

ELK从入门到精通

一、ELK是什么？

ElasticStack(ELK)=elasticsearch+Logstash+kibana

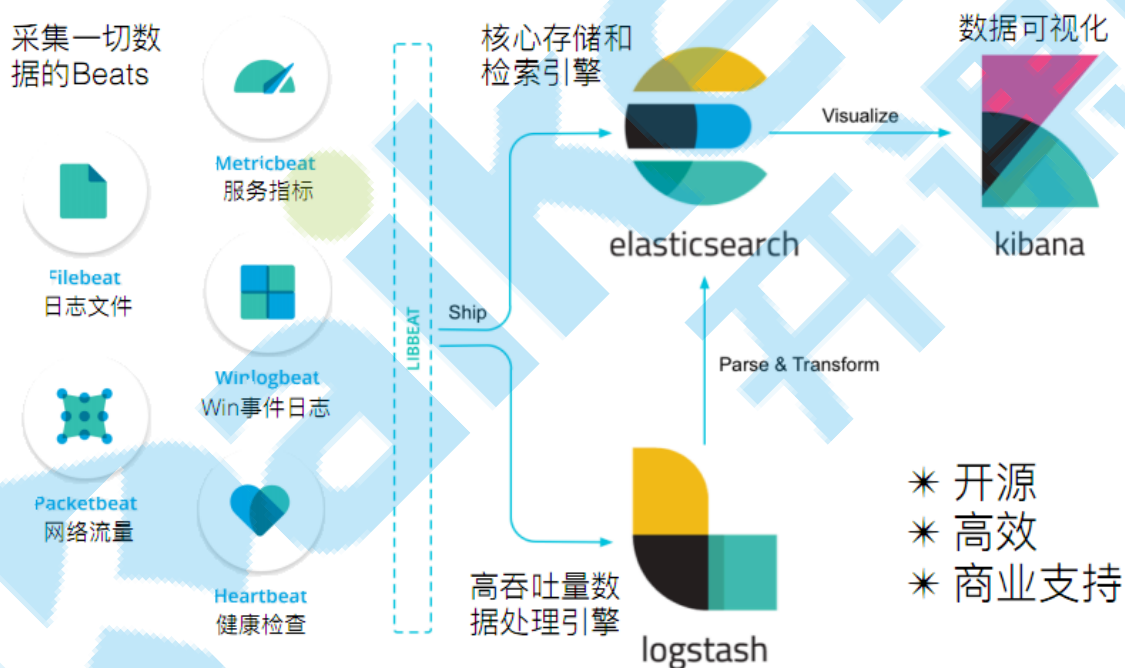
elasticsearch：后台分布式存储以及全文检索

logstash: 日志加工、“搬运工”

kibana：数据可视化展示。

ELK架构为数据分布式存储、可视化查询和日志解析创建了一个功能强大的管理链。三者相互配合，取长补短，共同完成分布式大数据处理工作。

ElasticStack的组成



二、ES安装

1、下载ES

下载：（文件比较大，建议手动下载）

Downloads

Download Elasticsearch

Want it hosted? Deploy on Elastic Cloud. [Get Started »](#)

Version: 7.2.0

Release date: June 26, 2019

License: [Elastic License](#)

Downloads: [WINDOWS](#) sha asc

[MACOS](#) sha asc

[LINUX](#) sha asc

[DEB](#) sha asc

[RPM](#) sha asc

[MSI \(BETA\)](#) sha asc

下载网址:

jdk-8u65-linux-x64.tar.gz	2015/11/13 16:30	GZ 文件	177,013 KB
elasticsearch-6.2.4.tar.gz	2019/6/26 16:11	GZ 文件	28,376 KB

2、安装

解压:

```
1 | tar -zxvf elasticsearch-6.2.4.tar.gz
```

注意: 把elasticsearch软件必须放入/home/es (es是新建用户) 的目录下, 并把elasticsearch设置为es用户所属

创建日志、数据存储目录: (留作备用, 初次先创建)

```
1 | mkdir -p /data/logs/es
2 | mkdir -p /data/es/{data,work,plugins,scripts}
```

创建用户

```
1 | useradd es -s /bin/bash #es不能在root用户下启动, 必须创建新的用户, 用来启动es
```

启动: ./elasticsearch

```
root@jackhu bin]# ./elasticsearch
[2019-06-26T16:14:40.309][WARN ][o.e.b.ElasticsearchUncaughtExceptionHandler] [] uncaught exception in thread [main]
org.elasticsearch.bootstrap.StartupException: java.lang.RuntimeException: can not run elasticsearch as root
    at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:125) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.bootstrap.Elasticsearch.execute(Elasticsearch.java:112) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.cli.EnvironmentAwareCommand.execute(EnvironmentAwareCommand.java:86) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.cli.Command.mainWithoutErrorHandling(Command.java:124) ~[elasticsearch-cli-6.2.4.jar:6.2.4]
    at org.elasticsearch.cli.Command.main(Command.java:90) ~[elasticsearch-cli-6.2.4.jar:6.2.4]
    at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:92) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:85) ~[elasticsearch-6.2.4.jar:6.2.4]
Caused by: java.lang.RuntimeException: can not run elasticsearch as root
    at org.elasticsearch.bootstrap.Bootstrap.initializeNatives(Bootstrap.java:105) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:172) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:323) ~[elasticsearch-6.2.4.jar:6.2.4]
    at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:121) ~[elasticsearch-6.2.4.jar:6.2.4]
```

注意：es不能在root用户下启动，必须创建新的用户，用来启动es

切换用户：su es

再次启动，发现还是报错，原因：当前用户没有执行权限

```
[root@jackhu bin]# su es
[es@jackhu bin]$ ./elasticsearch
Exception in thread "main" java.nio.file.AccessDeniedException: /home/es/elasticsearch-6.2.4/config/jvm.options
    at sun.nio.fs.UnixException.translateToIOException(UnixException.java:84)
    at sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:102)
    at sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:107)
    at sun.nio.fs.UnixFileSystemProvider.newByteChannel(UnixFileSystemProvider.java:214)
    at java.nio.file.Files.newByteChannel(Files.java:361)
    at java.nio.file.Files.newByteChannel(Files.java:407)
    at java.nio.file.spi.FileSystemProvider.newInputStream(FileSystemProvider.java:384)
    at java.nio.file.Files.newInputStream(Files.java:152)
    at org.elasticsearch.tools.launchers.JvmOptionsParser.main(JvmOptionsParser.java:58)
[es@jackhu bin]$ cd ..
```

授权：chown -R es:es elasticsearch-6.2.4

```
[root@jackhu es]# su es
[es@jackhu ~]$ ll
总用量 28376
drwxr-xr-x. 8 es es 143 4月 13 2018 elasticsearch-6.2.4
-rw-r--r--. 1 root root 29056810 6月 26 16:11 elasticsearch-6.2.4.tar.gz
[es@jackhu ~]$ cd elasticsearch-6.2.4
[es@jackhu elasticsearch-6.2.4]$ ll
总用量 224
drwxr-xr-x. 2 es es 4096 6月 26 16:13 bin
```

授权成功，发现elasticsearch已经在es用户下面了，可以启动了，但是启动成功，浏览器不能访问，因此还需要做如下配置：

配置修改：**

```
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 0.0.0.0
#
# Set a custom port for HTTP:
#
```

再次启动：报如下错误

```
[2019-06-26T16:40:34,368][INFO ][o.e.n.Node ] [4fZjnig] starting ...
[2019-06-26T16:40:35,232][INFO ][o.e.t.TransportService ] [4fZjnig] publish_address {192.168.66.66:9300}, b
[2019-06-26T16:40:35,249][INFO ][o.e.b.BootstrapChecks ] [4fZjnig] bound or publishing to a non-loopback a
ERROR: [3] bootstrap checks failed
[1]: max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]
[2]: max number of threads [3853] for user [es] is too low, increase to at least [4096]
[3]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]
[2019-06-26T16:40:35,273][INFO ][o.e.n.Node ] [4fZjnig] stopping ...
[2019-06-26T16:40:35,300][INFO ][o.e.n.Node ] [4fZjnig] stopped
```

1) max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]

每个进程最大同时打开文件数太小，可通过下面2个命令查看当前数量

```
1 | ulimit -Hn
2 | ulimit -Sn
```

```
[es@jackhu bin]$ ulimit -Hn
4096
[es@jackhu bin]$ ulimit -Hs
unlimited
[es@jackhu bin]$ ulimit -Sn
1024
```

修改/etc/security/limits.conf文件，增加配置，用户退出后重新登录生效

1	*	soft	nofile	65536
2	*	hard	nofile	65536

```
[root@jackhu bin]# ulimit -Hn
65536
[root@jackhu bin]# ulimit -Sn
65536
```

2) max number of threads [3818] for user [es] is too low, increase to at least [4096]

可通过命令查看

1	ulimit -Hu
2	ulimit -Su

```
[root@jackhu bin]# ulimit -Hu
3853
[root@jackhu bin]# ulimit -Su
3853
[root@jackhu bin]#
```

问题同上，最大线程个数太低。修改配置文件/etc/security/limits.conf，增加配置

1	`*`	soft	nproc	4096`
2	`*`	hard	nproc	4096`

```
[root@jackhu bin]# ulimit -Hu
4096
[root@jackhu bin]# ulimit -Su
4096
```

3) 、max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]

修改/etc/sysctl.conf文件

1	vi /etc/sysctl.conf
2	sysctl -p #执行命令sysctl -p生效
3	#增加配置vm.max_map_count=262144

```
[root@jackhu bin]# vi /etc/sysctl.conf
# sysctl settings are defined through files in
vm.max_map_count=262144
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
"/etc/sysctl.conf" 11L, 473C written
[root@jackhu bin]# sysctl -p
vm.max_map_count = 262144
[root@jackhu bin]#
```

错误解决完毕：重新启动

← → C ① 不安全 | 192.168.66.66:9200

```
{
  "name": "4fZjnis",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "amdhillZaSm6tgynYGfLGdQ",
  "version": {
    "number": "6.2.4",
    "build_hash": "ccec39f",
    "build_date": "2018-04-12T20:37:28.497551Z",
    "build_snapshot": false,
    "lucene_version": "7.2.1",
    "minimum_wire_compatibility_version": "5.6.0",
    "minimum_index_compatibility_version": "5.0.0"
  },
  "tagline": "You Know, for Search"
}
```

后台启动:

```
1 | ./elasticsearch -d
```

3、容器安装

```
1 #搜索镜像
2 docker search elasticsearch
3 #拉取镜像
4 docker pull elasticsearch:6.2.4
5 #创建容器
6 docker create --name elasticsearch --net host -e "discovery.type=single-
  node" -e
7 "network.host=192.168.66.66" elasticsearch:6.2.4
8 #启动
9 docker start elasticsearch
10 #查看日志
11 docker logs elasticsearch
```

访问容器elasticsearch:

← → C ① 不安全 | 192.168.66.66:9200

```
{
  "name": "4fZjnis",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "amdhillZaSm6tgynYGfLGdQ",
  "version": {
    "number": "6.2.4",
    "build_hash": "ccec39f",
    "build_date": "2018-04-12T20:37:28.497551Z",
    "build_snapshot": false,
    "lucene_version": "7.2.1",
    "minimum_wire_compatibility_version": "5.6.0",
    "minimum_index_compatibility_version": "5.0.0"
  },
  "tagline": "You Know, for Search"
}
```

后台

三、head插件安装

1、head插件主要用途

elasticsearch-head是一个用来浏览、与Elastic Search簇进行交互的web前端展示插件。
elasticsearch-head是一个用来监控Elastic Search状态的客户端插件。

elasticsearch主要有以下三个主要操作——

- 1) 簇浏览，显示簇的拓扑并允许你执行索引(index)和节点层面的操作。
- 2) 查询接口，允许你查询簇并以原始json格式或表格的形式显示检索结果。
- 3) 显示簇状态，有许多快速访问的tabs用来显示簇的状态。
- 4) 支持Restful API接口，包含了许多选项产生感兴趣的结果，包括：
第一，请求方式: get, put, post, delete; json请求数据，节点node，路径path。
第二，JSON验证器。
第三，定时请求的能力。
第四，用javascript表达式传输结果的能力。
第五，统计一段时间的结果或该段时间结果比对的能力。
- 第六，以简单图标的形式绘制传输结果

2、安装

安装步骤：

```
1 #下载nodejs,head插件运行依赖node
2 wget https://nodejs.org/dist/v9.9.0/node-v9.9.0-linux-x64.tar.xz
3 #解压
4 tar -xf node-v9.9.0-linux-x64.tar.xz
5 #重命名
6 mv node-v9.9.0-linux-x64 nodeJs
7 #配置文件
8 vim /etc/profile
9 #刷新配置
10 source /etc/profile
11 #查询node版本，同时查看是否安装成功
12 node -v
13 #下载head插件
14 wget https://github.com/mobz/elasticsearch-head/archive/master.zip
15 #解压
16 unzip master.zip
17 #使用淘宝的镜像库进行下载，速度很快
18 npm install -g cnpm --registry=https://registry.npm.taobao.org
19 #进入head插件解压目录，执行安装命令
20 cnpm install
```

3、运行

```
1 | npm start #启动head插件
```

启动运行端口为：9100

```
[root@jackhu elasticsearch-head-master]# npm start

> elasticsearch-head@0.0.0 start /opt/elasticsearch-head-master
> grunt server

(node:3946) ExperimentalWarning: The http2 module is an experimental API.
Running "connect:server" (connect) task
Waiting forever...
Started connect web server on http://localhost:9100
```

访问：

验证索引及页面详解

集群名称: elk-cluster 集群健康值: green (12 of 12)

zlsindex 索引名称

size: 4.95ki (9.89ki) docs: 1 (2)

主节点和副本节点

elk01 elk02

各个节点及对应节点分片状态

粗框: 主分片 细框: 副本分片

集群状态:

- 绿色: 所有分片都正常
- 黄色: 有副本分片丢失但主分片还在
- 红色: 主分片丢失

此时未连接，需要配置才能连接：

修改 Gruntfile.js文件：

```
[root@jackhu elasticsearch-head-master]# ll
总用量 228
-rw-r--r--. 1 root root 248 6月 25 21:18 Dockerfile
-rw-r--r--. 1 root root 221 6月 25 21:18 Dockerfile-alpine
-rw-r--r--. 1 root root 104 6月 25 21:18 elasticsearch-head.sublime-project
-rw-r--r--. 1 root root 2171 6月 25 21:18 Gruntfile.js
-rw-r--r--. 1 root root 3482 6月 25 21:18 grunt_fileSets.js
-rw-r--r--. 1 root root 1100 6月 25 21:18 index.html
-rw-r--r--. 1 root root 559 6月 25 21:18 LICENCE
drwxr-xr-x. 376 root root 12288 6月 26 18:43 node_modules
-rw-r--r--. 1 root root 886 6月 25 21:18 package.json
-rw-r--r--. 1 root root 169953 6月 26 18:40 package-lock.json
-rw-r--r--. 1 root root 100 6月 25 21:18 plugin-descriptor.properties
drwxr-xr-x. 4 root root 53 6月 25 21:18 proxy
-rw-r--r--. 1 root root 7034 6月 25 21:18 README.textile
drwxr-xr-x. 5 root root 140 6月 25 21:18 _site
drwxr-xr-x. 4 root root 31 6月 25 21:18 src
drwxr-xr-x. 4 root root 70 6月 25 21:18 test
```

修改如下：


```

connect: {
  server: {
    options: {
      port: 9100,
      base: '.',
      keepalive: true,
      hostname: '*'
    }
  }
}

```

修改_site/app.js

修改IP地址，连接elasticsearch

```

app.App = ui.AbstractWidget.extend({
  defaults: {
    base_uri: null
  },
  init: function(parent) {
    this._super();
    this.prefs = services.Preferences.instance();
    this.base_uri = this.config.base_uri || this.prefs.get("app-base_uri") || "http://192.168.66.66:9200";
    if( this.base_uri.charAt( this.base_uri.length - 1 ) !== "/" ) {
      // XHR request fails if the URL is not ending with a "/"
      this.base_uri += "/";
    }
  }
}

```

启用CORS:

当head插件访问es时，您必须在elasticsearch中启用CORS，否则您的浏览器将拒绝跨域。

在elasticsearch配置中:

```
1 http.cors.enabled: true
```

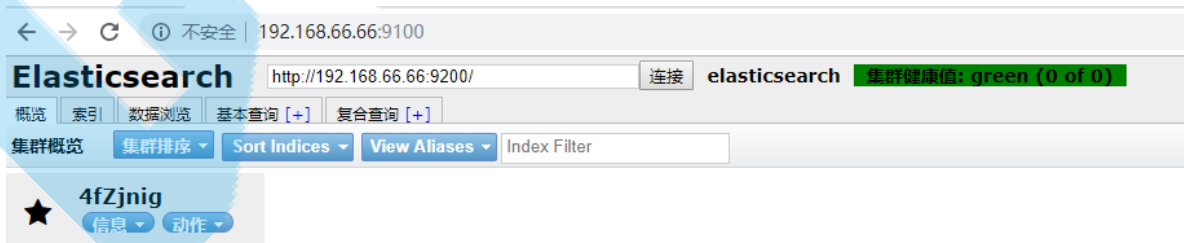
您还必须设置，http.cors.allow-origin因为默认情况下不允许跨域。http.cors.allow-origin: "*" 是允许配置的，但由于这样配置的任何地方都可以访问，所以有安全风险。

我在集群安装的时候已经配好了、如果你刚配置、需要重启ElasticSearch服务

```
1 http.cors.enabled: true
2 http.cors.allow-origin: "*"

```

访问head插件



四、LogStash安装

1、LogStash插件介绍

Logstash是一个具有实时管道的开源数据收集引擎。可以动态地统一不同来源的数据，并将数据归到不同目的地。也是一个管理事件和日志工具。你可以用它来收集日志，分析它们，并将它们储存起来以供以后使用。

Logstash 通常都是和 Kibana 以及 Elasticsearch 一起使用。

2、logStash安装

```
1 #注意版本和elasticsearch,kibana 必须保持一致，es,kibana都是6.2.4版本
2 wget https://artifacts.elastic.co/downloads/logstash/logstash-6.2.4.tar.gz
3 #解压
4 tar -zxvf logstash-6.2.4.tar.gz
5 #启动 基本的 input output
6 #stdin stdout 输入输出插件
7 ./logstash -e 'input{ stdin{} } output{ stdout{} }'
8
9 # codec
10 ./logstash -e 'input{ stdin{} } output{ stdout{ codec => json } }'
11
12 #日志内容写入elasticsearch
13 ./logstash -e 'input{ stdin{} } output{ elasticsearch{hosts =>
14 ["192.168.66.66:9200"]} }'
15
16 #日志内容写入elasticsearch，同时输出
17 #注意elasticsearch插件的语法格式：hosts 对应数组
18 ./logstash -e 'input{ stdin{} } output{ elasticsearch{hosts =>
19 ["192.168.66.66:9200"]} stdout{} }'
```

3、logStash插件

3.1、input插件

Input plugins

An input plugin enables a specific source of events to be read by Logstash.

The following input plugins are available below. For a list of Elastic supported plugins, please consult the [Support Matrix](#).

stdin	Reads events from standard input	logstash-input-stdin
file	Streams events from files	logstash-input-file
http	Receives events over HTTP or HTTPS	logstash-input-http
tcp	Reads events from a TCP socket	logstash-input-tcp

输入比较常见的几个插件：stdin、file、http、tcp

3.2、output插件

Output plugins



An output plugin sends event data to a particular destination. Outputs are the final stage in the event pipeline.

The following output plugins are available below. For a list of Elastic supported plugins, please consult the [Support Matrix](#).

stdout	Prints events to the standard output	logstash-output-stdout
file	Writes events to files on disk	logstash-output-file
http	Sends events to a generic HTTP or HTTPS endpoint	logstash-output-http
tcp	Writes events over a TCP socket	logstash-output-tcp

把日志内容输出到elasticsearch插件:

elasticsearch	Stores logs in Elasticsearch	logstash-output-elasticsearch
-------------------------------	------------------------------	---

3.3、codec插件

Codec (Code Decode) Plugin作用于input和output plugin, 负责将数据在原始与Logstash之间转换, 常见的codec有:

- plain** 读取原始内容
- dots** 将内容简化为点进行输出
- rubydebug** 将内容按照ruby格式输出, 方便调试
- line** 处理带有换行符的内容
- json** 处理json格式的内容
- multiline** 处理多行数据的内容

4、logStash配置

4.1、创建配置

```
[root@jackhu config]# touch logstash.conf
[root@jackhu config]# pwd
/root/.logstash-6.2.4/config
[root@jackhu config]# ll
总用量 28
-rw-r--r--. 1 root root 1846 4月 13 2018 jvm.options
-rw-r--r--. 1 root root 4466 4月 13 2018 log4j2.properties
-rw-r--r--. 1 root root 0 10月 11 15:13 logstash.conf
-rw-r--r--. 1 root root 6346 4月 13 2018 logstash.yml
-rw-r--r--. 1 root root 3244 4月 13 2018 pipelines.yml
-rw-r--r--. 1 root root 1702 4月 13 2018 startup.options
```

在config目录下建立logstash.conf配置文件

4.2、配置语法

Configuring Logstash



To configure Logstash, you create a config file that specifies which plugins you want to use and settings for each plugin. You can reference event fields in a configuration and use conditionals to process events when they meet certain criteria. When you run logstash, you use the `-f` to specify your config file.

Let's step through creating a simple config file and using it to run Logstash. Create a file named "logstash-simple.conf" and save it in the same directory as Logstash.

```
input { stdin { } }
output {
  elasticsearch { hosts => ["localhost:9200"] }
  stdout { codec => rubydebug }
}
```

开始配置:

```
1 input { stdin { } }
2 output {
3   elasticsearch { hosts => ["192.168.66.66:9200"] }
4   stdout { codec => rubydebug }
5 }
6
7 #启动命令
8 bin/logstash -f config/logstash.conf
```

5、file日志收集

```
1 #建立新的配置文件
2 mv logstash.conf file.conf
3 #详细配置如下
4 input {
5   file{
6     path => "/var/log/messages" #收集messages文件日志
7     type => "system"
8     start_position => "beginning" #记录上次收集的位置
9   }
10 }
11 output {
12   elasticsearch {
13     hosts => ["192.168.66.66:9200"] #写入elasticsearch的地址
14     index => "system-%{+YYYY.MM.dd}" #定义索引的名称
15   }
16   stdout { codec => rubydebug }
17 }
18
19 #启动logstash,配置文件名字叫什么无所谓
20 bin/logstash -f config/file.conf
```

收集效果如下所示:



注：二次启动logstash时，日志文件不会重复读取。因为系统以及记录了文件读取的位置。如果想重复读取需要删除sincedb。

使用：find / -name .sincedb_*

```
1 [root@es129 log]# find / -name .sincedb_*
2 /opt/logstash-
  6.2.4/data/plugins/inputs/file/.sincedb_0d03f373e2679dd1b3903fab34424ae
```

6、项目日志

1)、logstash收集日志

```
1 #通过tcp协议输入
2 input {
3   tcp {
4     port => 9601
5     codec => json
6   }
7 }
8 output {
9   elasticsearch {
10    hosts => ["192.168.66.66:9200"]
11    index => "kkb-log-%{+YYYY.MM.dd}"
12  }
13   stdout {codec => rubydebug }
14 }
```

五、Kibana安装

1、kibana 插件介绍

kibana 插件提供了Marvel监控的UI界面。

kibana是一个与elasticsearch一起工作的开源的分析和可视化的平台。使用kibana可以查询、查看并与存储在elasticsearch索引的数据进行交互操作。使用kibana能执行高级的数据分析，并能以图表、表格和地图的形式查看数据。 kibana使得理解大容量的数据变得非常容易。它非常简单，基于浏览器的接口使我们能够快速创建和分享显示elasticsearch查询结果实时变化的仪表盘。

2、kibana下载

下载命令：

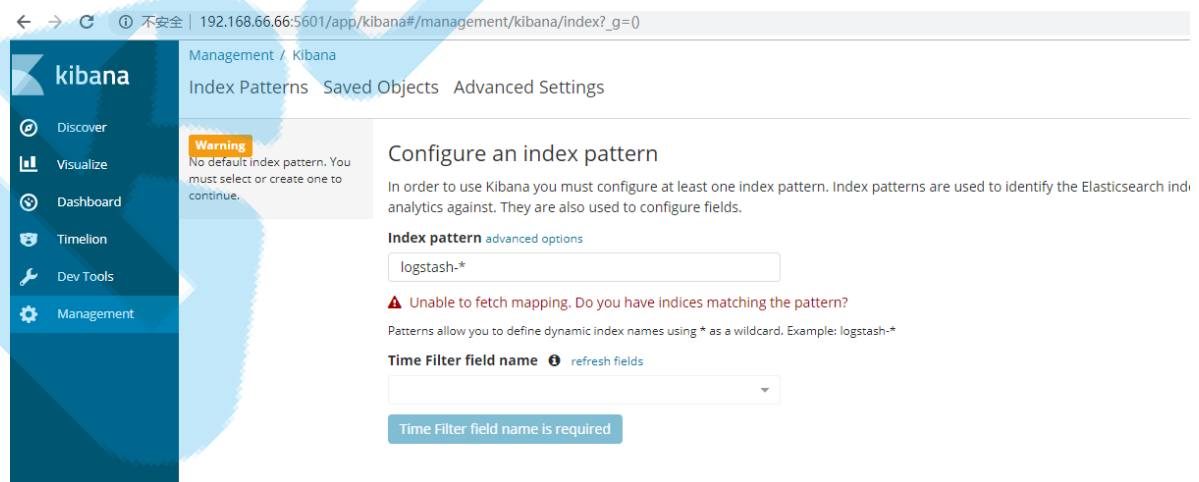
```
1 #kibana版本必须和es版本一致，否则监控不到es
2 wget https://artifacts.elastic.co/downloads/kibana/kibana-6.2.4-linux-x86_64.tar.gz
3 #将shasum生产的SHA与已发布的SHA进行比较。
4 shasum -a 512 kibana-6.4.2-linux-x86_64.tar.gz
5 tar -xzf kibana-6.4.2-linux-x86_64.tar.gz
6 # 归档包解压的目录为$KIBANA_HOME
7 cd kibana-6.4.2-linux-x86_64/
```

3、环境配置

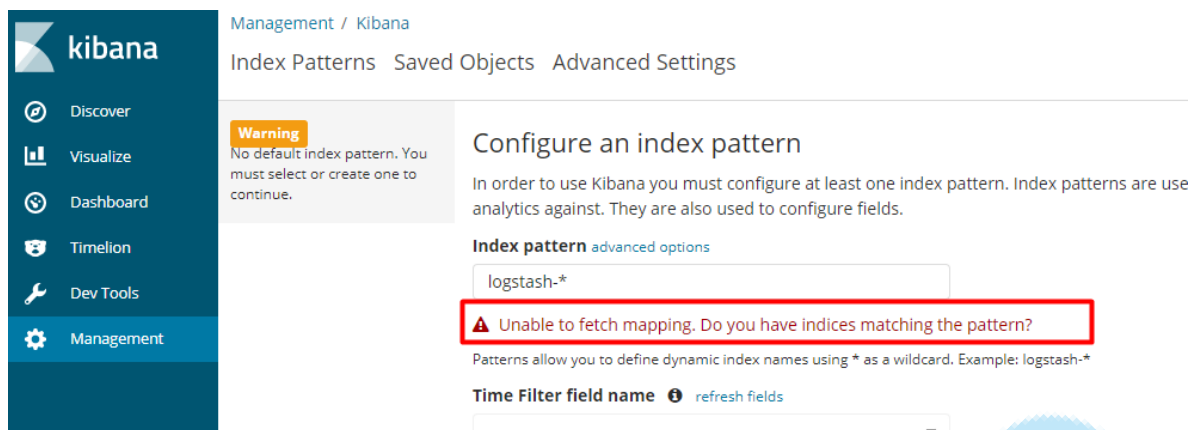
详细配置地址：<https://www.elastic.co/guide/en/kibana/5.6/settings.html>

```
1 # 将默认配置改成如下：
2 server.port: 5601
3 server.host: "192.168.66.66"
4 #修改成自己集群的端口号及IP
5 elasticsearch.url: "http://192.168.66.66:9200"
6 kibana.index: ".kibana"
```

4、安装成功



##5、kibana使用



可以看到这里有个红色的错误：“Unable to fetch mapping. Do you have indices matching the pattern?”，这个是因为在elasticsearch中还没有任何有关logstash-*格式相关的数据，所以这里才提示报错，可以暂时忽略。

初次使用我们可以进行一些简单的测试，比如查看elasticsearch集群的状态，在Kibana控制台中运行命令GET /_cat/health?v，得到如下所示：

1	GET	/_cat/health?v								
1	epoch	timestamp	cluster	status	node.total	node.data	shards	pri	relo	init
2	unassign	pending_tasks	max_task_wait_time	active_shards_percent						
2	1562842677	18:57:57	elasticsearch	yellow	1	1	6	6	0	0
3	6	0				50.0%				

6、日志视图

1) 创建日志

Step 1 of 2: Define index pattern

Index pattern

es-info*

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

> Next step

✓ Success! Your index pattern matches 1 index.

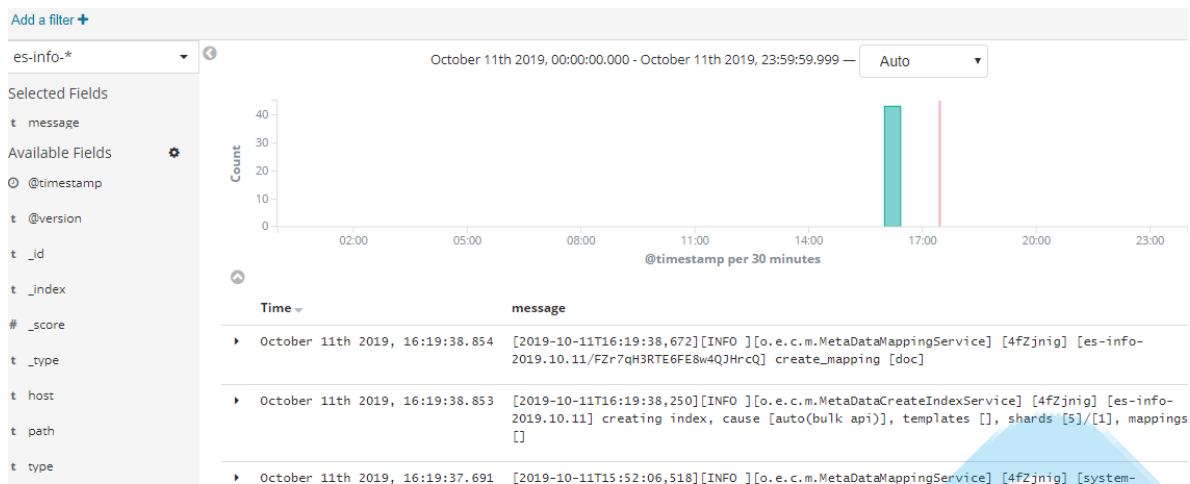
es-info-2019.10.11

Rows per page: 10

system-2019.10.11 size: 1.57Mi (1.57Mi) docs: 6,029 (6,029) 信息 动作	logstash-2019.10.11 size: 21.4ki (21.4ki) docs: 5 (5) 信息 动作	kbb size: 14.5ki (14.5ki) docs: 2 (2) 信息 动作	es-info-2019.10.11 size: 67.5ki (67.5ki) docs: 43 (43) 信息 动作
0 1 2 3 4	0 1 2 3 4	0 1	0 1 2 3 4
0 1 2 3 4	0 1 2 3 4	0 1	0 1 2 3 4

收集索引库那个日志
名字匹配

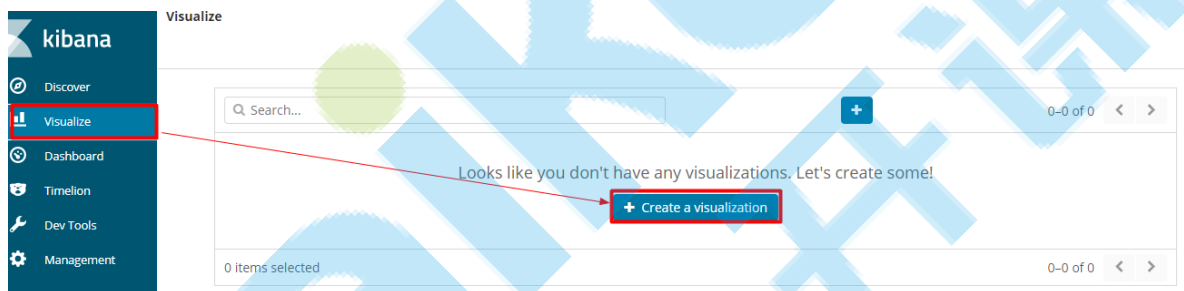
可以发现，日志可视化展示以及成功：



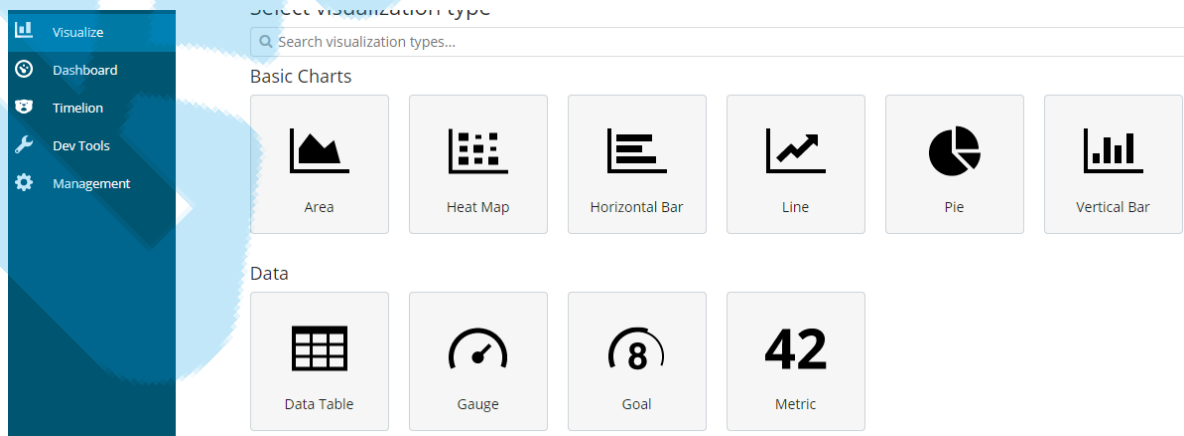
7、可视化数据

在Visualize应用程序中，你可以使用各种图表、表格和地图等来塑造数据，你将创建四个可视化效果：饼图、柱状图、坐标图和Markdown小部件。

1) 点击Visualize



2) 点击相应图标



3) 选择new search

From a New Search, Select Index

Filter... 2 of 2

Name ▲

*

kkb* 选择其中一个

Or, From a Saved Search

Saved Searches Filter... 0-0 of 0 Manage saved searches

Name ▲

No matching saved searches found.

Add a filter +

kkb*

Data Options

Metrics

Slice Size Count

Buckets

Select buckets type

Split Slices

Split Chart

Cancel

4) 保存

Visualize / 哈哈哈哈哈 (unsaved) Save Share Refres

Save Visualization

哈哈哈哈哈

Save

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

kkb*

Data Options

Metrics

Slice Size Count

Buckets

5) 添加视图模块

kibana Dashboard

Discover

Visualize

Dashboard

Timeline

Dev Tools

Management

添加视图模块

Search...

+

1-1 of 1

Name ↑	Description	Actions
哈哈哈哈哈		Edit

1-1 of 1

This dashboard is empty. Let's fill it up!

Click the **Add** button in the menu bar above to add a visualization to the dashboard.

If you haven't set up any visualizations yet, [visit the Visualize app](#) to create your first visualization.

Add Panels

Visualization

Saved Search

Visualizations Filter...

1-2 of 2

Add new Visualization

Name

哈哈哈哈哈

选择要添加的模块名称

测试一个图形