Instalación Windows

Primero validamos la versión de Java, debemos de tener instaldo la versión 1.8.0\_231:

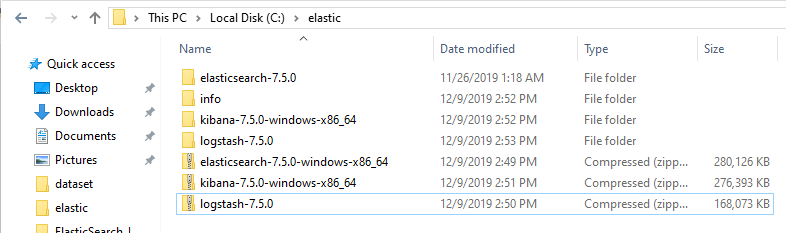
|  |
| --- |
| C:\Users\carlos.hernandez>java -version  java version "1.8.0\_231"  Java(TM) SE Runtime Environment (build 1.8.0\_231-b11)  Java HotSpot(TM) Client VM (build 25.231-b11, mixed mode, sharing)  C:\Users\carlos.hernandez> |

Descargar la última versión de los paquetes

<https://www.elastic.co/es/products/>

* **Elasticsearch**: <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.5.0-windows-x86_64.zip>
* **Kibana**: <https://artifacts.elastic.co/downloads/kibana/kibana-7.5.0-windows-x86_64.zip>
* **Logstash**: <https://artifacts.elastic.co/downloads/logstash/logstash-7.5.0.zip>

Guardar en *c:/elastic* y descomprimir los archivos .zip, al final se debe de ver asi:



# INSTALAR ELASTICSEARCH

## Iniciar Servicio

Vamos a iniciar ElasticSearch, para eso entramos a c:/elastic/elasticsearch-7.5.0/bin, e inicializamos con el .bat

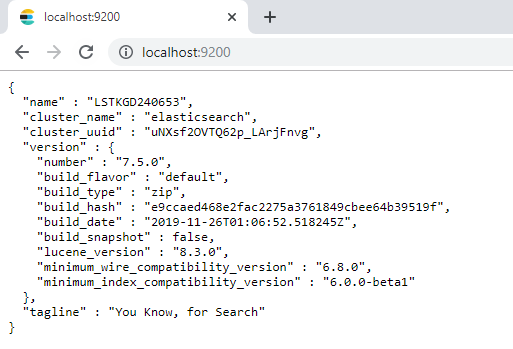
1. cd c:/elastic/elasticsearch-7.5.0/bin/
2. elasticsearch.bat
3. accesar navegador a la direccion http://localhost:9200/

|  |
| --- |
| C:\>cd elastic/elasticsearch-7.5.0/bin  C:\elastic\elasticsearch-7.5.0\bin>elasticsearch.bat  OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.  [2019-12-09T14:55:54,614][INFO ][o.e.e.NodeEnvironment ] [LSTKGD240653] using [1] data paths, mounts [[(C:)]], net usable\_space [408.5gb], net total\_space [464.3gb], types [NTFS]  [2019-12-09T14:55:54,625][INFO ][o.e.e.NodeEnvironment ] [LSTKGD240653] heap size [990.7mb], compressed ordinary object pointers [true]  [2019-12-09T14:55:54,634][INFO ][o.e.n.Node ] [LSTKGD240653] node name [LSTKGD240653], node ID [eiMWgFN\_SC2wIkE48Ea\_uQ], cluster name [elasticsearch]  [2019-12-09T14:55:54,637][INFO ][o.e.n.Node ] [LSTKGD240653] version[7.5.0], pid[8320], build[default/zip/e9ccaed468e2fac2275a3761849cbee64b39519f/2019-11-26T01:06:52.518245Z], OS[Windows 10/10.0/amd64], JVM[AdoptOpenJDK/OpenJDK 64-Bit Server VM/13.0.1/13.0.1+9]  [2019-12-09T14:55:54,638][INFO ][o.e.n.Node ] [LSTKGD240653] JVM home [C:\elastic\elasticsearch-7.5.0\jdk]  [2019-12-09T14:55:54,642][INFO ][o.e.n.Node ] [LSTKGD240653] JVM arguments [-Des.networkaddress.cache.ttl=60, -Des.networkaddress.cache.negative.ttl=10, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dio.netty.allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.locale.providers=COMPAT, -Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.io.tmpdir=C:\Users\CARLOS~1.HER\AppData\Local\Temp\elasticsearch, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs\_err\_pid%p.log, -Xlog:gc\*,gc+age=trace,safepoint:file=logs/gc.log:utctime,pid,tags:filecount=32,filesize=64m, -XX:MaxDirectMemorySize=536870912, -Delasticsearch, -Des.path.home=C:\elastic\elasticsearch-7.5.0, -Des.path.conf=C:\elastic\elasticsearch-7.5.0\config, -Des.distribution.flavor=default, -Des.distribution.type=zip, -Des.bundled\_jdk=true]  .  .  [2019-12-09T14:56:06,967][INFO ][o.e.p.PluginsService ] [LSTKGD240653] no plugins loaded  [2019-12-09T14:56:14,631][INFO ][o.e.x.s.a.s.FileRolesStore] [LSTKGD240653] parsed [0] roles from file [C:\elastic\elasticsearch-7.5.0\config\roles.yml]  [2019-12-09T14:56:15,718][INFO ][o.e.x.m.p.l.CppLogMessageHandler] [LSTKGD240653] [controller/1040] [Main.cc@110] controller (64 bit): Version 7.5.0 (Build 17d1c724ca38a1) Copyright (c) 2019 Elasticsearch BV  [2019-12-09T14:56:16,685][DEBUG][o.e.a.ActionModule ] [LSTKGD240653] Using REST wrapper from plugin org.elasticsearch.xpack.security.Security  [2019-12-09T14:56:16,877][INFO ][o.e.d.DiscoveryModule ] [LSTKGD240653] using discovery type [zen] and seed hosts providers [settings]  [2019-12-09T14:56:18,208][INFO ][o.e.n.Node ] [LSTKGD240653] initialized  [2019-12-09T14:56:55,578][INFO ][o.e.n.Node ] [LSTKGD240653] starting ...  [2019-12-09T14:56:59,938][INFO ][o.e.t.TransportService ] [LSTKGD240653] publish\_address {127.0.0.1:9300}, bound\_addresses {127.0.0.1:9300}, {[::1]:9300}  [2019-12-09T14:57:00,702][WARN ][o.e.b.BootstrapChecks ] [LSTKGD240653] the default discovery settings are unsuitable for production use; at least one of [discovery.seed\_hosts, discovery.seed\_providers, cluster.initial\_master\_nodes] must be configured  [2019-12-09T14:57:00,718][INFO ][o.e.c.c.ClusterBootstrapService] [LSTKGD240653] no discovery configuration found, will perform best-effort cluster bootstrapping after [3s] unless existing master is discovered  [2019-12-09T14:57:03,727][INFO ][o.e.c.c.Coordinator ] [LSTKGD240653] setting initial configuration to VotingConfiguration{eiMWgFN\_SC2wIkE48Ea\_uQ}  [2019-12-09T14:57:05,568][INFO ][o.e.c.s.MasterService ] [LSTKGD240653] elected-as-master ([1] nodes joined)[{LSTKGD240653}{eiMWgFN\_SC2wIkE48Ea\_uQ}{b76DWmszSkuVq3Ja323HtA}{127.0.0.1}{127.0.0.1:9300}{dilm}{ml.machine\_memory=17015447552, xpack.installed=true, ml.max\_open\_jobs=20} elect leader, \_BECOME\_MASTER\_TASK\_, \_FINISH\_ELECTION\_], term: 1, version: 1, delta: master node changed {previous [], current [{LSTKGD240653}{eiMWgFN\_SC2wIkE48Ea\_uQ}{b76DWmszSkuVq3Ja323HtA}{127.0.0.1}{127.0.0.1:9300}{dilm}{ml.machine\_memory=17015447552, xpack.installed=true, ml.max\_open\_jobs=20}]}  [2019-12-09T14:57:05,901][INFO ][o.e.c.c.CoordinationState] [LSTKGD240653] cluster UUID set to [uNXsf2OVTQ62p\_LArjFnvg]  [2019-12-09T14:57:06,208][INFO ][o.e.c.s.ClusterApplierService] [LSTKGD240653] master node changed {previous [], current [{LSTKGD240653}{eiMWgFN\_SC2wIkE48Ea\_uQ}{b76DWmszSkuVq3Ja323HtA}{127.0.0.1}{127.0.0.1:9300}{dilm}{ml.machine\_memory=17015447552, xpack.installed=true, ml.max\_open\_jobs=20}]}, term: 1, version: 1, reason: Publication{term=1, version=1}  [2019-12-09T14:57:06,703][INFO ][o.e.g.GatewayService ] [LSTKGD240653] recovered [0] indices into cluster\_state  [2019-12-09T14:57:06,708][INFO ][o.e.h.AbstractHttpServerTransport] [LSTKGD240653] publish\_address {127.0.0.1:9200}, bound\_addresses {127.0.0.1:9200}, {[::1]:9200}  [2019-12-09T14:57:09,237][INFO ][o.e.l.LicenseService ] [LSTKGD240653] license [fed88169-332e-46a2-b088-09101baea4cd] mode [basic] - valid  [2019-12-09T14:57:09,239][INFO ][o.e.x.s.s.SecurityStatusChangeListener] [LSTKGD240653] Active license is now [BASIC]; Security is disabled |

## Validar Servicio

Para validar que este encendido puedes pone resto en tu navegador:

http://localhost:9200/



Con esto tenemos instalado ElasticSearch, podemos también empezar a utilizar parte de los comandos de la API

* Conocer el estado del cluster ***http://localhost:9200/\_cluster/health?pretty***
* Conocer el estado de los nodos del cluster: ***http://localhost:9200/\_cat/nodes?v***
* Conocer el estado de los índices: ***http://localhost:9200/\_cat/indices?v***
* Conocer el estado de los shards: ***http://localhost:9200/\_cat/shards?v***

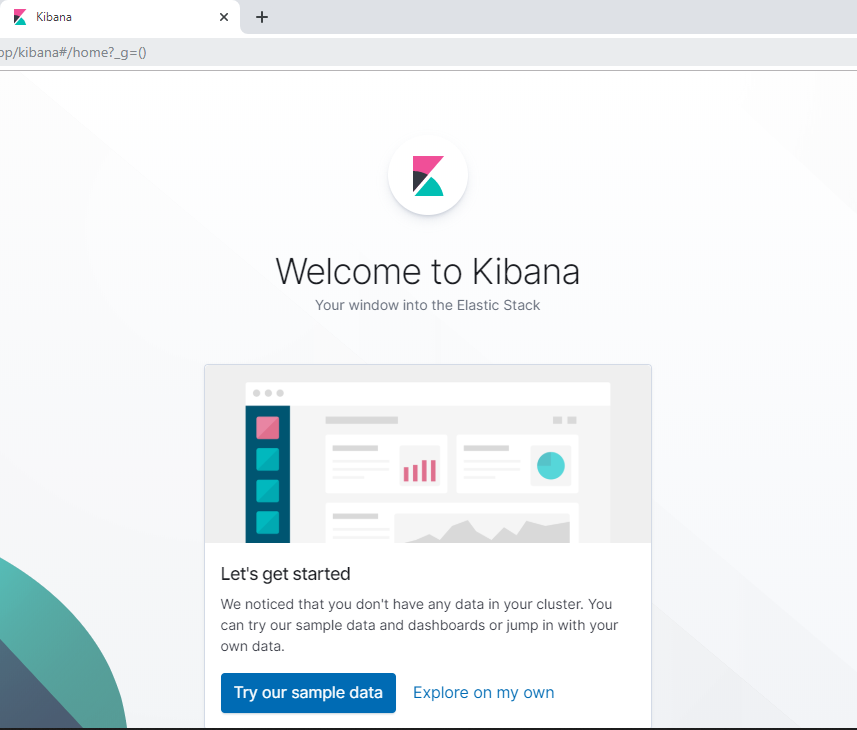
# Instalar KIBANA

1. cd c:/elastic/kibana-7.5.0-windows-x86\_64/bin/
2. kibana.bat

|  |
| --- |
| c:\elastic>cd c:/elastic/kibana-7.5.0-windows-x86\_64/bin/  c:\elastic\kibana-7.5.0-windows-x86\_64\bin>kibana.bat  log [21:30:20.567] [info][plugins-system] Setting up [15] plugins: [security,licensing,code,timelion,features,spaces,translations,uiActions,newsfeed,inspector,embeddable,advancedUiActions,expressions,data,eui\_utils]  log [21:30:20.598] [info][plugins][security] Setting up plugin  log [21:30:20.600] [warning][config][plugins][security] Generating a random key for xpack.security.encryptionKey. To prevent sessions from being invalidated on restart, please set xpack.security.encryptionKey in kibana.yml  log [21:30:20.601] [warning][config][plugins][security] Session cookies will be transmitted over insecure connections. This is not recommended.  log [21:30:20.656] [info][licensing][plugins] Setting up plugin  log [21:30:20.661] [info][code][plugins] Setting up plugin  log [21:30:20.663] [info][plugins][timelion] Setting up plugin  log [21:30:20.664] [info][features][plugins] Setting up plugin  log [21:30:20.666] [info][plugins][spaces] Setting up plugin  log [21:30:20.670] [info][plugins][translations] Setting up plugin  log [21:30:20.671] [info][data][plugins] Setting up plugin  log [21:31:11.456] [warning][licensing][plugins] License information could not be obtained from Elasticsearch for the [data] cluster. Error: Request Timeout after 30000ms  log [21:31:14.123] [warning][legacy-plugins] Skipping non-plugin directory at c:\elastic\kibana-7.5.0-windows-x86\_64\src\legacy\core\_plugins\visualizations  log [21:31:16.214] [info][plugins-system] Starting [8] plugins: [security,licensing,code,timelion,features,spaces,translations,data]  log [21:31:16.245] [info][migrations] Creating index .kibana\_task\_manager\_1.  log [21:31:16.250] [info][licensing][plugins] Imported changed license information from Elasticsearch for the [data] cluster: type: basic | status: active  log [21:31:16.253] [info][migrations] Creating index .kibana\_1.  log [21:31:17.480] [info][migrations] Pointing alias .kibana\_task\_manager to .kibana\_task\_manager\_1.  log [21:31:17.561] [info][migrations] Pointing alias .kibana to .kibana\_1.  log [21:31:17.663] [info][migrations] Finished in 1418ms.  log [21:31:17.727] [info][migrations] Finished in 1485ms.  log [21:31:35.956] [info][status][plugin:maps@7.5.0] Status changed from yellow to green - Ready  log [21:31:36.374] [info][status][plugin:spaces@7.5.0] Status changed from yellow to green - Ready  log [21:31:41.159] [warning][reporting] Generating a random key for xpack.reporting.encryptionKey. To prevent pending reports from failing on restart, please set xpack.reporting.encryptionKey in kibana.yml  log [21:31:41.176] [info][status][plugin:reporting@7.5.0] Status changed from uninitialized to green - Ready  log [21:31:41.271] [info][listening] Server running at http://localhost:5601  log [21:31:41.366] [info][server][Kibana][http] http server running at http://localhost:5601 |

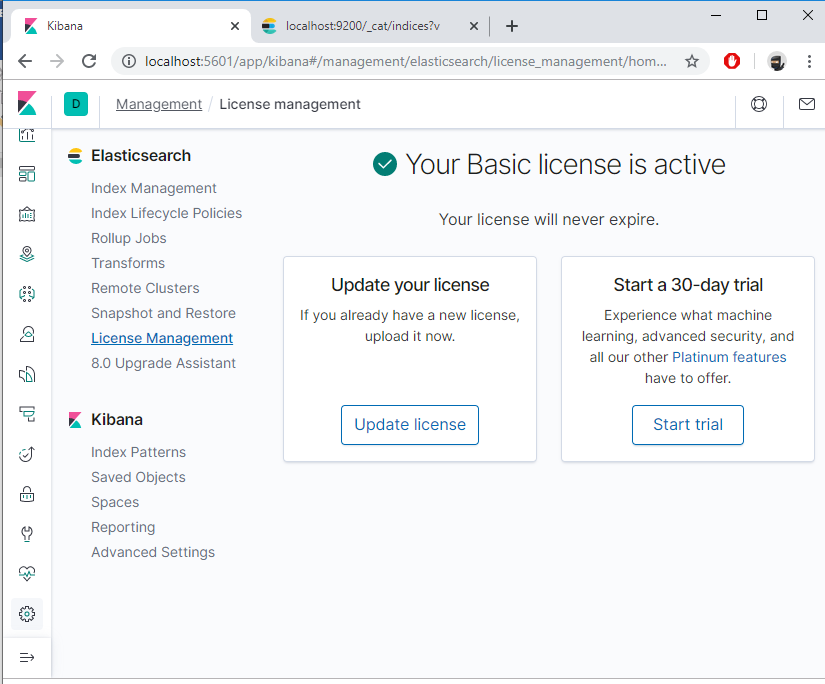
## Validar servicio KIBANA

Entramos al nevegador a la siguiente direccion ***https://localhost:5601/***



## Habilitar Licencia Temporal

Esta licencia te permitirá explorar las funcionalidades avanzadas del stack de elastic, para habilitarlo del lado izquierdo en el menú se debe de ver un icono de forma de engrane, llamado Managment: 



# Ingestar información utilizando LOGSTASH

Primero conozcamos nuestros datos, en github existe un repositorio donde pueden descargar diferentes datasets, para este ejemplo utilizaremos el llamado NYC TRAFFIC ACCIDENTS.

<https://github.com/elastic/examples/tree/master/Exploring%20Public%20Datasets/nyc_traffic_accidents>

Sin embargo, estos ejemplos son para versión 6, el problema es que no es compatible con versión 7, se tienen que hacer un par de ajustes, para corregir los errores vamos a ultizar los siguientes archivos:

* csv\_example.conf
* csv\_example2.conf
* export\_objects\_nyc.ndjson
* nyc\_crashes\_template.json
* ejemplo.csv
* NYPDCrashes2.csv

Para ejecutar logstash, primero debemos de definir el archivo de configuración:

## csv\_example.conf

|  |
| --- |
| input {  file {  path => ["C:/elastic/dataset/NYPDCrashes2.csv"]  start\_position => "beginning"  }  }  filter {  csv {  columns => [ "date","time","borough","zip\_code","latitude","longitude","location","on\_street\_name","cross\_street\_name","off\_street\_name","number\_of\_persons\_injured","number\_of\_persons\_killed","number\_of\_pedestrians\_injured","number\_of\_pedestrians\_killed","number\_of\_cyclist\_injured","number\_of\_cyclist\_killed","number\_of\_motorist\_injured","number\_of\_motorist\_killed","contributing\_factor\_vehicle\_1","contributing\_factor\_vehicle\_2","contributing\_factor\_vehicle\_3","contributing\_factor\_vehicle\_4","contributing\_factor\_vehicle\_5","collision\_id","vehicle\_type\_code\_1","vehicle\_type\_code\_2","vehicle\_type\_code\_3","vehicle\_type\_code\_4","vehicle\_type\_code\_5"  ]  separator => ","  }  }  output {  stdout { codec => rubydebug }  } |

## Arrancar logstash

Despues de descomprimir el archivo .zip, debemos de entrar desde cmd, y se arranca con el executable logstash.bat:

Pero antes debemos de entender los directorios que se encuentran dentro:

* **config**: donde se encuentran los archivos de configuración, como **jvm.option** donde se configuran las variables para arrancar la maquina virtual de java, **logstash.yml** donde se pueden agregar configuraciones avanzadas de logstash, **pipelines.yml** donde se pueden configurar todos los pipelines (muy útil cuando corremos como servicio)
* **data**: este directorio es importante cuando lees archivos desde logstash, aquí se almacena el apuntador de lectura, asi como el buffer en caso de que logstash detecte que ES no esta disponible, cuando estamos probando lo ideal es borrar todo el contenido dentro de este directorio
* **bin**: donde se encuentran los executables.

Para ejecutarlo aplicamos los siguientes comandos:

* Borrar el contenido de c:\elastic\logstash-7.5.0\data:
  + cd c:\elastic\logstash-7.5.0\data
  + >rd dead\_letter\_queue /s
  + >rd queue /s
  + >rd plugins /s
  + >del \*
* Guardar los archivos de configuración en un directorio en: c:\elastic\conf
* cd c:\elastic\logstash-7.5.0\bin
* logstash.bat -f c:\elastic\conf\csv\_ejemplo.conf

|  |
| --- |
| c:\elastic\logstash-7.5.0\bin>logstash.bat -f c:\elastic\conf\csv\_ejemplo.conf  Thread.exclusive is deprecated, use Thread::Mutex  Sending Logstash logs to c:/elastic/logstash-7.5.0/logs which is now configured via log4j2.properties  [2019-12-12T12:27:01,699][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.queue", :path=>"c:/elastic/logstash-7.5.0/data/queue"}  [2019-12-12T12:27:01,816][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.dead\_letter\_queue", :path=>"c:/elastic/logstash-7.5.0/data/dead\_letter\_queue"}  [2019-12-12T12:27:01,905][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified  [2019-12-12T12:27:01,919][INFO ][logstash.runner ] Starting Logstash {"logstash.version"=>"7.5.0"}  [2019-12-12T12:27:01,950][INFO ][logstash.agent ] No persistent UUID file found. Generating new UUID {:uuid=>"bf7060fb-a0b9-4f74-829a-4acacd75d2e8", :path=>"c:/elastic/logstash-7.5.0/data/uuid"}  [2019-12-12T12:27:03,766][INFO ][org.reflections.Reflections] Reflections took 46 ms to scan 1 urls, producing 20 keys and 40 values  [2019-12-12T12:27:04,964][WARN ][org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.RubyArray) has been create for key: cluster\_uuids. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team.  [2019-12-12T12:27:04,978][INFO ][logstash.javapipeline ][main] Starting pipeline {:pipeline\_id=>"main", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max\_inflight"=>500, "pipeline.sources"=>["c:/elastic/conf/csv\_ejemplo.conf"], :thread=>"#<Thread:0x3fa3f run>"}  [2019-12-12T12:27:05,772][INFO ][logstash.inputs.file ][main] No sincedb\_path set, generating one based on the "path" setting {:sincedb\_path=>"c:/elastic/logstash-7.5.0/data/plugins/inputs/file/.sincedb\_81977c4d3cee455bd0587fb716ec00f5", :path=>["C:/elastic/dataset/ejemplo.csv"]}  [2019-12-12T12:27:05,815][INFO ][logstash.javapipeline ][main] Pipeline started {"pipeline.id"=>"main"}  [2019-12-12T12:27:05,921][INFO ][logstash.agent ] Pipelines running {:count=>1, :running\_pipelines=>[:main], :non\_running\_pipelines=>[]}  [2019-12-12T12:27:05,927][INFO ][filewatch.observingtail ][main] START, creating Discoverer, Watch with file and sincedb collections  [2019-12-12T12:27:06,862][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}  C:/elastic/logstash-7.5.0/vendor/bundle/jruby/2.5.0/gems/awesome\_print-1.7.0/lib/awesome\_print/formatters/base\_formatter.rb:31: warning: constant ::Fixnum is deprecated  {  "number\_of\_cyclist\_injured" => "0",  "message" => "09/13/2019,21:15,BROOKLYN,11215,40.67116,-73.97142,POINT (-73.97142 40.67116),,,18 PROSPECT PARK WEST ,0,0,0,0,0,0,0,0,Unspecified,,,,,4206285,Sedan,,,,\r",  "number\_of\_persons\_killed" => "0",  "time" => "21:15",  "date" => "09/13/2019",  "location" => "POINT (-73.97142 40.67116)",  "zip\_code" => "11215",  "vehicle\_type\_code\_2" => nil,  "number\_of\_motorist\_killed" => "0",  "borough" => "BROOKLYN",  "contributing\_factor\_vehicle\_3" => nil,  "latitude" => "40.67116",  "off\_street\_name" => "18 PROSPECT PARK WEST ",  "number\_of\_persons\_injured" => "0",  "longitude" => "-73.97142",  "vehicle\_type\_code\_3" => nil,  "host" => "LSTKGD240653",  "contributing\_factor\_vehicle\_5" => nil,  "number\_of\_pedestrians\_injured" => "0",  "collision\_id" => "4206285",  "number\_of\_motorist\_injured" => "0",  "contributing\_factor\_vehicle\_1" => "Unspecified",  "contributing\_factor\_vehicle\_2" => nil,  "vehicle\_type\_code\_1" => "Sedan",  "number\_of\_cyclist\_killed" => "0",  "cross\_street\_name" => nil,  "contributing\_factor\_vehicle\_4" => nil,  "number\_of\_pedestrians\_killed" => "0",  "@timestamp" => 2019-12-12T18:27:07.074Z,  "path" => "C:/elastic/dataset/ejemplo.csv",  "@version" => "1",  "vehicle\_type\_code\_4" => nil,  "on\_street\_name" => nil,  "vehicle\_type\_code\_5" => nil  } |

En este primer ejemplo podemos ver como logstash proceso los mensajes que leyo desde el archivo csv, aquí tenemos varios detalles, sobre todo por el campo llamado location, como pueden ver se especifica que es un punto geolocalizable sin embargo para que elasticsearch lo tome como tal debe de estar en un formato “latitud,longitud”

# Ejemplo NYC - Ingestar Datos a ES

Utilizaremos el archivo **ejemplo\_csv2.conf**, donde ya después de analizarlo se pusieron muchas condiciones que establecen de dodne se va a tomar los valores de latitud y longitud.

## csv\_example2.conf

|  |
| --- |
| input {      file {          #path => ["C:/elastic/dataset/ejemplo.csv"]  path => ["C:/elastic/dataset/NYPDCrashes2.csv"]          start\_position => "beginning"      }  }  filter {  csv {          columns => ["date","time","borough","zip\_code","latitude","longitude","location\_2","on\_street\_name","cross\_street\_name","off\_street\_name","number\_of\_persons\_injured","number\_of\_persons\_killed","number\_of\_pedestrians\_injured","number\_of\_pedestrians\_killed","number\_of\_cyclist\_injured","number\_of\_cyclist\_killed","number\_of\_motorist\_injured","number\_of\_motorist\_killed","contributing\_factor\_vehicle\_1","contributing\_factor\_vehicle\_2","contributing\_factor\_vehicle\_3","contributing\_factor\_vehicle\_4","contributing\_factor\_vehicle\_5","unique\_key","vehicle\_type\_code\_1","vehicle\_type\_code\_2","vehicle\_type\_code\_3","vehicle\_type\_code\_4","vehicle\_type\_code\_5"]      }  # Quitar la primer linea del archivo  if ([date] == "DATE") {  drop { }  }  if [on\_street\_name] and [cross\_street\_name] {       ruby {          #crear un nuevo campo llamado intersecion que es la combinacion de los campos ross street & on street names  code => "event.set('intersection',[event.get('on\_street\_name'), event.get('cross\_street\_name')].sort.join('--'))"       }  }  #si los campos latitude y longitude existen, crear el campo location con estos valores separados por una ,  if [latitude] and [longitude] {          mutate {              add\_field => {               "location" => "%{latitude},%{longitude}"              }          }  }  #en caso de no contar con ellos pero que el campo location\_2 contenga la palabra POINT tomar esos valores, creando 2 nuevos campos llamados lat y lon  if [location\_2] =~ /POINT/ and [location] == "" {  grok {  match => { "location\_2" => "POINT \(%{DATA:long} %{DATA:lat}\)" }  add\_field => { "location" => "%{lat},%{long}" }  }  }  # combinar los campos date y time dentro de un nuevo campo llamado datetime      mutate {          add\_field => {              "datetime" => "%{date} %{time}"              "contributing\_factor\_vehicle" => "%{contributing\_factor\_vehicle\_1}"              "vehicle\_type" => "%{vehicle\_type\_code\_1}"            }    # convertir a tipo integer          convert => ["number\_of\_persons\_injured","integer","number\_of\_persons\_killed","integer","number\_of\_pedestrians\_injured","integer","number\_of\_pedestrians\_killed","integer","number\_of\_cyclist\_injured","integer","number\_of\_cyclist\_killed","integer","number\_of\_motorist\_injured","integer","number\_of\_motorist\_killed","integer"]          strip => ["on\_street\_name", "cross\_street\_name"]      }  #si el campo number\_of\_person\_killed esta vacio agregarle un 0  if ![number\_of\_persons\_killed]  {    mutate {        add\_field => {"number\_of\_persons\_killed" => "0"}    }  }  #si el campo number\_of\_person\_injured esta vacio agregarle un 0  if ![number\_of\_persons\_injured]  {    mutate {        add\_field => {"number\_of\_persons\_injured" => "0"}    }  }  #creamos un campo llamado number\_persons\_impacted  ruby {      # Obtener el numero total de personas lesionadas      code => "event.set('number\_persons\_impacted',event.get('number\_of\_persons\_killed') + event.get('number\_of\_persons\_injured'))"  }  # combinar los valores del arreglo contributing\_factor\_vehicle\_X (X=1,2,3,4,5) en el campo contributing\_factor\_vehicle      if [contributing\_factor\_vehicle\_2] and "Unspecified" != [contributing\_factor\_vehicle\_2] and [contributing\_factor\_vehicle\_2] not in [contributing\_factor\_vehicle] {          mutate {              merge => ["contributing\_factor\_vehicle", "contributing\_factor\_vehicle\_2"]          }      }      if [contributing\_factor\_vehicle\_3] and "Unspecified" != [contributing\_factor\_vehicle\_3] and [contributing\_factor\_vehicle\_3] not in [contributing\_factor\_vehicle] {          mutate {              merge => ["contributing\_factor\_vehicle", "contributing\_factor\_vehicle\_3"]          }      }      if [contributing\_factor\_vehicle\_4] and "Unspecified" != [contributing\_factor\_vehicle\_4] and [contributing\_factor\_vehicle\_4] not in [contributing\_factor\_vehicle] {          mutate {              merge => ["contributing\_factor\_vehicle", "contributing\_factor\_vehicle\_4"]          }      }      if [contributing\_factor\_vehicle\_5] and "Unspecified" != [contributing\_factor\_vehicle\_5] and [contributing\_factor\_vehicle\_5] not in [contributing\_factor\_vehicle] {          mutate {              merge => ["contributing\_factor\_vehicle", "contributing\_factor\_vehicle\_5"]          }      }  # combinar los valores del arreglo vehicle\_type\_code\_X (X=1,2,3,4,5) en vehicle\_type      if [vehicle\_type\_code\_2] and "Unspecified" != [vehicle\_type\_code\_2] and [vehicle\_type\_code\_2] not in [vehicle\_type] {          mutate {              merge => ["vehicle\_type", "vehicle\_type\_code\_2"]          }      }      if [vehicle\_type\_code\_3] and "Unspecified" != [vehicle\_type\_code\_3] and [vehicle\_type\_code\_3] not in [vehicle\_type] {          mutate {              merge => ["vehicle\_type", "vehicle\_type\_code\_3"]          }      }      if [vehicle\_type\_code\_4] and "Unspecified" != [vehicle\_type\_code\_4] and [vehicle\_type\_code\_4] not in [vehicle\_type] {          mutate {              merge => ["vehicle\_type", "vehicle\_type\_code\_4"]          }      }      if [vehicle\_type\_code\_5] and "Unspecified" != [vehicle\_type\_code\_5] and [vehicle\_type\_code\_5] not in [vehicle\_type] {          mutate {              merge => ["vehicle\_type", "vehicle\_type\_code\_5"]          }      }  ################################################      date {          match => [ "datetime", "MM/dd/YY HH:mm", "MM/dd/YY H:mm"]          timezone => "EST"      }  # crear un nuevo campo hour\_of\_day tomado de la hora del campo time  grok {  match => { "time" => "%{DATA:hour\_of\_day}:%{GREEDYDATA}" }  }  mutate {  convert => ["hour\_of\_day", "integer"]  }  # quitar algunos campos extras      mutate {          remove\_field => ["datetime", "contributing\_factor\_vehicle\_1", "contributing\_factor\_vehicle\_2", "contributing\_factor\_vehicle\_3", "contributing\_factor\_vehicle\_4", "contributing\_factor\_vehicle\_5","vehicle\_type\_code\_1", "vehicle\_type\_code\_2", "vehicle\_type\_code\_3", "vehicle\_type\_code\_4", "vehicle\_type\_code\_5"]      }  }  output {  #si deseamos enviarlo a elasticsearch podemos utilizarlo de la siguiente manera  elasticsearch {      hosts => "localhost:9200"      index => "nyc\_visionzero"  }  #las formas de salida, el codec ruby debug es nuestro mejor amigo  #stdout { codec => rubydebug }  stdout { codec => dots }  } |

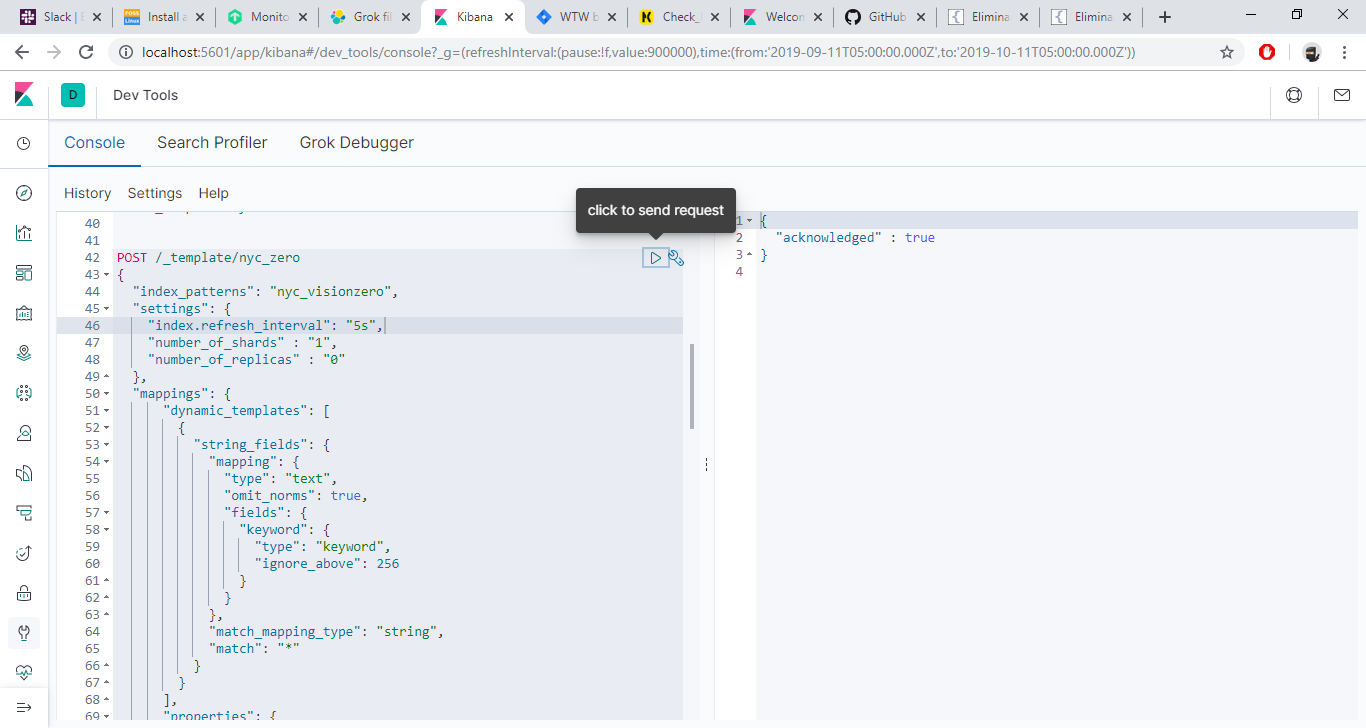
## Definir Template

Antes de Ingestar toda la info del archivo **NYPDCrashes2.csv**, primero debemos definir un template para que los campos geo localizados sean interpretados como tal, para eso utilizaremos la herramienta console de kibana y el archivo **nyc\_crashes\_template.json.**

Accesamos a la herramienta Dev Tools



Una vez Dentro buscamos la consola y pegamos el contenido del archivo nyc\_crashes\_template.json y damos click en el icono send request.



Del lado Derecho veremos un mensaje indicando que la respuesta fue recibida.

## Ingestar utilizando logstash

Procedemos a ingestar los datos del archivo **NYPDCrashes2.csv**, como viene indicado en la configuración de csv\_ejemplo2.conf, como pueden notar ahora se visualiza una serie de puntos, cada punto es un mensaje enviado por logstash a ES

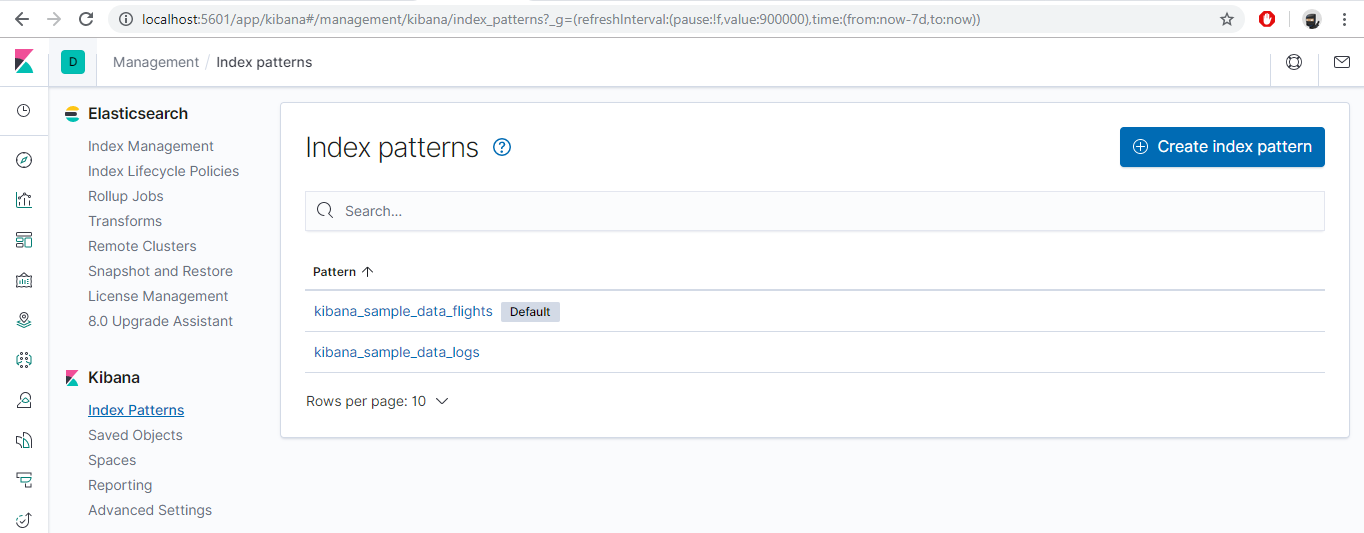
|  |
| --- |
| c:\elastic\logstash-7.5.0\bin>logstash.bat -f c:\elastic\conf\csv\_ejemplo2.conf  Thread.exclusive is deprecated, use Thread::Mutex  Sending Logstash logs to c:/elastic/logstash-7.5.0/logs which is now configured via log4j2.properties  [2019-12-12T13:57:57,842][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified  [2019-12-12T13:57:57,951][INFO ][logstash.runner ] Starting Logstash {"logstash.version"=>"7.5.0"}  [2019-12-12T13:58:00,565][INFO ][org.reflections.Reflections] Reflections took 43 ms to scan 1 urls, producing 20 keys and 40 values  [2019-12-12T13:58:02,398][INFO ][logstash.outputs.elasticsearch][main] Elasticsearch pool URLs updated {:changes=>{:removed=>[], :added=>[http://localhost:9200/]}}  [2019-12-12T13:58:02,606][WARN ][logstash.outputs.elasticsearch][main] Restored connection to ES instance {:url=>"http://localhost:9200/"}  [2019-12-12T13:58:02,663][INFO ][logstash.outputs.elasticsearch][main] ES Output version determined {:es\_version=>7}  [2019-12-12T13:58:02,669][WARN ][logstash.outputs.elasticsearch][main] Detected a 6.x and above cluster: the `type` event field won't be used to determine the document \_type {:es\_version=>7}  [2019-12-12T13:58:02,724][INFO ][logstash.outputs.elasticsearch][main] New Elasticsearch output {:class=>"LogStash::Outputs::ElasticSearch", :hosts=>["//localhost:9200"]}  [2019-12-12T13:58:02,830][INFO ][logstash.outputs.elasticsearch][main] Using default mapping template  [2019-12-12T13:58:02,981][INFO ][logstash.outputs.elasticsearch][main] Attempting to install template {:manage\_template=>{"index\_patterns"=>"logstash-\*", "version"=>60001, "settings"=>{"index.refresh\_interval"=>"5s", "number\_of\_shards"=>1}, "mappings"=>{"dynamic\_templates"=>[{"message\_field"=>{"path\_match"=>"message", "match\_mapping\_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false}}}, {"string\_fields"=>{"match"=>"\*", "match\_mapping\_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword", "ignore\_above"=>256}}}}}], "properties"=>{"@timestamp"=>{"type"=>"date"}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo\_point"}, "latitude"=>{"type"=>"half\_float"}, "longitude"=>{"type"=>"half\_float"}}}}}}}  [2019-12-12T13:58:03,297][WARN ][org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.specialized.RubyArrayOneObject) has been create for key: cluster\_uuids. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team.  [2019-12-12T13:58:03,304][INFO ][logstash.javapipeline ][main] Starting pipeline {:pipeline\_id=>"main", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max\_inflight"=>500, "pipeline.sources"=>["c:/elastic/conf/csv\_ejemplo2.conf"], :thread=>"#<Thread:0xd23070 run>"}  [2019-12-12T13:58:04,205][INFO ][logstash.inputs.file ][main] No sincedb\_path set, generating one based on the "path" setting {:sincedb\_path=>"c:/elastic/logstash-7.5.0/data/plugins/inputs/file/.sincedb\_dad529f77bb25e1dd86afd79d2a0bb0d", :path=>["C:/elastic/dataset/NYPDCrashes2.csv"]}  [2019-12-12T13:58:04,246][INFO ][logstash.javapipeline ][main] Pipeline started {"pipeline.id"=>"main"}  [2019-12-12T13:58:04,323][INFO ][filewatch.observingtail ][main] START, creating Discoverer, Watch with file and sincedb collections  [2019-12-12T13:58:04,327][INFO ][logstash.agent ] Pipelines running {:count=>1, :running\_pipelines=>[:main], :non\_running\_pipelines=>[]}  [2019-12-12T13:58:04,853][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}  .................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................. |

## Visualizar datos utilizando KIBANA

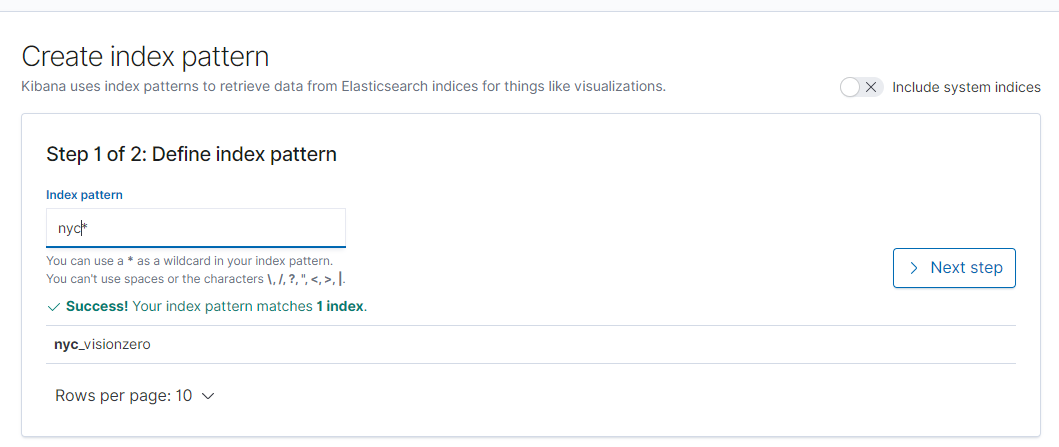
## Crear Index\_pattern

Ahora podemos Visualizar nuestros datos desde kibana creando un index\_pattern:

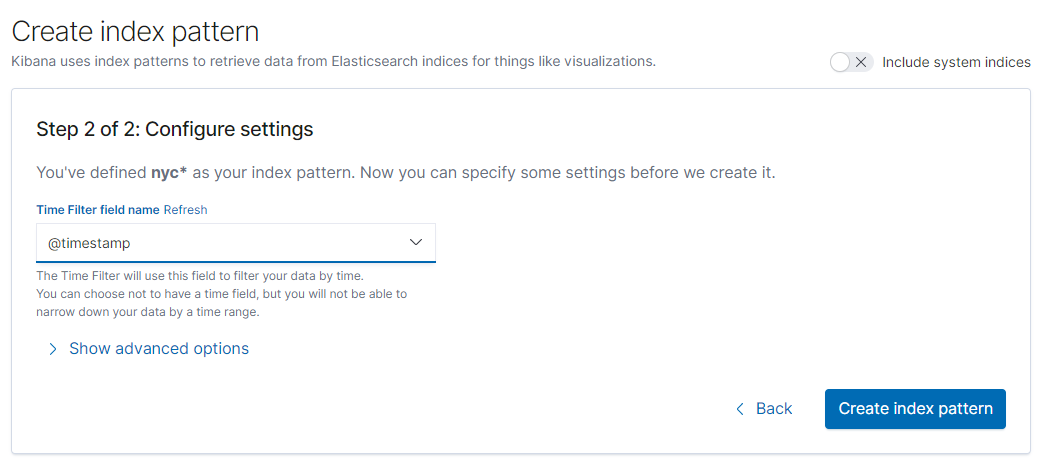
Nuevamente nos vamos a managment en la opción de kibana vemos un link llamado Index\_patterns



Click en **Create index pattern**

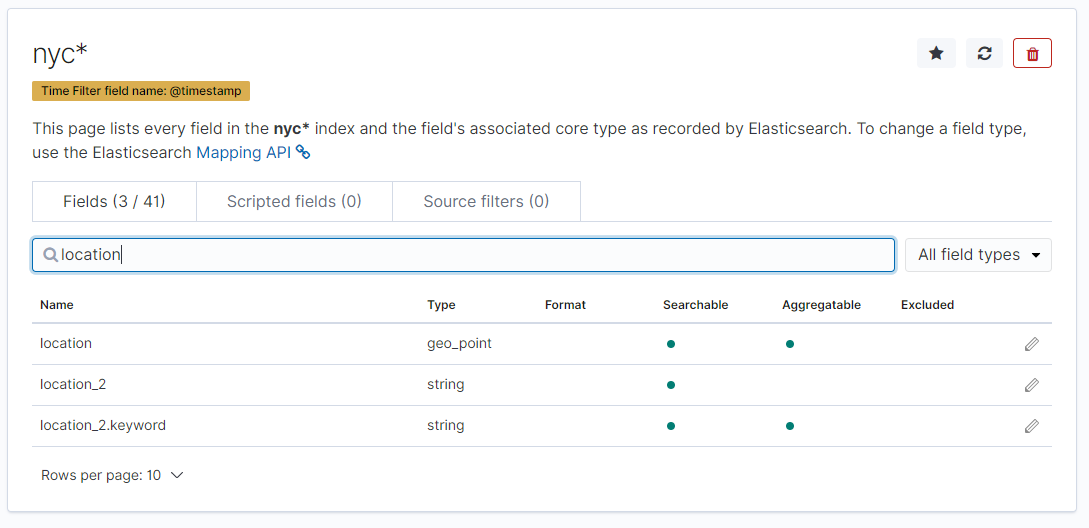


Ponemos el nombre nyc\* y click en **Next step**



Seleccionamos el campo @timestamp y click en **Create index pattern**

A continuación deberemos de ver la siguiente ventana:

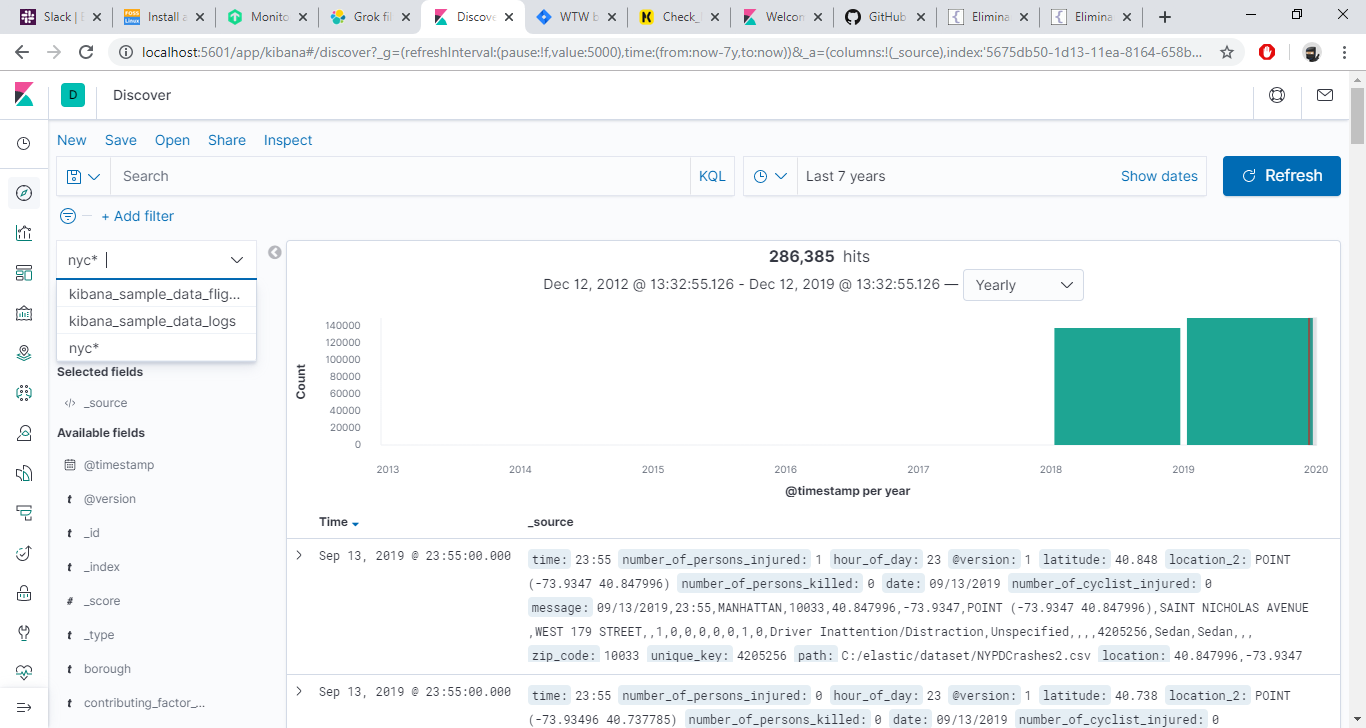


Les aparecerá un resumen de los campos que fueron descubiertos desde el template del index, si buscamos el campo **location**, veremos que el tipo es geo\_point.

Ya finalmente accesamos en la herramieta Discover:



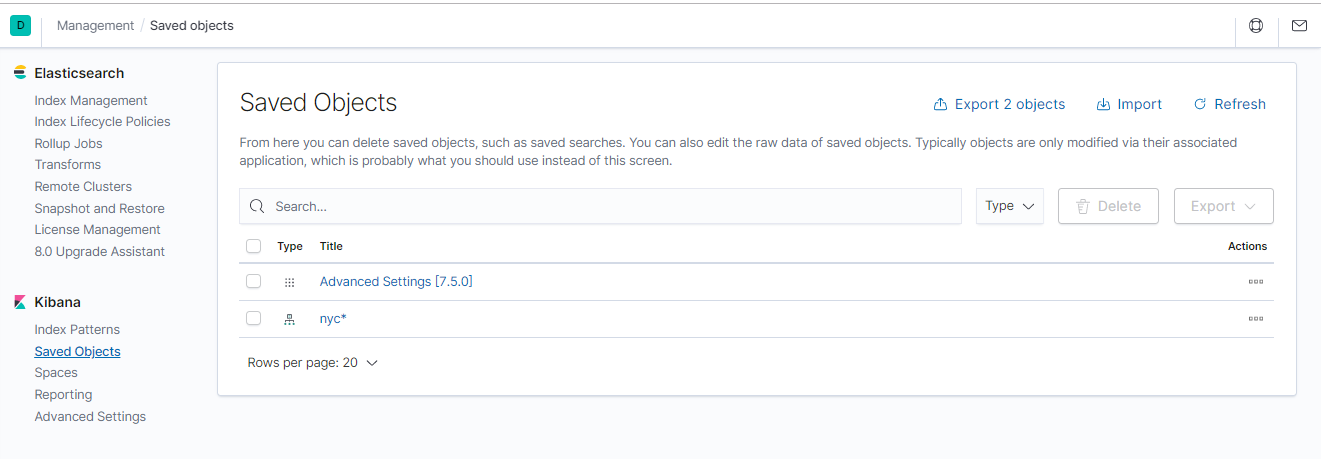
Y nos aseguramos que estamos viendo los valores del index\_pattern que creamos **nyc\***



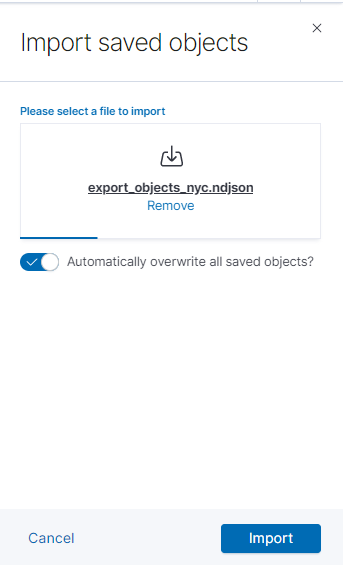
## Importar Objetos de Kibana

Para este demo he agregado un archivo llamado export\_objects\_nyc.ndjson, que incluye un pequeño dashboard donde podamos ver la información que acabaos de ingestar, para importarlo procedemos con lo siguiente:

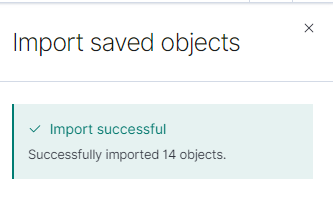
Accesar a managment, en la sección Kibana click en saved\_objects



Damos click en Import



Seleccionamos el archivo **export\_objects\_nyc.ndjson**, y le damos click a import



Veremos la siguiente leyenda, damos click en Done y listo, ahora podremos visualizar nuestro dashboard:

