

# HOW TO LOAD CSV FILE INTO ELASTICSEARCH WITH LOGSTASH



Here we show how to load CSV data into Elasticsearch using Logstash.

The file we use is network traffic. There are no heading fields, so we will add them.

*(This article is part of our [ElasticSearch Guide](#). Use the right-hand menu to navigate.)*

## Download and Unzip the Data

Download [this file](#) eecs498.zip from Kaggle. Then unzip it. The resulting file is conn250K.csv. It has 256,670 records.

Next, change permissions on the file, since the permissions are set to no permissions.

```
chmod 777 conn250K.csv
```

Now, create this logstash file **csv.config**, changing the path and server name to match your environment.

```
input {  
  file {  
    path => "/home/ubuntu/Documents/esearch/conn250K.csv"  
    start_position => "beginning"  
  }  
}  
  
filter {
```

```
    csv {
      columns =>
    }
  }
}
```

```
output {
  elasticsearch {
    hosts =>
    index => "network"
  }
}
```

Then start logstash giving that config file name.

```
sudo bin/logstash -f config/csv.conf
```

While the load is running, you can list some documents:

```
curl XGET http://parisx:9200/network/_search?pretty
```

results in:

```
"_index" : "network",
  "_type" : "_doc",
  "_id" : "dmx9emwB7Q7sfK_2g0Zo",
  "_score" : 1.0,
  "_source" : {
    "record_id" : "72552",
    "duration" : "0",
    "src_bytes" : "297",
    "host" : "paris",
    "message" : "72552,0,297,9317",
    "@version" : "1",
    "@timestamp" : "2019-08-10T07:45:41.642Z",
    "dest_bytes" : "9317",
    "path" : "/home/ubuntu/Documents/esearch/conn250K.csv"
  }
}
```

You can run this query to follow when the data load is complete, which is when the document count is 256,670.

```
curl XGET http://parisx:9200/_cat/indices?v
```

## Create Index Pattern in Kibana

Open Kibana.

Management / Create index pattern

**Elasticsearch**

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

**Kibana**

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

**Beats**

- Central Management

**Create index pattern**

- ★ filebeat-\*
- logstash\*
- nginx
- ny\*
- nytaxi
- taxi\*

## Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices and visualizations.

### Step 1 of 2: Define index pattern

**Index pattern**

network|

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

✓ **Success!** Your index pattern matches 1 index.

**network**

Rows per page: 10 ▾

Create the

**Index Pattern**. Don't use **@timestamp** as a key field as that only refers to the time we loaded the data into Logstash. Unfortunately, the data provided by Kaggle does not include any date, which is strange for network data. But we can use the **record\_id** in later time series analysis.

## network

This page lists every field in the **network** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#)

Fields (22) | Scripted fields (0) | Source filters (0)

Q Filter | All field types ▾

| Name             | Type    | Format | Searchable | Aggregata... | Excluded |
|------------------|---------|--------|------------|--------------|----------|
| @timestamp       | date    |        | ●          | ●            | ✎        |
| @version         | string  |        | ●          |              | ✎        |
| @version.keyword | string  |        | ●          | ●            | ✎        |
| _id              | string  |        | ●          | ●            | ✎        |
| _index           | string  |        | ●          | ●            | ✎        |
| _score           | number  |        |            |              | ✎        |
| _source          | _source |        |            |              | ✎        |
| _type            | string  |        | ●          | ●            | ✎        |

Now go to the

**Discover** tab and list some documents:

**\_source**

```
record_id: 256669 duration: 0 src_bytes: 198 host: paris message: 256669,0,198,2169 @version: 1 @timestamp: Aug 18, 2019 @ 18:
dest_bytes: 2169 path: /home/ubuntu/Documents/esearch/conn258K.csv _id: q299emwB7Q7sfK_22hXN _type: _doc _index: network _score
```

Expanded document [View single](#)

Table JSON

|              |   |
|--------------|---|
| @timestamp   | Aug 18, 2019 @ 10:46:03.971                 |
| t @version   | 1   |
| t _id        | q299emwB7Q7sfK_22hXN                        |
| t _index     | network                                     |
| # _score     | -   |
| t _type      | _doc  |
| t dest_bytes | 2169  |
| t duration   | 0   |
| t host       | paris                                       |
| t message    | 256669,0,198,2169                           |
| t path       | /home/ubuntu/Documents/esearch/conn258K.csv |
| t record_id  | 256669                                      |
| t src_bytes  | 198   |

In the next blog post we will show how to use Elasticsearch Machine Learning to do Anomaly Detection on this network traffic.