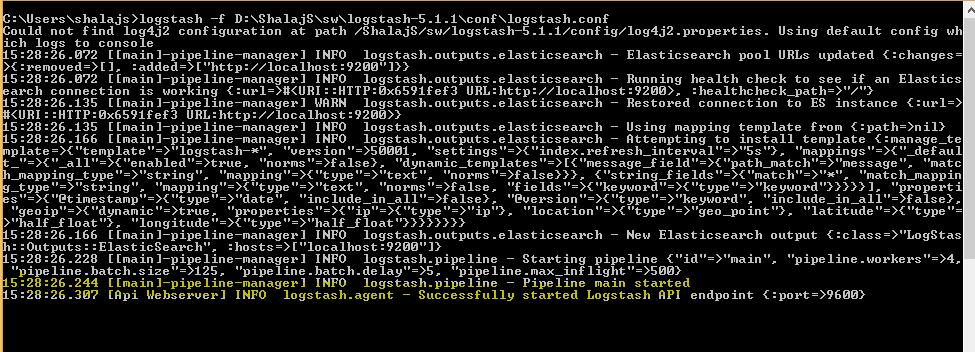
Under output section we are loading this data to elasticsearch under index(schema) **sample**\_index and type(table) **sample**

We can also use stdout to print on log

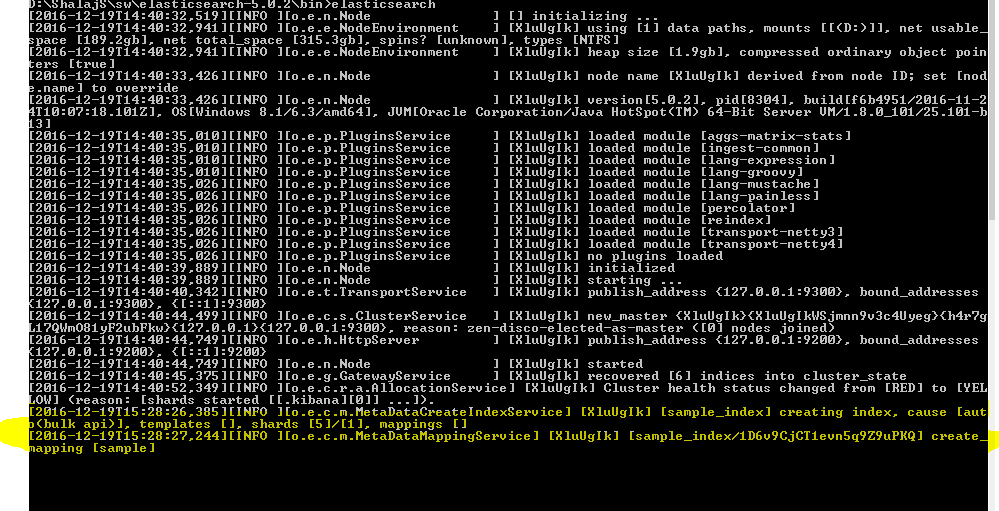
Now run the logstash using below command

|  |
| --- |
| logstash -f D:\ShalajS\sw\logstash-5.1.1\conf\logstash.conf |

We need to define path of our conf file here



Now if you check elasticsearch log, you will find new REST request

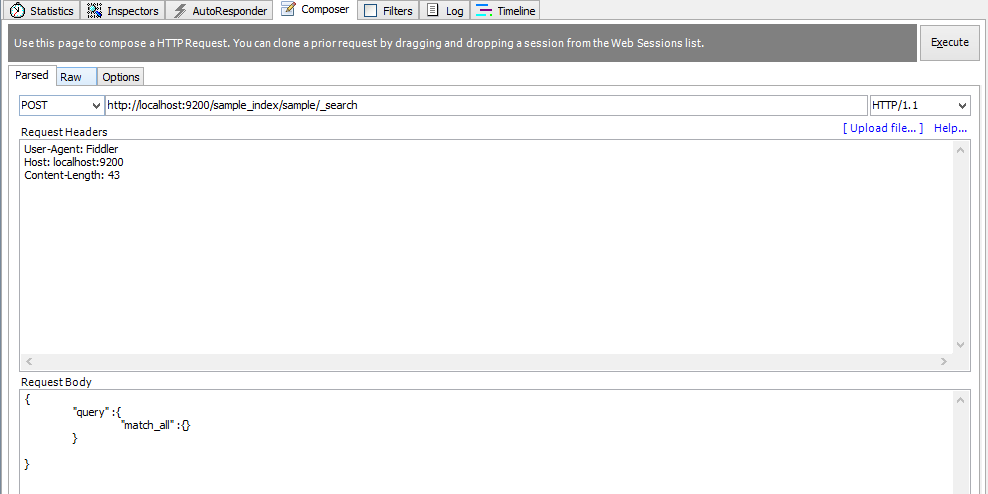


We can check if new index and type is crated in elasticsearch by executing below url in fiddler

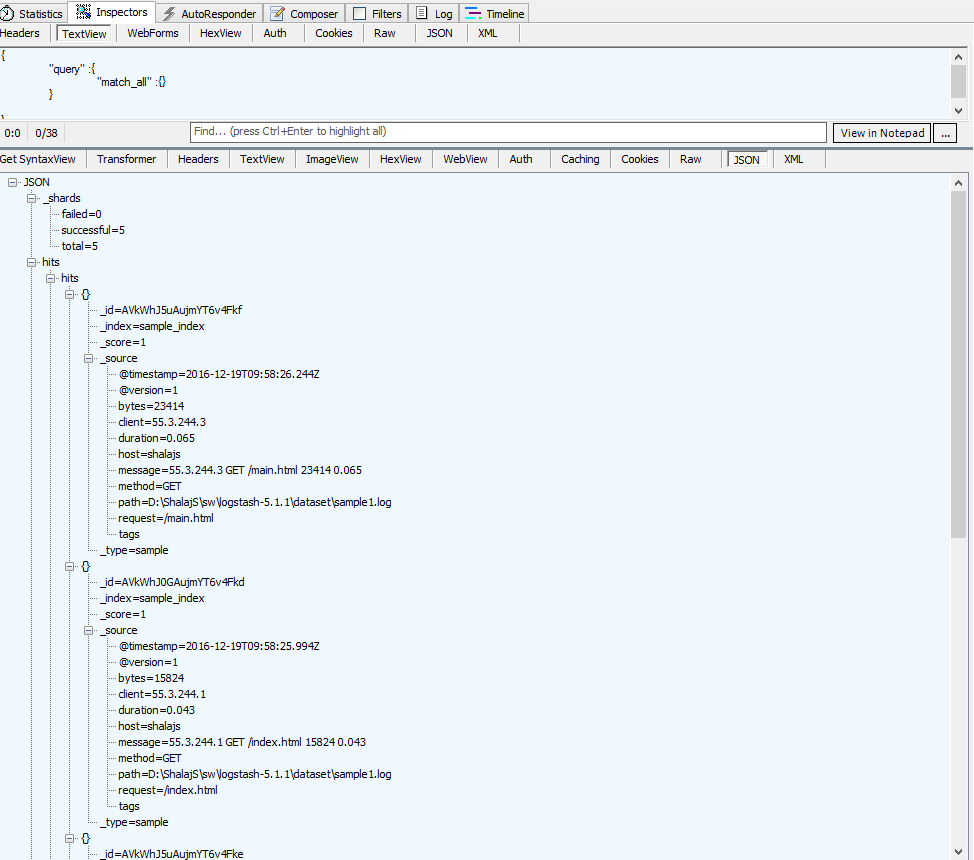
<http://localhost:9200/sample_index/sample/_search>

And the request body should include

|  |
| --- |
| {  "query" :{  "match\_all" :{}  }    } |



Now check http output

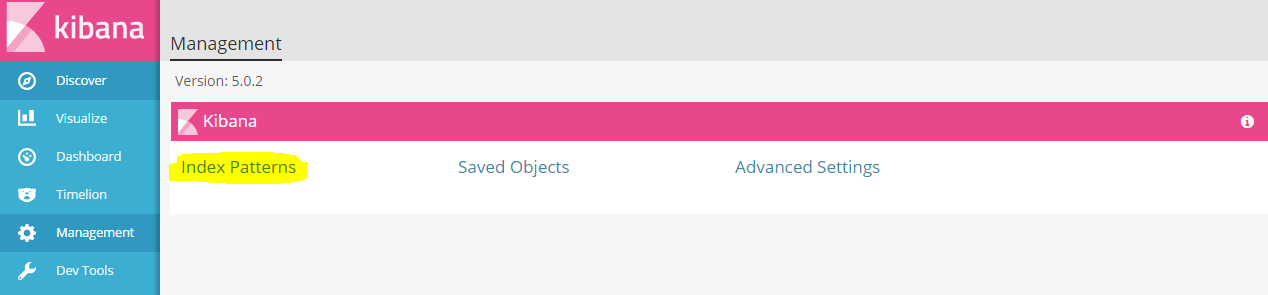


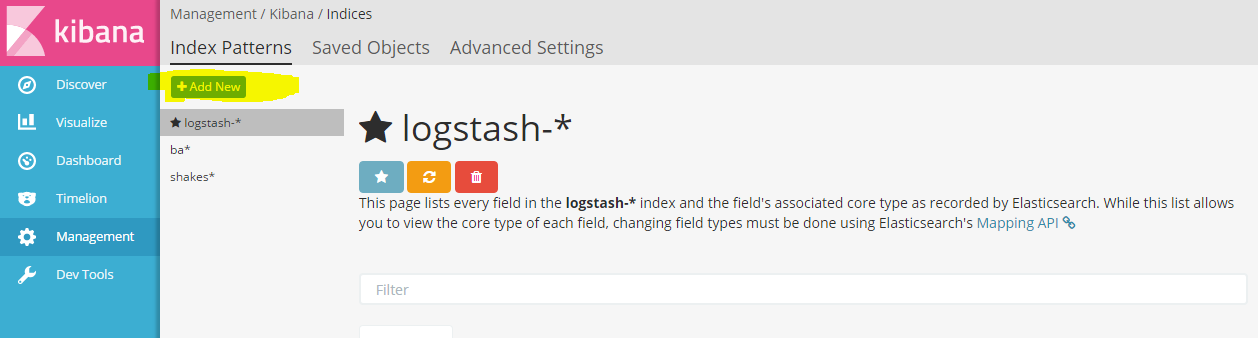
All 5 rows are loaded successfully we can see the values of client, method, request, bytes and duration

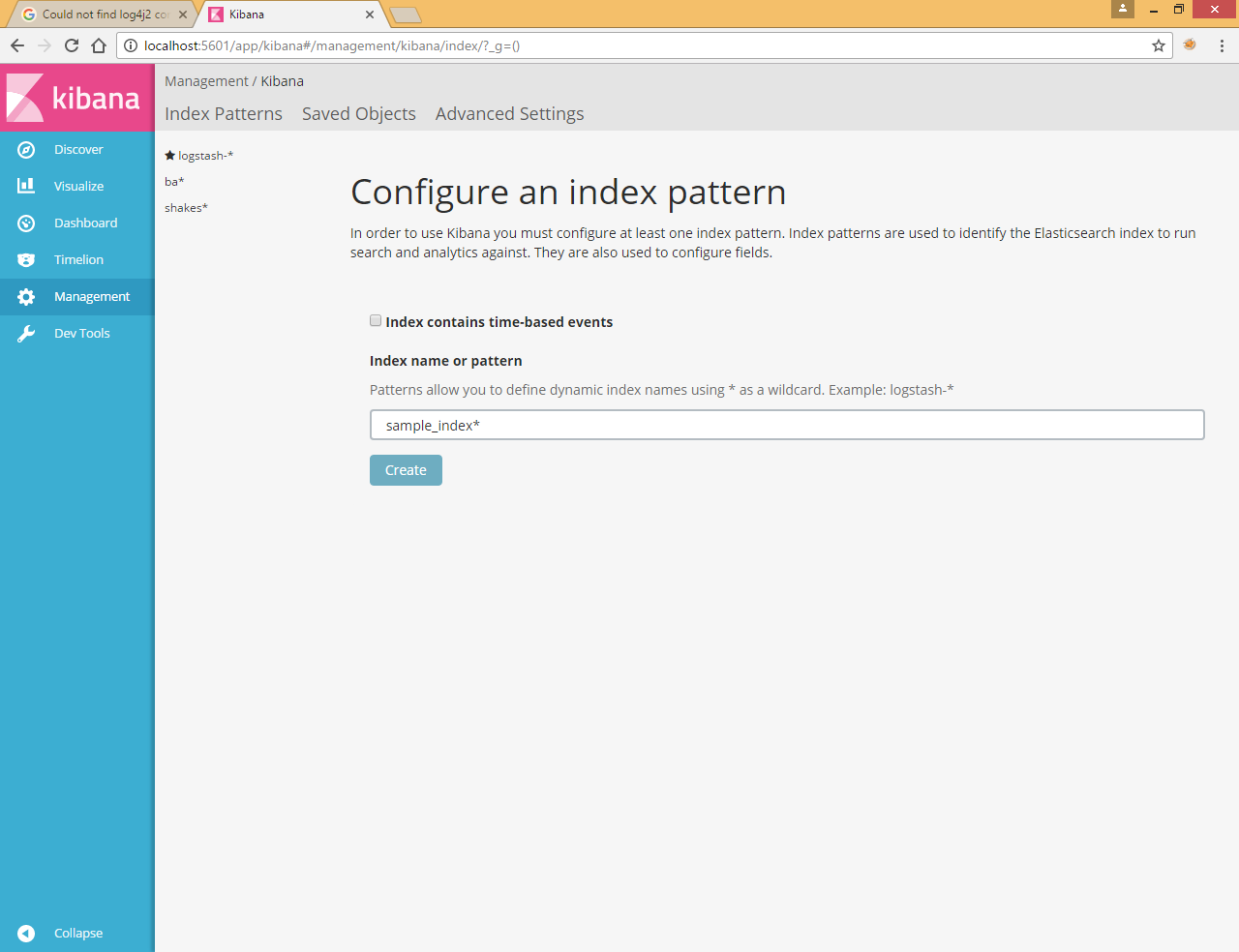
Now start Kibana

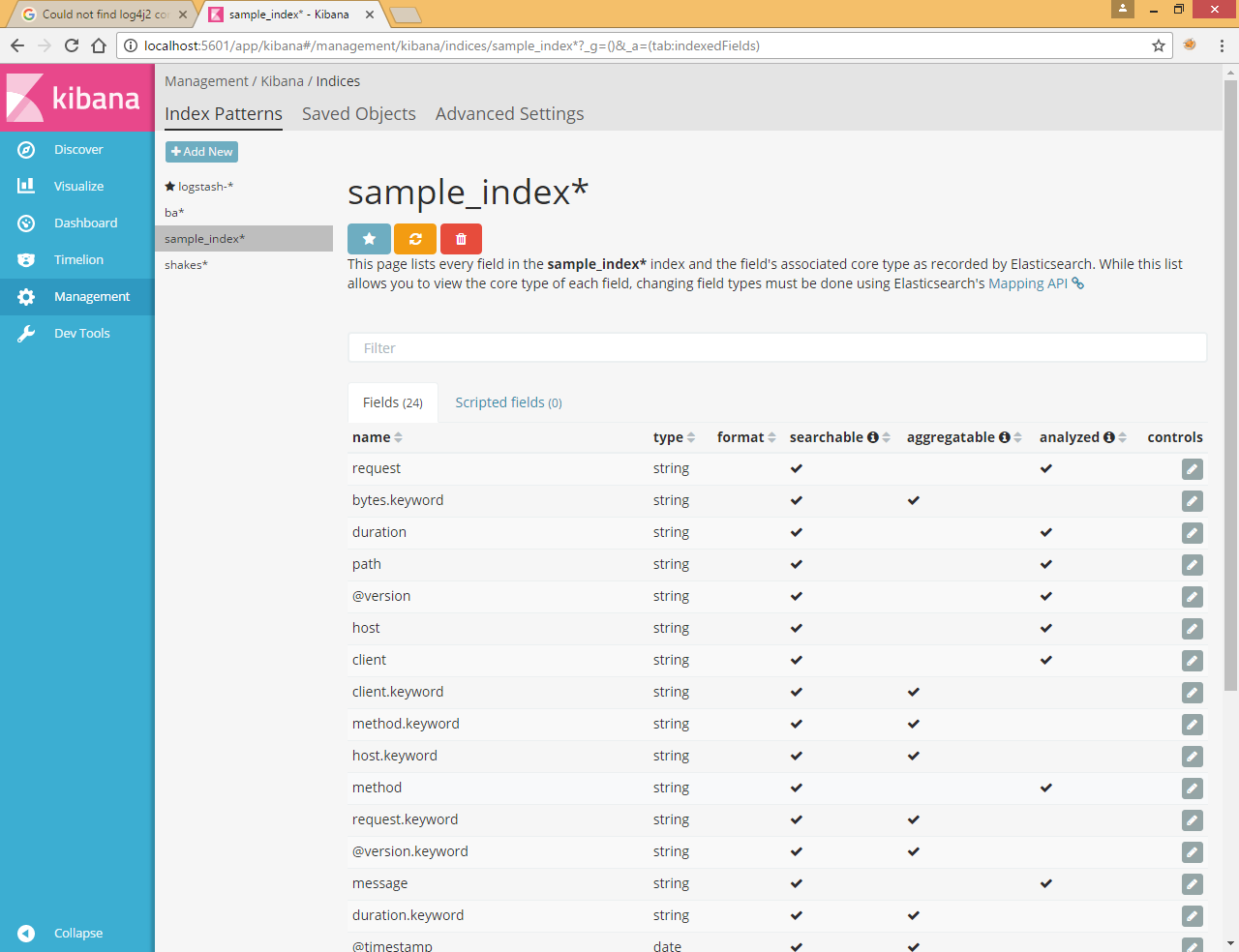


And create new index pattern for sample\_index









And later you can create visualization and dashboard in kibana