

# ELK 实时日志平台 安装部署手册

by

author: 孟子浩

team: 郑州信源运维部

version: V2.2.0

# 目 录

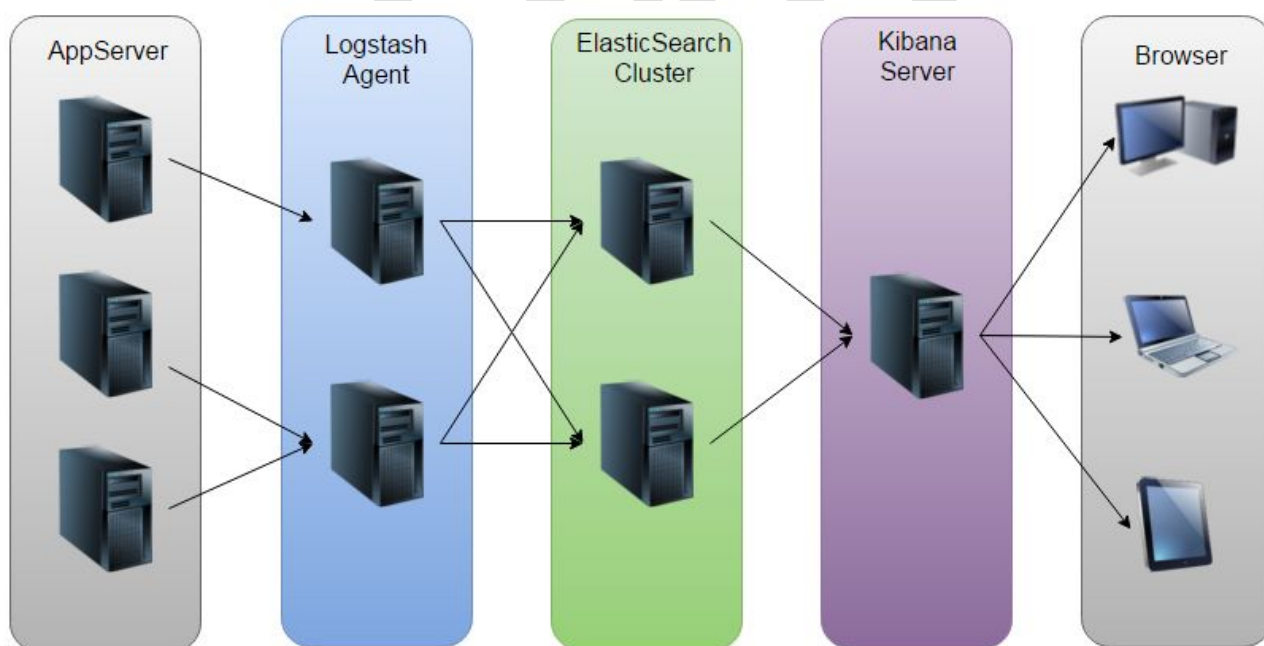
前言.....	4
第一章 工作原理图.....	4
第二章 安装部署.....	5
第一节 安装准备.....	5
第二节 安装环境列表.....	5
2.2.1 操作系统: Linux redhat 6.4.....	5
2.2.2 JDK 安装包: jdk-8u11-linux-x64.tar.gz.....	5
2.2.3 Elasticsearch 安装包: elasticsearch-5.2.2.zip.....	5
2.2.4 Logstash 安装包: logstash-5.2.2.zip.....	5
2.2.5 Kibana 安装包: kibana-5.2.2-linux-x86_64.tar.gz.....	6
2.2.6 X-pack 安装包: x-pack-5.2.2.zip.....	6
第三节 Linux 环境配置.....	6
2.3.1 创建 elk 用户和组.....	6
2.3.2 创建并移动部署包到 elk 所需目录及赋予相应权限.....	6
2.3.3 安装并配置 JDK1.8.....	6
2.3.4 配置操作系统相关参数及资源限制.....	7
第四节 Elasticsearch.....	8
2.4.1 解压并配置.....	8
2.4.2 安装 X-pack 扩展包.....	8
2.4.3 启动 Elasticsearch.....	9
第五节 Kibana.....	9
2.5.1 解压并配置.....	9
2.5.2 启动 kibana.....	10
第六节 Logstash.....	11
2.6.1 解压并配置.....	11
2.6.2 启动 Logstash.....	12
2.6.3 API.....	14
第三章 Kibana 数据展示.....	14
第一节 登陆 <a href="http://192.168.8.201:8888">http://192.168.8.201:8888</a> , 并输入账号和密码。.....	14
第二节 选择 Management-->Index Patterns 添加索引。.....	15
第三节 选择 Discover 查看数据展示。(第五章详讲).....	16
第四节 选择 Monitoring 可对服务器运行状态、性能指标进行监控。.....	16
第五节 对数据进行自定义展示, 效果如下图。(第五章详讲).....	17
第四章 架构升级.....	17
第一节 Filebeat.....	17
4.1.1 优化后部署架构图.....	18
4.1.2 安装并配置.....	18
第五章 平台使用维护及数据分析.....	19

第一节 电商 weblogic 应用数据采集.....	19
第二节 电商 weblogic 应用数据展示.....	19
第三节 电商 weblogic 应用数据作图.....	21
第四节 电商 weblogic 应用数据搜索.....	23
第五节 平台日常维护.....	23
5.5.1 定期检查 ELK 服务器上磁盘使用情况.....	23
5.5.2 定期检查 JVM 的运行情况.....	23
5.5.3 定期清理 ELK 服务器上缓存日志.....	23
5.5.4 使用脚本一键操作停启服务.....	24
第六章 总结.....	27

# 前言

为满足**电子商务系统**日益增长的数据量及减轻运维人员的工作量，引入日志集中分析展示平台 ELK，集中收集展示电商中间件 weblogic、tomcat 产生的数据日志。ELK 集 Elasticsearch、Logstash、Kibana 三大开源工具为一体，**Elasticsearch** 是个开源分布式搜索引擎，它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful 风格接口，多数据源，自动搜索负载等；**Logstash** 是一个完全开源的工具，它可以对你的日志进行收集、过滤，并将其存储供以后使用（如，搜索）；**Kibana** 也是一个开源和免费的工具，它可以为 Logstash 和 Elasticsearch 提供的日志分析友好的 Web 界面，可以帮助您汇总、分析和搜索重要数据日志。

## 第一章 工作原理图



如图所示：Logstash 收集 AppServer 产生的 log 日志，并存放放到 Elasticsearch 集群中，而 Kibana 则从 ES 集群中查询数据生成图表，再返回给 Browser。

## 第二章 安装部署

### 第一节 安装准备

Elasticsearch 下载地址: <https://www.elastic.co/downloads/elasticsearch>

Logstash 下载地址: <https://www.elastic.co/downloads/logstash>

Kibana 下载地址: <https://www.elastic.co/downloads/kibana>

### 第二节 安装环境列表

#### 2.2.1 操作系统: Linux redhat 6.4

```
[root@elk01 桌面]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 6.4 (Santiago)
```

#### 2.2.2 JDK 安装包: jdk-8u11-linux-x64.tar.gz

```
[es@elk01 ~]$ java -version
java version "1.8.0_11"
Java(TM) SE Runtime Environment (build 1.8.0_11-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.11-b03, mixed mode)
```

#### 2.2.3 Elasticsearch 安装包: elasticsearch-5.2.2.zip

 elasticsearch-5.2.2.zip	2017/10/18 19:25	好压 ZIP 压缩文件	33,032 KB
---	------------------	-------------	-----------


#### 2.2.4 Logstash 安装包: logstash-5.2.2.zip

 logstash-5.2.2.zip	2017/10/18 21:24	好压 ZIP 压缩文件	97,724 KB
--	------------------	-------------	-----------

## 2.2.5 Kibana 安装包: kibana-5.2.2-linux-x86\_64.tar.gz

 kibana-5.2.2-linux-x86\_64.tar.gz 2017/10/18 21:23 好压 GZ 压缩文件 37,943 KB

## 2.2.6 X-pack 安装包: x-pack-5.2.2.zip

 x-pack-5.2.2.zip 2017/10/18 17:09 好压 ZIP 压缩文件 122,141 KB

## 第三节 Linux 环境配置

### 2.3.1 创建 elk 用户和组

```
[root@elk01 ~]# groupadd elk
[root@elk01 ~]# useradd -g elk elk
[root@elk01 ~]# id elk
uid=500(elk) gid=500(elk) 组=500(elk)
```

### 2.3.2 创建并移动部署包到 elk 所需目录及赋予相应权限

```
[root@elk01 ~]# mkdir /elk201710
[root@elk01 ~]# chown -R elk:elk /elk201710/
[root@elk01 ~]# ll /elk201710/
总用量 446140
-rw-r--r--. 1 elk elk 33824223 10 月 18 19:34 elasticsearch-5.2.2.zip
-rw-r--r--. 1 elk elk 159019376 10 月 18 19:34 jdk-8u11-linux-x64.tar.gz
-rw-r--r--. 1 elk elk 38853061 10 月 18 19:34 kibana-5.2.2-linux-x86_64.tar.gz
-rw-r--r--. 1 elk elk 100068713 10 月 18 19:35 logstash-5.2.2.zip
-rw-r--r--. 1 elk elk 125071734 10 月 18 19:35 x-pack-5.2.2.zip
```

### 2.3.3 安装并配置 JDK1.8

```
[root@elk01 ~]# su - elk
[elk@elk01 ~]$ cd /elk201710/
[elk@elk01 elk201710]$ tar -zxvf jdk-8u11-linux-x64.tar.gz
```

```
[elk@elk01 elk201710]$ vim /home/elk/.bash_profile
#add
export JAVA_HOME=/elk201710/jdk1.8.0_11
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
[elk@elk01 elk201710]$ source /home/elk/.bash_profile
[elk@elk01 elk201710]$ java -version
java version "1.8.0_11"
Java(TM) SE Runtime Environment (build 1.8.0_11-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.11-b03, mixed mode)
```

## 2.3.4 配置操作系统相关参数及资源限制

```
[root@elk01 elk201710]$ vim /etc/security/limits.conf
#End of file
*          soft    nofile    102400
*          hard    nofile    102400
*          soft    nproc     102400
*          hard    nproc     102400

*          hard    core      1000000
*          soft    core      1000000
[root@elk01 ~]# vim /etc/sysctl.conf
#add
vm.max_map_count=655360
[root@elk01 ~]# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
vm.max_map_count = 655360
[root@elk01 ~]# vim /etc/security/limits.d/90-nproc.conf
#add
*          soft    nproc     2048
```

## 第四节 Elasticsearch

### 2.4.1 解压并配置

```
[elk@elk01 elk201710]$ unzip elasticsearch-5.2.2.zip
[elk@elk01 elasticsearch-5.2.2]$ vim config/elasticsearch.yml
#add
path.data: /elk201710/es/data
path.logs: /elk201710/es/logs
bootstrap.system_call_filter: false
network.host: 192.168.8.201
http.port: 9200
http.cors.enabled: true
http.cors.allow-origin: ""
```

### 2.4.2 安装 X-pack 扩展包

```
[elk@elk01 elasticsearch-5.2.2]$ bin/elasticsearch-plugin install file:///elk201710/x-pack-5.2.2.zip
-> Downloading file:///elk201710/x-pack-5.2.2.zip
[=====] 100%
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@
@      WARNING: plugin requires additional permissions      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@
* java.lang.RuntimePermission accessClassInPackage.com.sun.activation.registries
* java.lang.RuntimePermission getClassLoader
* java.lang.RuntimePermission setContextClassLoader
* java.lang.RuntimePermission setFactory
* java.security.SecurityPermission createPolicy.JavaPolicy
* java.security.SecurityPermission getPolicy
* java.security.SecurityPermission putProviderProperty.BC
* java.security.SecurityPermission setPolicy
* java.util.PropertyPermission * read,write
* java.util.PropertyPermission sun.nio.ch.bugLevel write
* javax.net.ssl.SSLPermission setHostnameVerifier
See http://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.
```



Continue with installation? [y/N]y

-> Installed x-pack

## 2.4.3 启动 Elasticsearch

```
[elk@elk01 elasticsearch-5.2.2]$ bin/elasticsearch
```

```
[2017-10-18T20:16:37,765][INFO ][o.e.n.Node               ] [] initializing ...
[2017-10-18T20:16:37,973][INFO ][o.e.e.NodeEnvironment ] [jHpNwJ4] using [1] data paths, mounts [(/dev/mapper/vg_elk01-lv_root)], net usable_space [39gb], net total_space [44.8gb], spins? [possibly], types [ext4]
[2017-10-18T20:16:37,973][INFO ][o.e.e.NodeEnvironment ] [jHpNwJ4] heap size [1.9gb], compressed ordinary object pointers [true]
[2017-10-18T20:16:37,980][INFO ][o.e.n.Node               ] node name [jHpNwJ4] derived from node ID [jHpNwJ4DSPihWWj35MzVPg]; set [node.name] to override
[2017-10-18T20:16:37,988][INFO ][o.e.n.Node               ] version[5.2.2], pid[3280], build[f9d9b74/2017-02-24T17:26:45.835Z], OS[Linux/2.6.32-358.el6.x86_64/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_11/25.11-b03]
[2017-10-18T20:16:42,450][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [aggs-matrix-stats]
[2017-10-18T20:16:42,450][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [ingest-common]
[2017-10-18T20:16:42,451][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [lang-expression]
[2017-10-18T20:16:42,451][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [lang-groovy]
[2017-10-18T20:16:42,454][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [lang-mustache]
[2017-10-18T20:16:42,456][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [lang-painless]
[2017-10-18T20:16:42,456][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [percolator]
[2017-10-18T20:16:42,458][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [reindex]
[2017-10-18T20:16:42,462][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [transport-netty3]
[2017-10-18T20:16:42,463][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded module [transport-netty4]
[2017-10-18T20:16:42,470][INFO ][o.e.p.PluginsService   ] [jHpNwJ4] loaded plugin [x-pack]
[2017-10-18T20:16:46,995][DEBUG][o.e.a.ActionModule      ] Using REST wrapper from plugin org.elasticsearch.xpack.XPackPlugin
[2017-10-18T20:16:50,140][INFO ][o.e.n.Node               ] initialized
[2017-10-18T20:16:50,141][INFO ][o.e.n.Node               ] [jHpNwJ4] starting ...
```

## 第五节 Kibana

### 2.5.1 解压并配置

```
[elk@elk01 elk201710]$ tar -zxvf kibana-5.2.2-linux-x86_64.tar.gz
```

```
[elk@elk01 kibana-5.2.2-linux-x86_64]$ vim config/kibana.yml
```

```
#add
server.port: 8888
server.host: "192.168.8.201"
elasticsearch.url: "http://192.168.8.201:9200"
kibana.index: ".kibana"
```

## 2.5.2 启动 kibana

```
[elk@elk01 kibana-5.2.2-linux-x86_64]$ ./bin/kibana
```

```
log [05:57:30.279] [info][status][plugin:kibana@5.2.2] Status changed from uninitialized to green - Ready
log [05:57:30.505] [info][status][plugin:elasticsearch@5.2.2] Status changed from uninitialized to yellow -
Waiting for Elasticsearch
log [05:57:30.589] [info][status][plugin:xpack_main@5.2.2] Status changed from uninitialized to yellow -
Waiting for Elasticsearch
log [05:57:30.641] [info][status][plugin:elasticsearch@5.2.2] Status changed from yellow to green - Kibana
index ready
log [05:57:30.658] [info][status][plugin:graph@5.2.2] Status changed from uninitialized to yellow - Waiting
for Elasticsearch
log [05:57:30.676] [info][license][xpack] Imported license information from Elasticsearch: mode: trial |
status: active | expiry date: 2017-11-17T20:16:55+08:00
log [05:57:30.690] [info][status][plugin:xpack_main@5.2.2] Status changed from yellow to green - Ready
log [05:57:30.692] [info][status][plugin:graph@5.2.2] Status changed from yellow to green - Ready
log [05:57:30.723] [info][status][plugin:monitoring@5.2.2] Status changed from uninitialized to yellow -
Waiting for Monitoring Health Check
log [05:57:30.745] [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 [05:57:30.757] [info][status][plugin:reporting@5.2.2] Status changed from uninitialized to green - Ready
log [05:57:30.824] [info][status][plugin:security@5.2.2] Status changed from uninitialized to green - Ready
log [05:57:30.836] [warning][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 [05:57:30.847] [warning][security] Session cookies will be transmitted over insecure connections. This is
not recommended.
log [05:57:30.939] [info][status][plugin:searchprofiler@5.2.2] Status changed from uninitialized to green -
Ready
log [05:57:30.953] [info][status][plugin:monitoring@5.2.2] Status changed from yellow to green - Ready
log [05:57:31.164] [info][status][plugin:tilemap@5.2.2] Status changed from uninitialized to green - Ready
log [05:57:31.188] [info][status][plugin:console@5.2.2] Status changed from uninitialized to green - Ready
log [05:57:31.554] [info][status][plugin:timelion@5.2.2] Status changed from uninitialized to green - Ready
```

## 第六节 Logstash

环境搭建成功后，在服务器进行本地日志收集，做测试环境。以 192.168.8.201 为例，少数日志测试平台运行，设置最基本的客户端配置。

### 2.6.1 解压并配置

```
[elk@elk01 elk201710]$ unzip logstash-5.2.2.zip
```

```
[elk@elk01 logstash-5.2.2]$ vim test/test3.conf
```

```
input{
  file{
    path => ["/elk201710/ebs/ebs01.log"]
    type => "ebslog"
    codec => multiline{
      pattern => "^[^%{NOTSPACE} %{NUMBER}"
      negate => true
      what => "previous"
    }
  }
}

filter{
  # codec=>rubydebug
  mutate{
    split=>["message","> <"]
    add_field => {
      "date" => "%{[message][0]}"
    }
    add_field => {
      "severity" => "%{[message][1]}"
    }
    add_field => {
      "stdin" => "%{[message][2]}"
    }
    add_field => {
      "hostname" => "%{[message][3]}"
    }
    add_field => {
      "servername" => "%{[message][4]}"
    }
    add_field => {
      "log4j2" => "%{[message][5]}"
    }
    add_field => {
```

```

        "kzt" => "%{[message][6]}"
    }
    add_field => {
        "threadid" => "%{[message][7]}"
    }
    add_field => {
        "userid" => "%{[message][8]}"
    }
    add_field => {
        "affairid" => "%{[message][9]}"
    }
    add_field => {
        "contxtid" => "%{[message][10]}"
    }
    add_field => {
        "information" => "%{[message][11]}"
    }
}

grok{
    match %> %{"information"
=>"%{TIMESTAMP_ISO8601:log_timestamp} %{NOTSPACE:operation} %{NOTSPACE:LOGLEVEL} %{GREEDYDATA:we
b_log}" }
}
mutate{ remove_field =>"information"
        remove_field =>"message"
}

}

output{
    elasticsearch {
        hosts => ["192.168.8.201:9200"]
        index => "ebslog-%{+YYYY.MM.dd}"
        user => elastic
        password => changeme
    }
}

```

## 2.6.2 启动 Logstash

```
[elk@elk01 logstash-5.2.2]$ ./bin/logstash -f test/test3.conf
```

Sending Logstash's logs to /elk201710/logstash-5.2.2/logs which is now configured via log4j2.properties

[2017-10-21T14:57:12,875][INFO ][logstash.outputs.elasticsearch] Elasticsearch pool URLs updated  
{:changes=>{:removed=>[], :added=>[http://elastic:xxxxxx@192.168.8.201:9200/]}

[2017-10-21T14:57:12,885][INFO ][logstash.outputs.elasticsearch] Running health check to see if an Elasticsearch connection is working {healthcheck\_url=>http://elastic:xxxxxx@192.168.8.201:9200/, :path=>"/"}

[2017-10-21T14:57:13,325][WARN ][logstash.outputs.elasticsearch] Restored connection to ES instance  
{url=>#<URI::HTTP:0x1c5bb88b URL:http://elastic:xxxxxx@192.168.8.201:9200/>}

[2017-10-21T14:57:13,328][INFO ][logstash.outputs.elasticsearch] Using mapping template from {path=>nil}

[2017-10-21T14:57:13,481][INFO ][logstash.outputs.elasticsearch] Attempting to install template  
{:manage\_template=>{"template"=>"logstash-\*", "version"=>50001, "settings"=>{"index.refresh\_interval"=>"5s"},  
"mappings"=>{"\_default\_"=>{"\_all"=>{"enabled"=>true, "norms"=>false},  
"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"}}}], "properties"=>{"@timestamp"=>{"type"=>"date",  
"include\_in\_all"=>false}, "@version"=>{"type"=>"keyword", "include\_in\_all"=>false}, "geoip"=>{"dynamic"=>true,  
"properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo\_point"}, "latitude"=>{"type"=>"half\_float"},  
"longitude"=>{"type"=>"half\_float"}}]}}}

[2017-10-21T14:57:13,500][INFO ][logstash.outputs.elasticsearch] New Elasticsearch output  
{:class=>"LogStash::Outputs::ElasticSearch", :hosts=>[#<URI::Generic:0x72f2a1fe URL://192.168.8.201:9200>]}

[2017-10-21T14:57:13,589][INFO ][logstash.pipeline] Starting pipeline {"id"=>"main",  
"pipeline.workers"=>2, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>250}

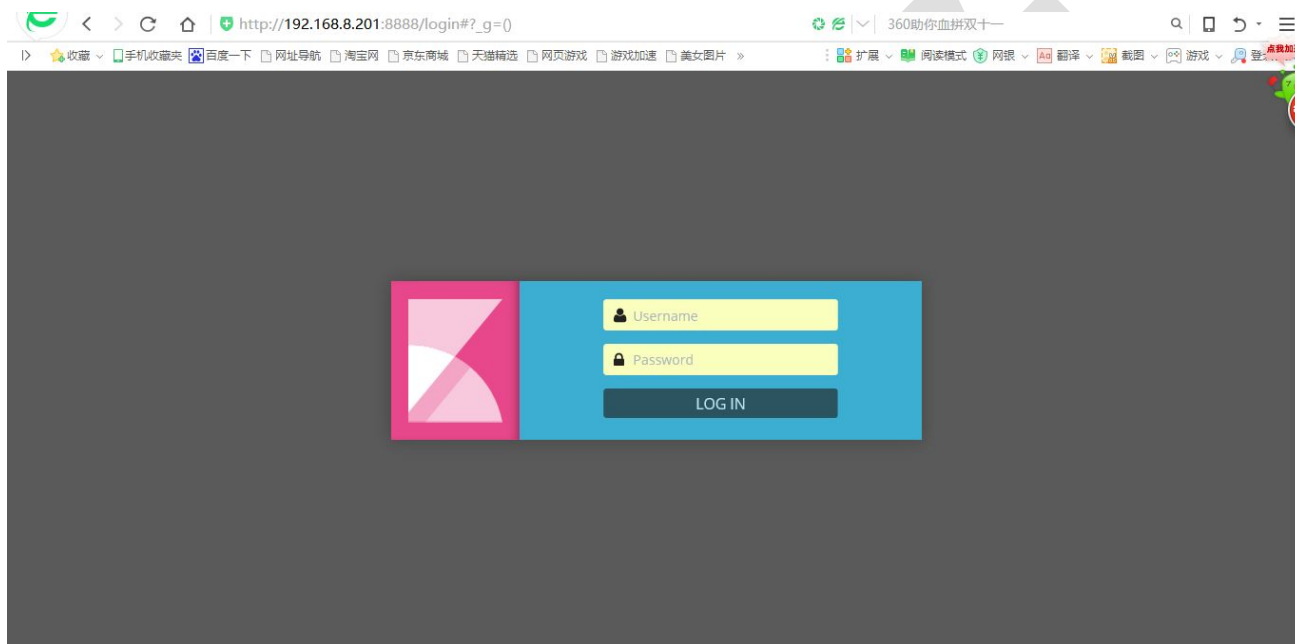
[2017-10-21T14:57:13,906][INFO ][logstash.pipeline] Pipeline main started

[2017-10-21T14:57:14,077][INFO ][logstash.agent] Successfully started Logstash API endpoint  
{:port=>9601}

## 2.6.3 API

# 第三章 Kibana 数据展示

## 第一节 登陆 <http://192.168.8.201:8888>，并输入账号和密码。



## 第二节 选择 Management-->Index Patterns 添加索引。

The image shows two screenshots of the Kibana web interface. The top screenshot displays the 'Management' page with a sidebar on the left containing links to Discover, Visualize, Dashboard, Timeline, Graph, Dev Tools, Monitoring, and Management. The main content area shows 'Version: 5.2.2' and a list of items including 'Elasticsearch', 'Users', 'Roles', 'Kibana', 'Index Patterns', 'Saved Objects', 'Reporting', and 'Advanced Settings'. The bottom screenshot shows the 'Configure an index pattern' page. The sidebar is the same. The main content area has a title 'Configure an index pattern' and a description: 'In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.' Below this, there are several configuration options: a checked checkbox 'Index contains time-based events', an unchecked checkbox 'Use event times to create index names [DEPRECATED]', a text input field for 'Index name or pattern' containing 'ebslog-\*', an unchecked checkbox 'Do not expand index pattern when searching (Not recommended)', and a dropdown menu for 'Time-field name' set to '@timestamp'. A 'Create' button is at the bottom.

Management

Version: 5.2.2

Elasticsearch

Users Roles

Kibana

Index Patterns Saved Objects Reporting Advanced Settings

Management / Kibana

Index Patterns Saved Objects Reporting Advanced Settings

★ es-log-\*  
ebslog-\*  
system-log-\*

### Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

☒ Index contains time-based events  
☐ Use event times to create index names [DEPRECATED]

**Index name or pattern**  
Patterns allow you to define dynamic index names using \* as a wildcard. Example: logstash-\*

ebslog-\*

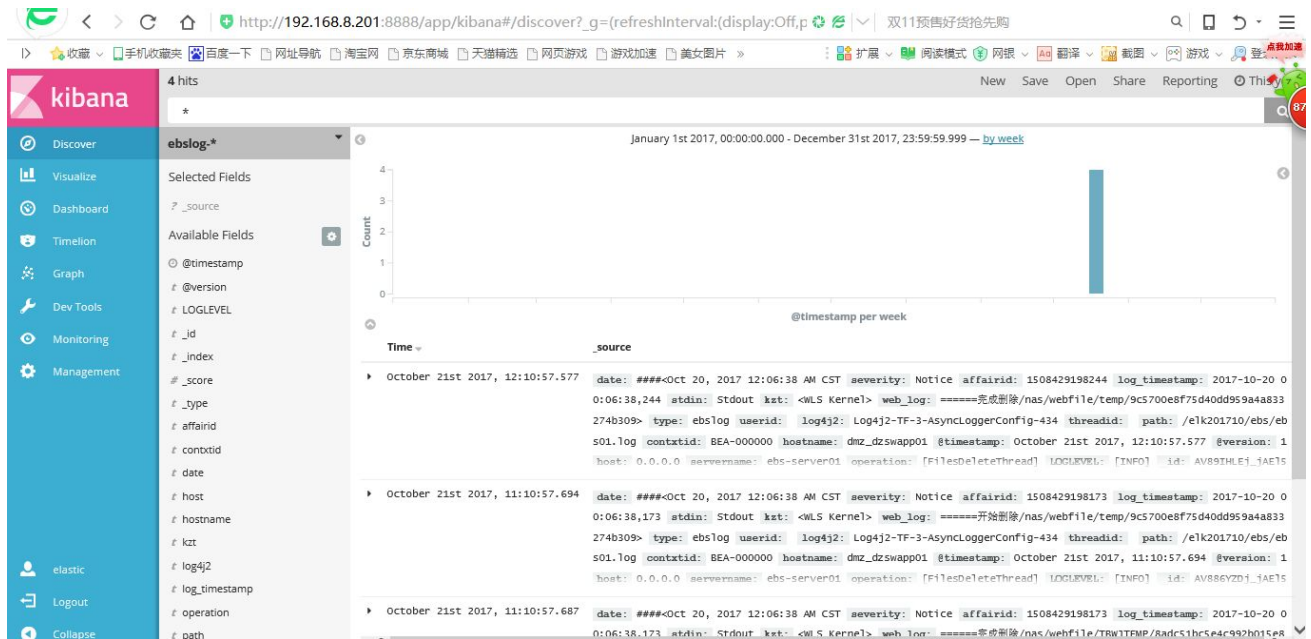
☐ Do not expand index pattern when searching (Not recommended)  
By default, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.

Searching against the index pattern *logstash-\** will actually query elasticsearch for the specific matching indices (e.g. *logstash-2015.12.27*) that fall within the current time range.

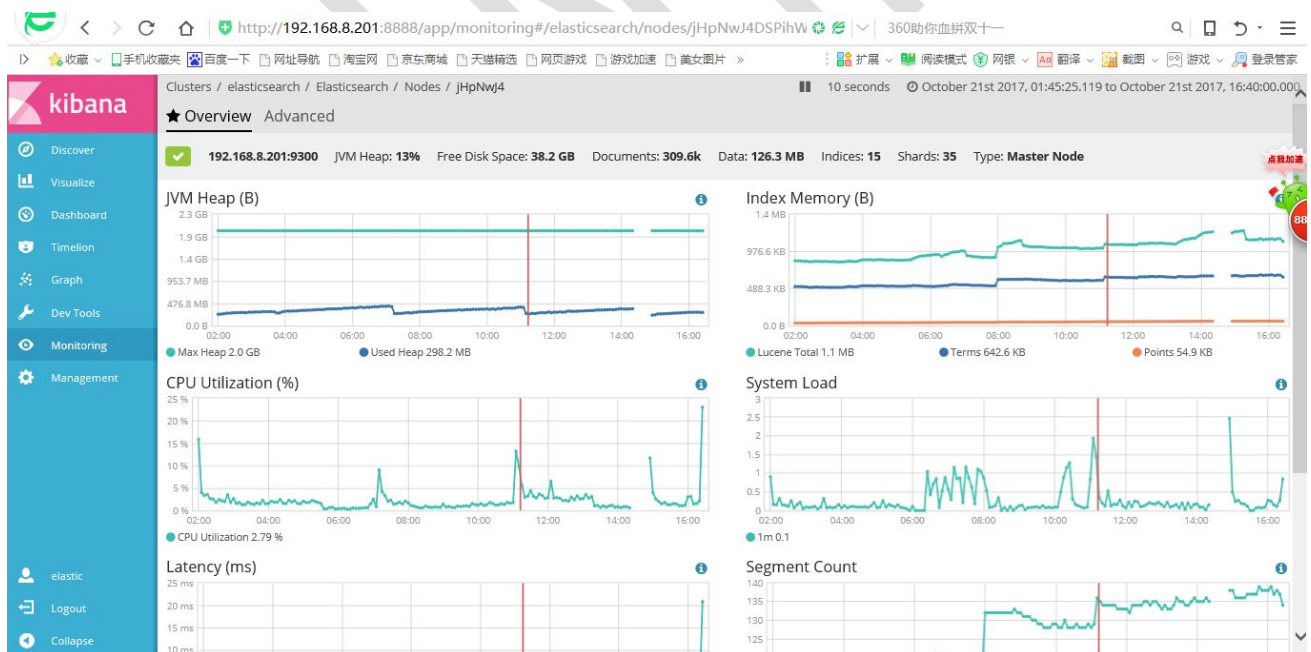
**Time-field name** [refresh fields](#)  
@timestamp

Create

### 第三节 选择 Discover 查看数据展示。（第五章详讲）

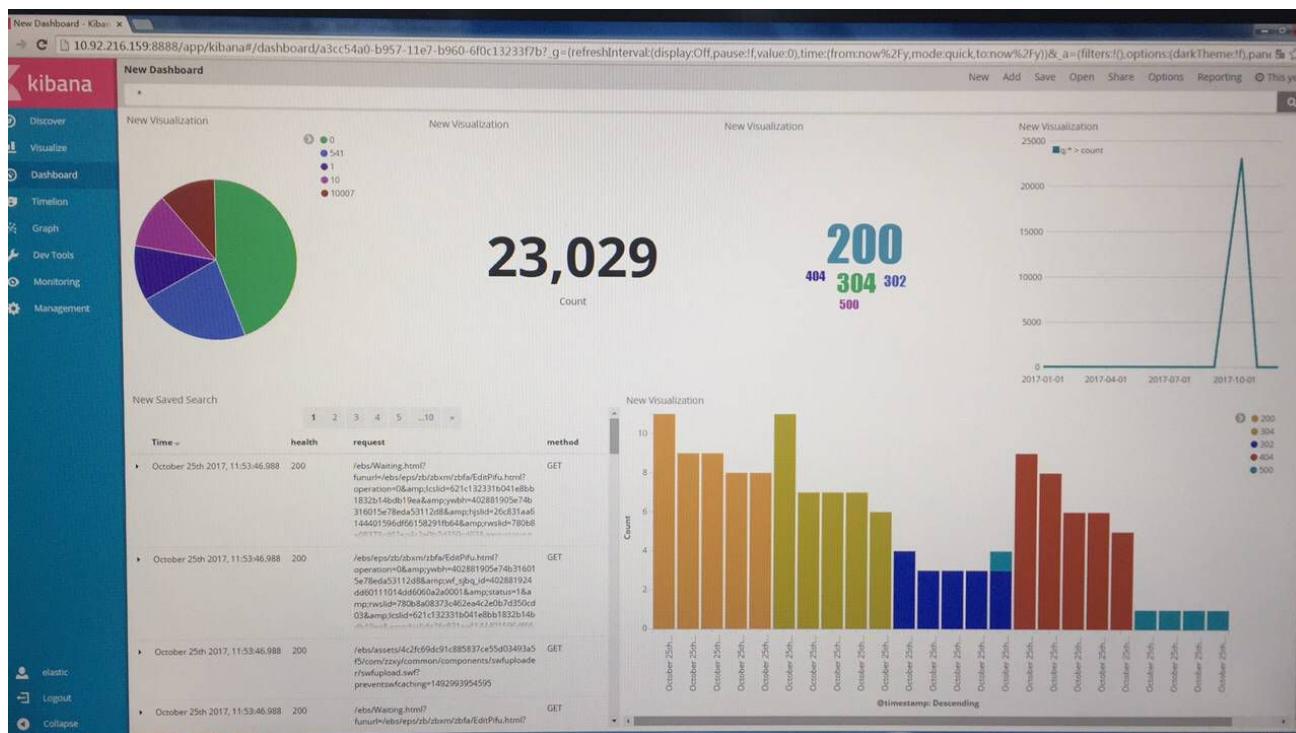


#### 第四节 选择 Monitoring 可对服务器运行状态、性能指标进行监控。





## 第五节 对数据进行自定义展示，效果如下图。（第五章详讲）



## 第四章 架构升级

Logstash to Elasticsearch to Kibana 架构局限于小型拓扑架构中，如生产环境拓扑架构复杂关联性高，日志处理量多，那么就需要对本文的架构进行升级，参考搭建思路：[Filebeat to Redis/Kafka to Logstash to Elasticsearch to Kibana](#)。

### 第一节 Filebeat

Logstash 致命的问题是它的性能以及资源消耗（默认的堆大小是 1GB）。尽管它的性能在近几年已经有很大提升，与它的替代者们相比还是要慢很多的，它在大数据量的情况下会是个问题。所以结合[电商系统](#)的实际数据量以及为了减少收集日志信息所带来的资源消耗影响业务正常运行，引入 [Filebeat](#) 轻量级日志传输工具。

### 4.1.1 优化后部署架构图



如图所示：应用将日志落地在本地文件，部署在每台服务器上的 Filebeat 负责收集日志，然后将日志发送给 Logstash；Logstash 将日志进行处理之后，比如 parse 等；然后将处理后的 Json 对象传递给 Elasticsearch，进行落地并进行索引处理；最后通过 Kibana 来提供 web 界面，来查看日志等。因为 ES 是基于 Lucene 的，所以 Kibana 支持 Lucene 查询语法。

### 4.1.2 安装并配置

```
[elk@filebeat elk201710]$ tar -zxvf filebeat-5.2.2-linux-x86_64.tar.gz
[elk@filebeat filebeat-5.2.2-linux-x86_64]$ mv filebeat.yml filebeat_01.yml
[elk@filebeat filebeat-5.2.2-linux-x86_64]$ vim filebeat.yml

filebeat.prospectors:
- input_type: log
  paths:
    - /elk201710/access/access.log*
  fields:
    log_source: accesslog
output.logstash:
  hosts: ["192.168.8.201:5044"]

filebeat.prospectors:
- input_type: log
  paths:
    - /elk201710/ebs/*.log*
  fields:
    log_source: ebslog
output.logstash:
  hosts: ["192.168.8.201:5045"]
multiline.pattern: ^\#\#\#\#<
multiline.negate: false
multiline.match: after
[elk@filebeat filebeat-5.2.2-linux-x86_64]$ ./filebeat
```

## 第五章 平台使用维护及数据分析

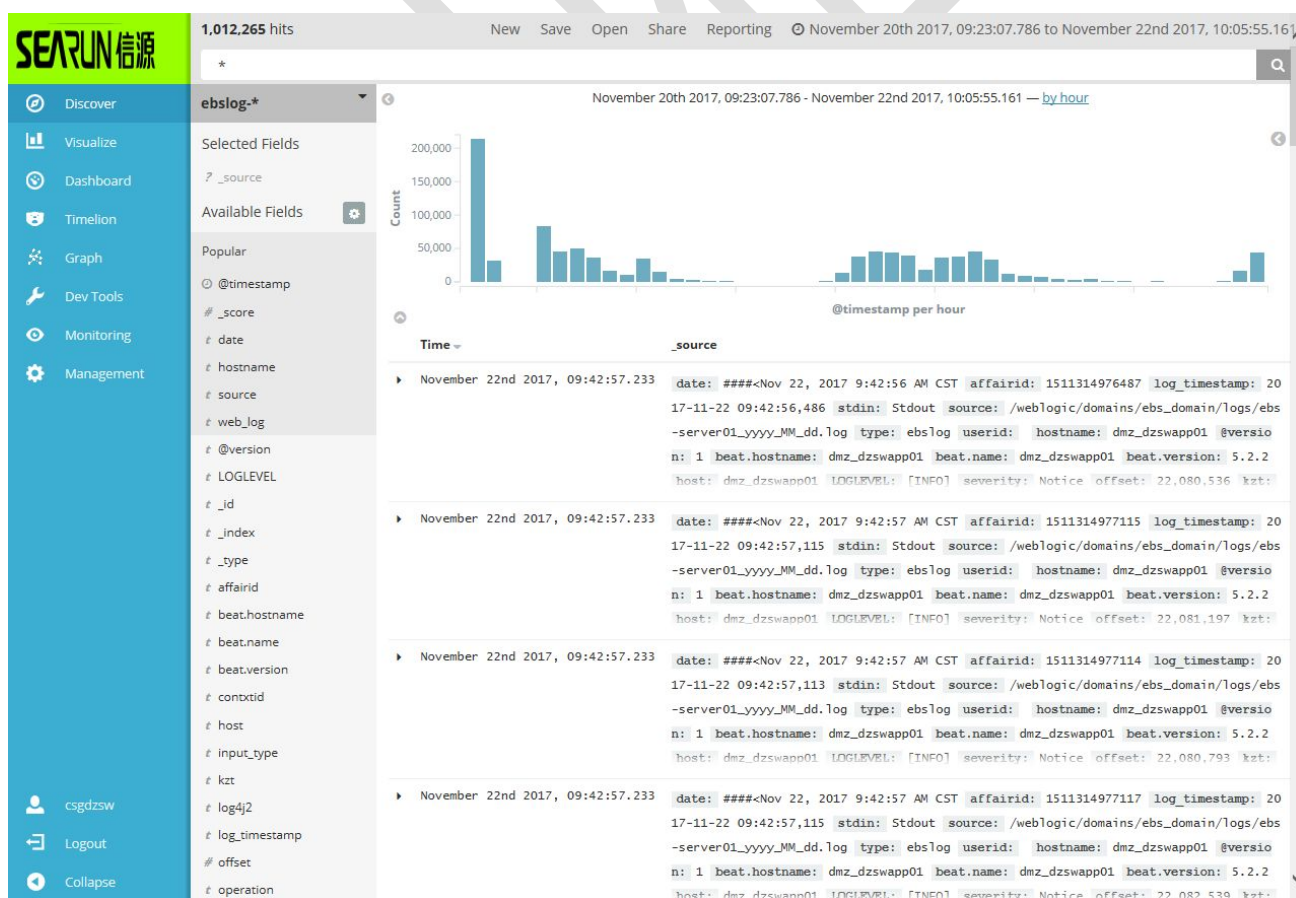
测试运行阶段在网公司电商测试环境下搭建了一套 ELK 平台，目的对电商测试环境的 weblogic 应用服务器日志进行了采集，及每日收集和监控平台的运行情况，为后续优化及上生产环境做准备。

### 第一节 电商 weblogic 应用数据采集

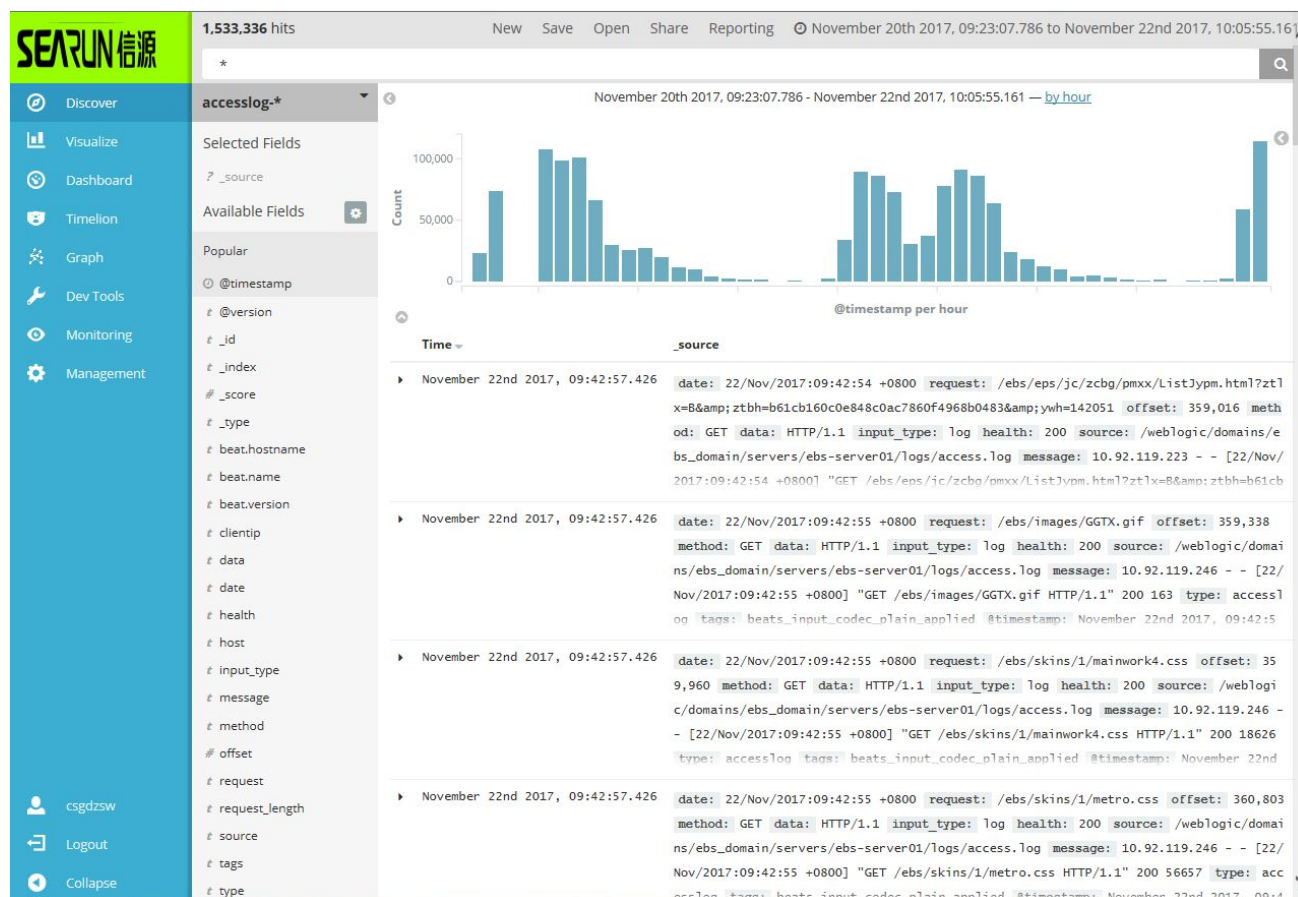
过程：对电商的应用日志 ebs-server0\*.log\* 及 http 服务日志 access.log\* 进行采集，输出至 kibana 展示。

### 第二节 电商 weblogic 应用数据展示

应用日志展示效果如下图：



http 服务日志展示效果如下图:



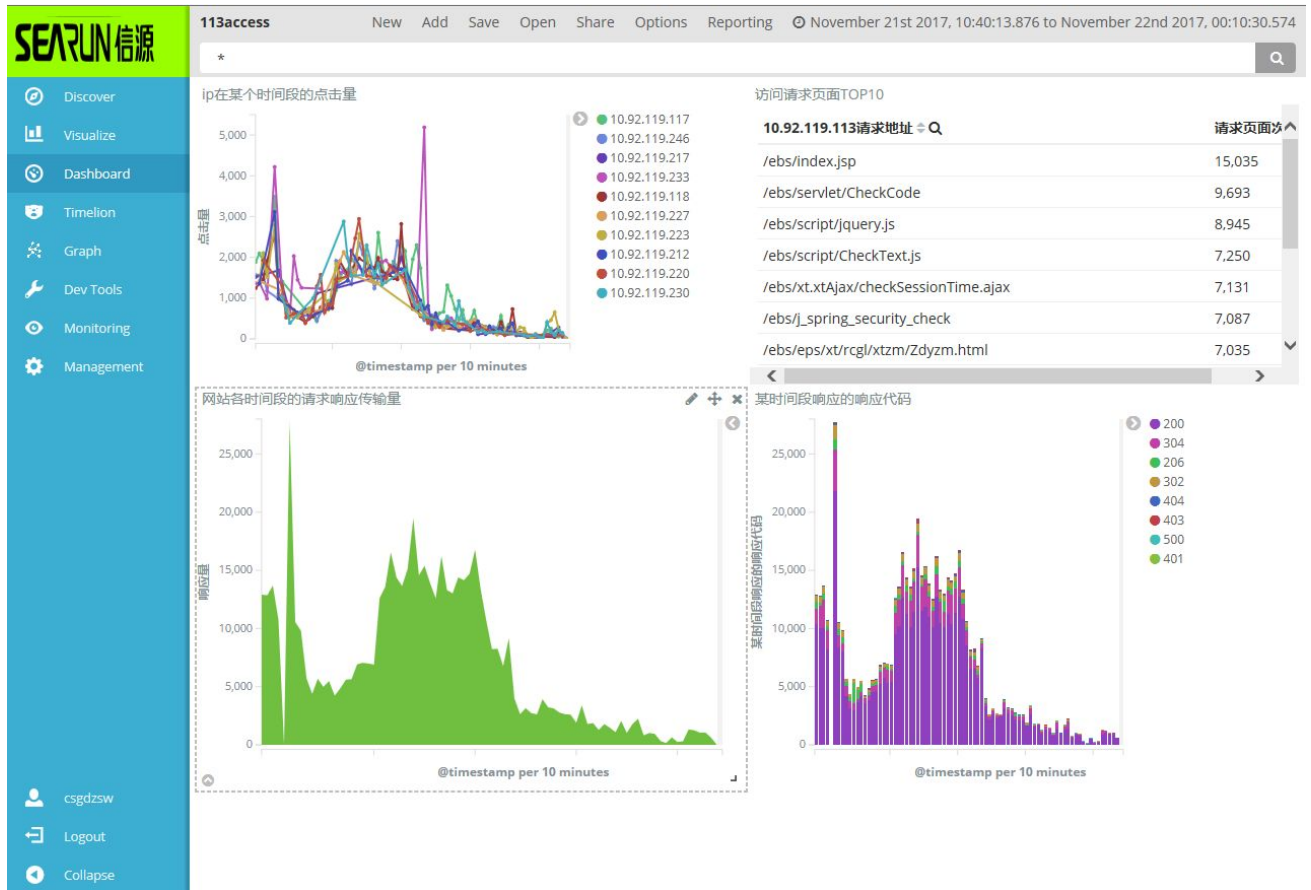
### 第三节 电商 weblogic 应用数据作图

应用日志作图效果如下图：

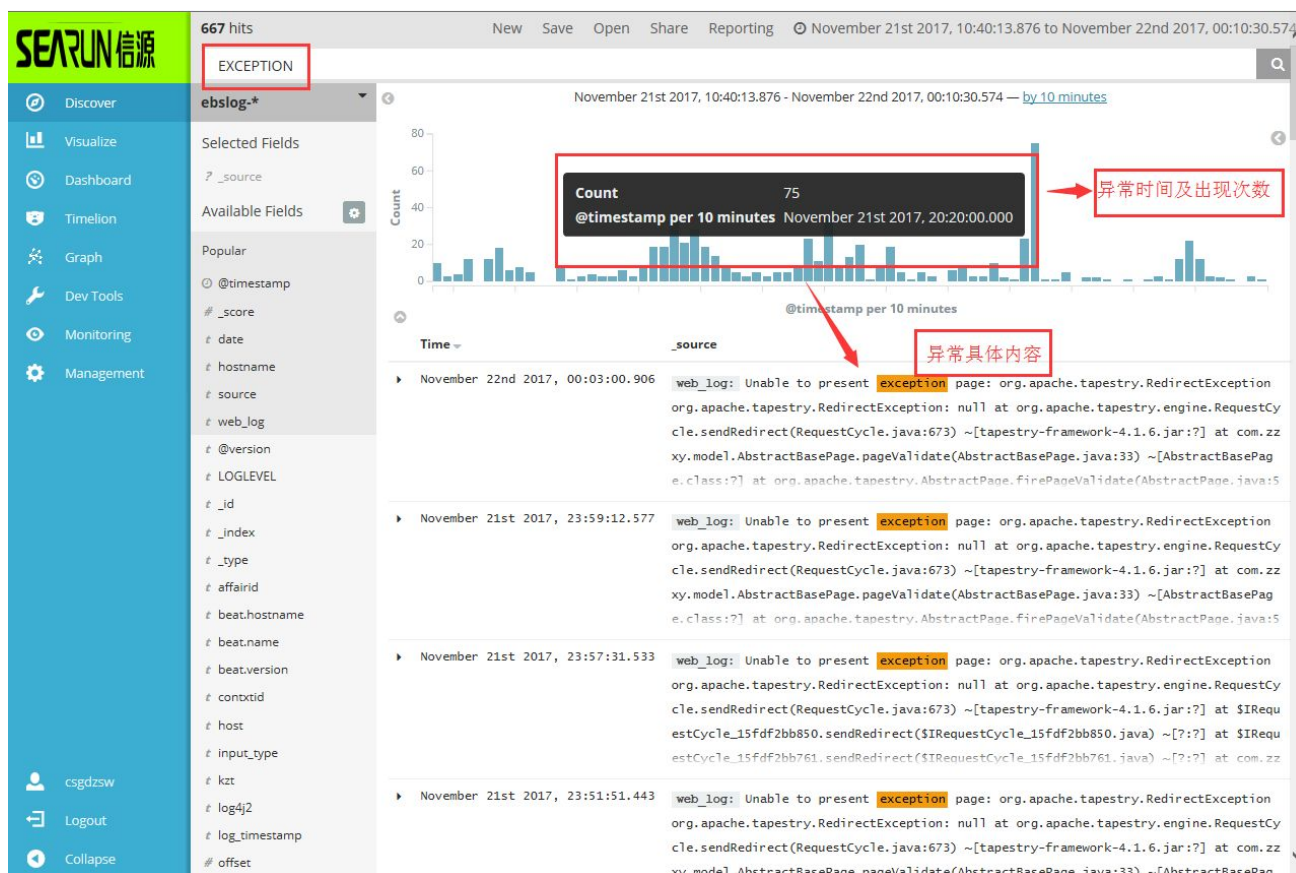




http 服务日志作图效果如下图:



## 第四节 电商 weblogic 应用数据搜索



## 第五节 平台日常维护

### 5.5.1 定期检查 ELK 服务器上磁盘使用情况

```
df -h
```

### 5.5.2 定期检查 JVM 的运行情况

```
jstat -gcutil pid
```

### 5.5.3 定期清理 ELK 服务器上缓存日志

esdelete.sh 脚本内容:

```
#!/bin/bash
#autor:mengxiaohao
#date:2017-11-28

curl -XDELETE http://192.168.8.201/ebslog-\* -uelastic
curl -XDELETE http://102.168.8.201/accesslog-\* -uelastic
```

建议加入定时任务，每一周清理一次。

```
*****/7 /home/elk/esdelete.sh
```

## 5.5.4 使用脚本一键操作停启服务

elkmxh.sh 脚本内容:

```
#!/bin/bash
#autor:mengxiaohao
#email:mangchiho777@aliyun.com
#date:2017-11-28
#This script is specially used to serve elk the log centralized collection platform.

Time=`date "+%Y-%m-%d %T"`
Date=`date "+%Y-%m-%d"`

if [ ! -n "$1" ];then
    echo "please enter a value of: start | stop | status"
elif [ "$1" == "start" ];then
    case $2 in
        k)
            nohup /elk201711/kibana/bin/kibana >> /elk201711/kibana/log/kibana_"$Date".log 2>/dev/null &
            echo -e "\033[32mkibana server is running!!!\033[0m"
            ;;
        e)
    
```



```

nohup /elk201711/es/bin/elasticsearch >> /elk201711/es/log/es_"$Date".log 2>/dev/null &
echo -e "\033[32melasticsearch server is running!!!\033[0m"
;;
l)
nohup /elk201711/logstash/bin/logstash -f /elk201711/config/log.conf >/dev/null 2>&1 &
echo -e "\033[32mlogstash server is running!!!\033[0m"
;;
*)
echo "please enter a value of: e | l | k"
;;
esac
elif [ "$1" == "stop" ];then
case $2 in
k)
number_k=`ps -ef | grep kibana | grep -v "grep kibana" | wc -l`
if [ $number_k == 0 ];then
echo -e "\033[36mhave no kibana process to kill!\033[0m"
else
ps -ef | grep kibana | grep -v "grep kibana" >/elk201711/kibana_pid.log
for i in `seq 1 $number_k`;do
kibana_pid=`cat /elk201711/kibana_pid.log | sed -n "$i"p' | awk '{print $2}`
kill -9 $kibana_pid
rm -f /elk201711/kibana_pid.log
echo -e "\033[31mkibana server has stopped!!!\033[0m"
done
fi
;;
e)
number_e=`ps -ef | grep elasticsearch | grep -v "grep elasticsearch" | wc -l`
if [ $number_e -eq 0 ];then
echo -e "\033[36mhave no es process to kill!\033[0m"
else
ps -ef | grep elasticsearch | grep -v "grep elasticsearch" >/elk201711/es_pid.log
for i in `seq 1 $number_e`;do
es_pid=`cat /elk201711/es_pid.log | sed -n "$i"p' | awk '{print $2}`
kill -9 $es_pid
rm -f /elk201711/es_pid.log
echo -e "\033[31melasticsearch server has stopped!!!\033[0m"
done
fi
;;
l)
number_l=`ps -ef | grep logstash | grep -v "grep logstash" | wc -l`
if [ $number_l -eq 0 ];then

```

```

        echo -e "\033[36mhave no logstash process to kill!\033[0m"
else
    ps -ef | grep logstash | grep -v "grep logstash" >/elk201711/logstash_pid.log
    for i in `seq 1 $number_l`;do
        logstash_pid=`cat /elk201711/logstash_pid.log | sed -n "$i'p' | awk '{print $2}'`
        kill -9 $logstash_pid
    rm -f /elk201711/logstash_pid.log
    echo -e "\033[31mlogstash server has stopped!!!\033[0m"
done
fi
;;
*)
    echo "please enter a value of: e | l | k"
;;
esac
elif [ "$1" == "status" ];then
case $2 in
k)
    number_k=`ps -ef | grep kibana | grep -v "grep kibana" | wc -l`
    if [ $number_k -eq 0 ];then
        echo -e "\033[31mkibana server is no running!\033[0m"
    else
        echo -e "\033[32mkibana server is running!\033[0m"
    fi
;;
e)
    number_e=`ps -ef | grep elasticsearch | grep -v "grep elasticsearch" | wc -l`
    if [ $number_e -eq 0 ];then
        echo -e "\033[31melasticsearch server is no running!\033[0m"
    else
        echo -e "\033[32melasticsearch server is running!\033[0m"
    fi
;;
l)
    number_l=`ps -ef | grep logstash | grep -v "grep logstash" | wc -l`
    if [ $number_l -eq 0 ];then
        echo -e "\033[31mlogstash server is no running!\033[0m"
    else
        echo -e "\033[32mlogstash server is running!\033[0m"
    fi
;;
*)
    echo "please enter a value of: e | l | k"
;;

```

```
esac
else
    echo "please enter a value of: start | stop | status"
fi
```

脚本用法截图：

```
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh
please enter a value of: start | stop | status
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh start
please enter a value of: e | l | k
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh start e
elasticsearch server is running!!!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh start l
logstash server is running!!!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh start k
kibana server is running!!!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh status e
elasticsearch server is running!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh status l
logstash server is running!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh status k
kibana server is running!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh stop e
elasticsearch server has stopped!!!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh stop l
logstash server has stopped!!!
[elk@US-ZABBIX elk201711]$ ./elkmxh.sh stop k
kibana server has stopped!!!
```

## 第六章 总结

经在网公司测试环境搭建平台后日常观察分析总结，得出以下几点个人看法。

- 1、如日后公司不满足与现在 kibana 所提供的服务，可针对性的对自己业务及需求，对 kibana 进行二次开发。
- 2、如何针对自己的业务及需求进行有效绘图。
- 3、建议单独申请服务器搭建日志展示平台。
- 4、根据业务量选择 ELK 架构，如果日志量大建议增加 redis 做缓存及 kafka 消息队列。