**什么是ELK**

**• ELK是一整套解决方案,是三个软件产品的首字母缩写,很多公司都在使用,如:Sina、携程、华为、美团等**

**• ELK分别代表**

**– Elasticsearch:负责日志检索和储存**

**– Logstash:负责日志的收集和分析、处理**

**– Kibana:负责日志的可视化**

**• 这三款软件都是开源软件,通常是配合使用,而且又先后归于Elastic.co公司名下,故被简称为ELK**

**ELK能做什么**

**• ELK组件在海量日志系统的运维中,可用于解决**

**– 分布式日志数据集中式查询和管理**

**– 系统监控,包含系统硬件和应用各个组件的监控**

**– 故障排查**

**– 安全信息和事件管理**

**– 报表功能**

**Linux apache mysql php**

**L A M P**

**L K E L**

**Kibana Elasticsearch logstash**

**• ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎,基于RESTful API的Web接口**

**• Elasticsearch是用Java开发的,并作为Apache许可条款下的开放源码发布,是当前流行的企业级搜索引擎。设计用于云计算中,能够达到实时搜索,稳定,可靠,快速,安装使用方便**

**Elasticsearch部分(续1)**

**• 主要特点**

**– 实时分析**

**– 分布式实时文件存储,并将每一个字段都编入索引**

**– 文档导向,所有的对象全部是文档**

**– 高可用性,易扩展,支持集群(Cluster)、分片和复制(Shards 和 Replicas)**

**– 接口友好,支持JSON**

**• ES没有什么**

**– Elasticsearch没有典型意义的事务**

**– Elasticsearch是一种面向文档的数据库**

**– Elasticsearch没有提供授权和认证特性**

**• 相关概念**

**– Node: 装有一个ES服务器的节点**

**– Cluster: 有多个Node组成的集群**

**– Document: 一个可被搜索的基础信息单元**

**– Index: 拥有相似特征的文档的集合**

**– Type: 一个索引中可以定义一种或多种类型**

**– Filed: 是ES的最小单位,相当于数据的某一列**

**– Shards: 索引的分片,每一个分片就是一个Shard**

**– Replicas: 索引的拷贝**

**• ES与关系型数据库的对比**

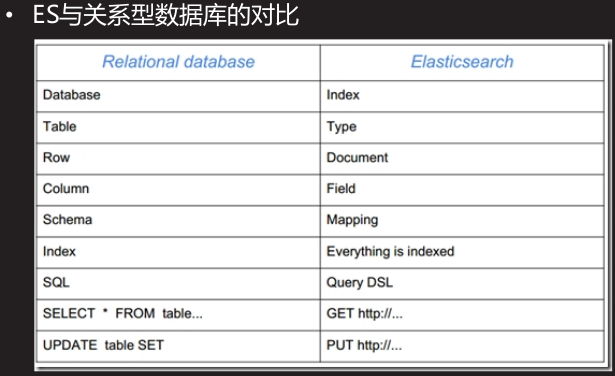
**– 在ES中,文档归属于一种 类型(type),而这些类型存在于索引(index)中,类比传统关系型数据库**

**– DB -> Databases -> Tables -> Rows -> Columns**

**– 关系型 数据库 表 行 列**

**– ES -> Indices -> Types -> Documents -> Fields**

**– ES 索引 类型 文档 域(字段)**

****

**ES集群安装**

**• 安装第一台ES服务器**

**– 设置主机名称和ip对应关系**

**– 解决依赖关系**

**– 安装软件包**

**– 修改配置文件**

**– 启动服务**

**– 检查服务**

**配置/etc/hosts**

**192.168.1.61 es1**

**192.168.1.62 es2**

**192.168.1.63 es3**

**192.168.1.64 es4**

**192.168.1.65 es5**

**yum install -y java-1.8.0-openjdk**

**rpm -ivh elasticsearch-2.3.4-1.noarch**

**修改配置文件**

**– elasticsearch.yml**

**network.host: 0.0.0.0 监听所有端口可以通过网络访问**

**Cluster.name: nasd1807 集群名**

**Node.name: es1# 本机名**

**Discovery.zen.ping.unicast.hosts:[“es1”,”es2”,”es3”] 集群的成员**

**验证 <http://192.168.1.61:9200/_cluster/health?pretty>**

**• ES集群配置**

**– ES集群配置也很简单,只需要对配置文件做少量的修**

**改即可,其他步骤和单机完全一致**

**– ES集群配置文件**

**cluster.name: my-es**

**node.name: node1**

**network.host: 0.0.0.0**

**discovery.zen.ping.unicast.hosts: ["node1", "node2",**

**"node3"]**

**• ES集群配置**

**– 集群中的所有节点要相互能够ping通,要在所有集群机器上配置/etc/hosts中的主机名与ip对应关系**

**– 集群中所有机器都要安装Java环境**

**– cluster.name集群名称配置要求完全一致**

**– node.name为当前节点标识,应配置本机的主机名**

**– discovery为集群节点机器,不需要全部配置**

**– 配置完成以后启动所有节点服务**

**ES集群配置**

**– 验证集群,使用ES内置字段 \_cluster/health**

**[root@room9pc01 elk]# curl http://192.168.1.61:9200/\_cluster/health?pretty**

**{**

**"cluster\_name" : "nsd1807",**

**"status" : "green",**

**"timed\_out" : false,**

**"number\_of\_nodes" : 4,**

**"number\_of\_data\_nodes" : 4,**

**"active\_primary\_shards" : 0,**

**"active\_shards" : 0,**

**"relocating\_shards" : 0,**

**"initializing\_shards" : 0,**

**"unassigned\_shards" : 0,**

**"delayed\_unassigned\_shards" : 0,**

**"number\_of\_pending\_tasks" : 0,**

**"number\_of\_in\_flight\_fetch" : 0,**

**"task\_max\_waiting\_in\_queue\_millis" : 0,**

**"active\_shards\_percent\_as\_number" : 100.0**

**}**

**• ES 集群验证**

**– 返回字段解析**

**– status " : " green " 集群状态,绿色为正常,黄色表示有问题但不是很严重,红色表示严重故障**

**– "number\_of\_nodes" : 5, 表示集群中节点的数量**

**[root@room9pc01 ~]# curl http://118.144.89.240/info.php -d oo.txt**

**<pre>**

**[ REQUEST\_METHOD] ==> POST**

**[ REMOTE\_ADDR] ==> 61.132.72.238**

**[HTTP\_USER\_AGENT] ==> curl/7.29.0**

**[ HTTP\_REFERER] ==>**

**--- --- POST options is --- ---**

**Array**

**(**

**[oo\_txt] =>**

**)**

**• http请求由三部分组成**

**– 分别是:请求行、消息报头、请求正文**

**– 请求行以一个方法符号开头,以空格分开,后面跟着**

**请求的URI和协议版本,格式如下:**

**Method Request-URI HTTP-Version CRLF**

**http请求方法**

**– 常用方法 GET,POST,HEAD**

**– 其他方法 OPTIONS,PUT,DELETE,TRACE和CONNECT**

**• ES 常用**

**– PUT --- 增**

**– DELETE --- 删**

**– POST --- 改**

**– GET --- 查**

**• 在linux中curl是一个利用URL规则在命令行下工作的文件传输工具,可以说是一款很强大的http命令行工具。它支持多种请求模式,自定义请求头等强大功能,是一款综合工具**

**• curl 常用参数介绍**

**– -A 修改请求 agent**

**– -X 设置请求方法**

**– -i 显示返回头信息**

**ES插件的使用**

**• ES常用插件**

**• head插件**

**– 它展现ES集群的拓扑结构,并且可以通过它来进行索引(Index)和节点(Node)级别的操作**

**– 它提供一组针对集群的查询API,并将结果以json和表格形式返回**

**– 它提供一些快捷菜单,用以展现集群的各种状态**

**• kopf插件**

**– 是一个ElasticSearch的管理工具**

**– 它提供了对ES集群操作的API**

**• bigdesk插件**

**– 是elasticsearch的一个集群监控工具**

**– 可以通过它来查看es集群的各种状态,如:cpu、内存使用情况,索引数据、搜索情况,http连接数等**

**• ES插件安装、查看**

**– 查看安装的插件**

**/usr/share/elasticsearch/bin/plugin list**

**– 安装插件**

**/usr/share/elasticsearch/bin/plugin install**

**ftp://192.168.4.254/head.zip**

**/usr/share/elasticsearch/bin/plugin install**

**file:///tmp/kopf.zip**

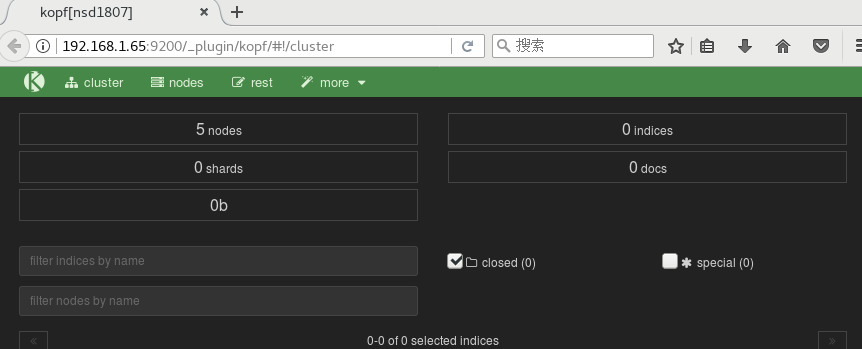
**– 这里必须使用 url 的方式进行安装,如果文件在本地,我们也需要使用 file:// 的方式指定路径,例如文件在/tmp/xxx下面,我们要写成 file:///tmp/xxx , 删除使用remove 指令**

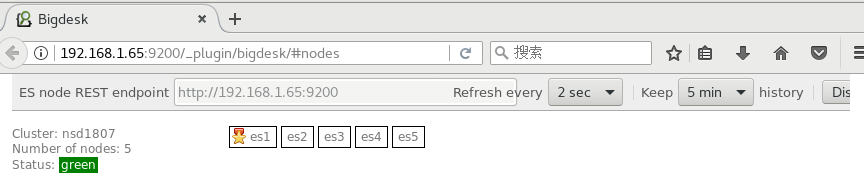
**[root@es5 bin] cd /usr/share/elasticsearch/bin**

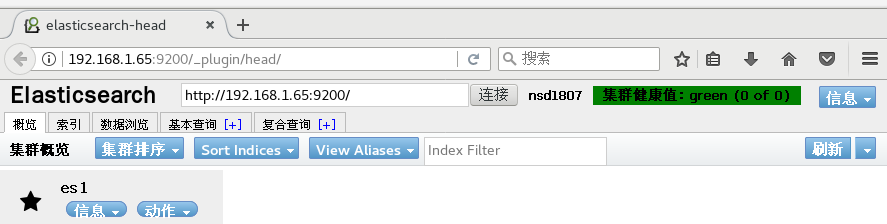
**[root@es5 bin]# ./plugin install ftp://192.168.1.254/elk/elasticsearch-head-master.zip**

**./plugin install ftp://192.168.1.254/elk/elasticsearch-kopf-master.zip**

**bigdesk-master.zip**

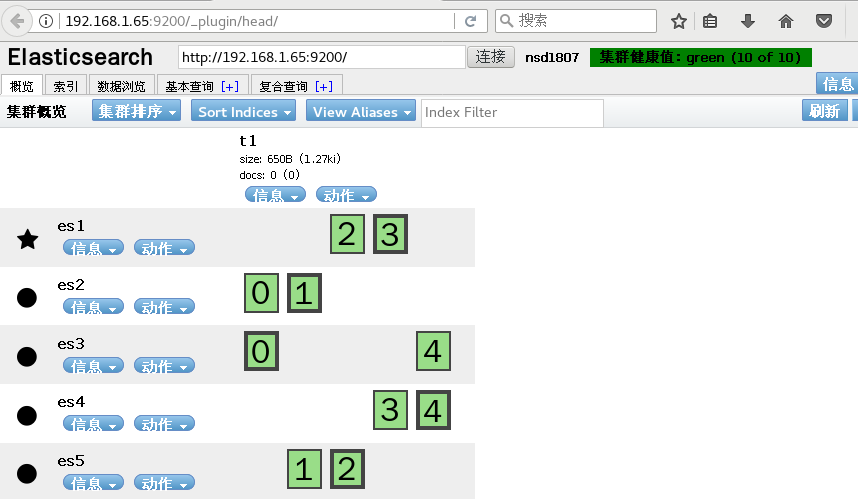
****

****

****

****

**分片5 是把数据分五分 写提升五倍 安全降低五倍**

****

**框浅的是源数据 深的是备份数据 列 2018-11-23 15-53-39 的屏幕截图**

**分片数量 和 副本数不能超过节点数量**

**RESTful API**

**• Elasticsearch提供了一系列RESTful的API**

**– 检查集群、节点、索引的健康度、状态和统计**

**– 管理集群、节点、索引的数据及元数据**

**– 对索引进行CRUD操作及查询操作**

**– 执行其他高级操作如分页、排序、过滤等**

**• POST或PUT数据使用json格式**

**• JSON**

**– JSON(JavaScript Object Notation),意思是JavaScript对象表示法,它是一种基于文本独立于语言的轻量级数据交换格式。**

**– JSON传输的就是一个字符串**

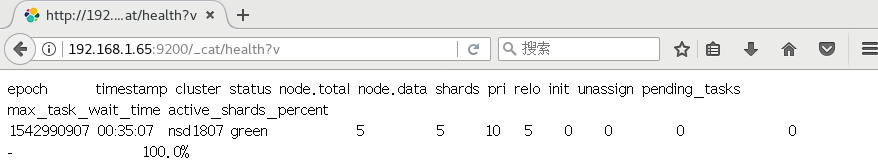
**– Python中对应的字符串,列表,字典都可以转换成对应的JSON格式**

**RESTful API的简单使用**

**– \_cat API查询集群状态,节点信息**

**– v参数显示详细信息**

**<http://192.168.1.65:9200/_cat/health?v>详细信息 ?help 帮助**

****

**– help显示帮助信息**

**http://192.168.1.65:9200/\_cat/health?help**

**创建索引**

**[root@es5 ~]# curl -XPOST http://192.168.1.61:9200/tedu/ -d'**

**{**

**"setting":{**

**"index":{**

**number\_of\_shards:5, 分片数量（不写数量 默认五片）**

**number\_of\_replicas:1 副本集数量**

**}**

**}**

**}'**

**{"acknowledged":true}**

**[root@es5 ~]#**

**增 PUT**

**[root@es5 ~]# curl -XPUT http://192.168.1.61:9200/tedu/nsd1807/1 -d'**

**> {**

**> "姓名":"孙傻逼",**

**> "性别":"女",**

**> "爱好":"碰瓷",**

**> "年龄":"35"**

**> }**

**> '**

**{"\_index":"tedu","\_type":"nsd1807","\_id":"1","\_version":1,"\_shards":{"total":2,"successful":2,"failed":0},"created":true}[root@es5 ~]#**

**改 POST**

**[root@es5 ~]# curl -XPOST http://192.168.1.61:9200/tedu/nsd1807/3/\_update -d'**

**{**

**"doc":{"年龄":18 }**

**}'**

**{"\_index":"tedu","\_type":"nsd1807","\_id":"3","\_version":2,"\_shards":{"total":2,"successful":2,"failed":0}}[root@es5 ~]#**

**查寻 XGET 可以省略 后面?pretty 顺序显示**

**[root@es5 ~]# curl -XGET http://192.168.1.61:9200/tedu/nsd1807/3**

**{"\_index":"tedu","\_type":"nsd1807","\_id":"3","\_version":2,"found":true,"\_source":{"姓名":"丁丁","性别":"男","爱好":"丝袜","年龄":18}}[root@es5 ~]#**

**[root@es5 ~]# curl -XGET http://192.168.1.61:9200/tedu/nsd1807/3?pretty**

**{**

**"\_index" : "tedu",**

**"\_type" : "nsd1807",**

**"\_id" : "3",**

**"\_version" : 2,**

**"found" : true,**

**"\_source" : {**

**"姓名" : "丁丁",**

**"性别" : "男",**

**"爱好" : "丝袜",**

**"年龄" : 18**

**}**

**}**

**删除**

**[root@es5 ~]# curl -XDELETE http://192.168.1.61:9200/tedu/nsd1807/4**

**{"found":true,"\_index":"tedu","\_type":"nsd1807","\_id":"4","\_version":2,"\_shards":{"total":2,"successful":2,"failed":0}}[root@es5 ~]#**

**Kibana安装与配置**

**• kibana是什么**

**– 数据可视化平台工具**

**• 特点:**

**– 灵活的分析和可视化平台**

**– 实时总结流量和数据的图表**

**– 为不同的用户显示直观的界面**

**– 即时分享和嵌入的仪表板**

**• kibana安装**

**– kibana 的安装非常简单,我们使用 rpm 方式安装**

**rpm –ivh kibana-4.5.2-1.x86\_64.rpm**

**– kibana 默认安装在 /opt/kibana 下面,配置文件在**

**/opt/kibana/config/kibana.yml**

**– 我们只需要修改少量的配置就可以启动**

**• kibana.yml的配置**

**– server.port: 5601**

**– server.host: "0.0.0.0"**

**– elasticsearch.url: "http://192.168.1.65:9200"**

**– kibana.index: ".kibana"**

**– kibana.defaultAppId: "discover" 默认首页**

**– elasticsearch.pingTimeout: 1500 请求返回时间**

**– elasticsearch.requestTimeout: 30000**

**– elasticsearch.startupTimeout: 5000**

**• kibana.yml的配置**

**– 除elasticsearch.url需要配置为我们ES集群的地址之外,其他保持默认值**

**– 设置开机启动**

**systemctl enable kibana**

**– 启动服务**

**systemctl start kibana**

**– web访问kibana**

**http://192.168.1.66:5601/**

**批量导入数据**

**• 使用\_bulk批量导入数据**

**– 批量导入数据使用POST方式,数据格式为json,url编码使用data-binary**

**– 导入含有index配置的json文件**

**# gzip –d logs.jsonl.gz**

**# gzip –d shakespeare.json.gz**

**# curl -XPOST 'http://192.168.1.65:9200/\_bulk' --data-binary**

**@logs.jsonl**

**# curl -XPOST 'http://192.168.1.65:9200/\_bulk' --data-binary**

**@shakespeare.json**

**– 导入没有index配置的json文件**

**– 我们需要在ur里面制定index和type**

**# gzip –d accounts.json.gz**

**# curl -XPOST**

**'http://192.168.1.65:9200/accounts/act/\_bulk?pretty' --data-**

**binary @accounts.json**

**• 数据批量查询使用GET**

**Curl -XGET <http://192.168.1.65:9200/_mget?pretty> -d’**

**{**

**“doc”:[**

**{**

**“index”:”shakespeare”,**

**“type”:”line”,**

**“id”:”2”**

**}**

**{**

**“index”:”shakespeare”,**

**“type”:”line”,**

**“id”:”2”**

**}**

**{**

**“index”:”shakespeare”,**

**“type”:”line”,**

**“id”:”2”**

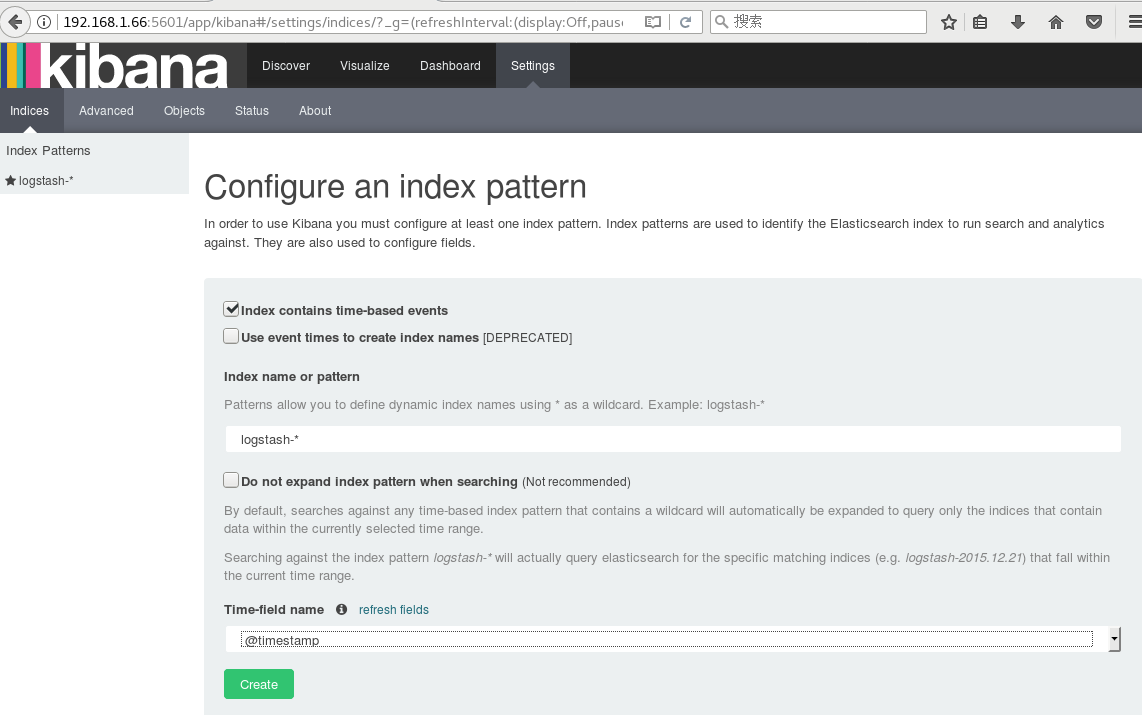
**}**

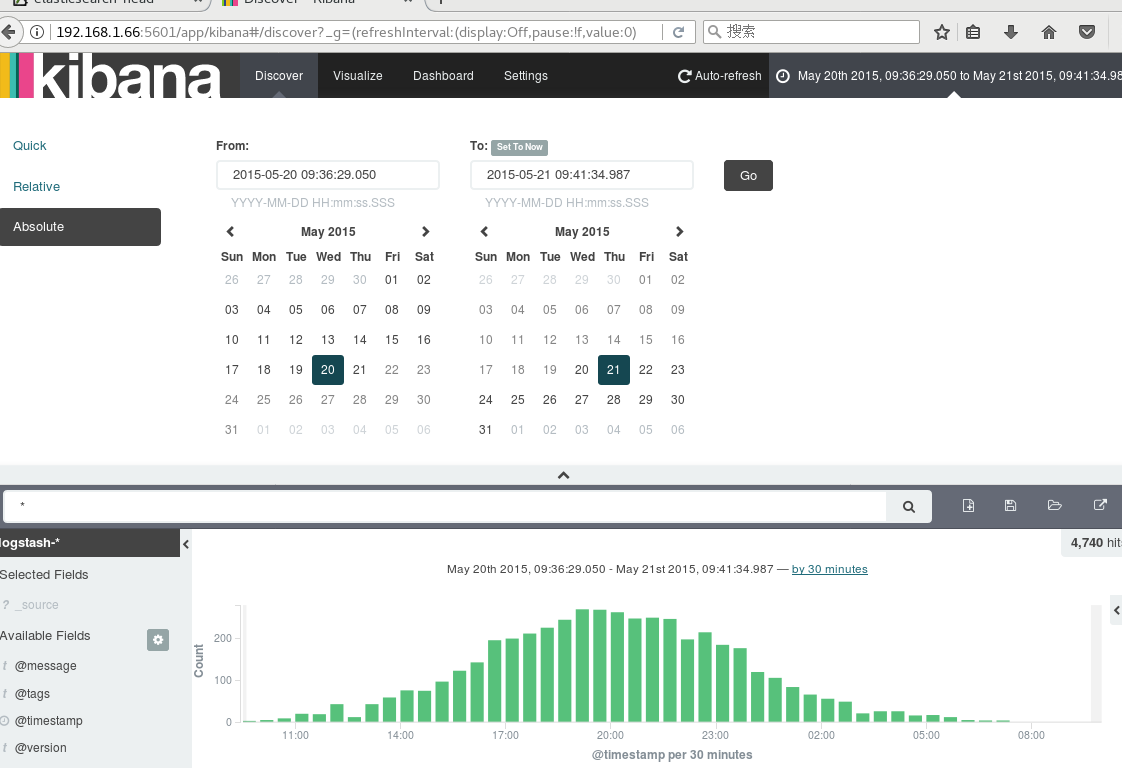
**]**

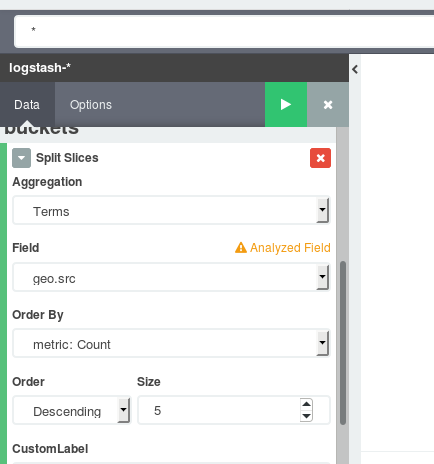
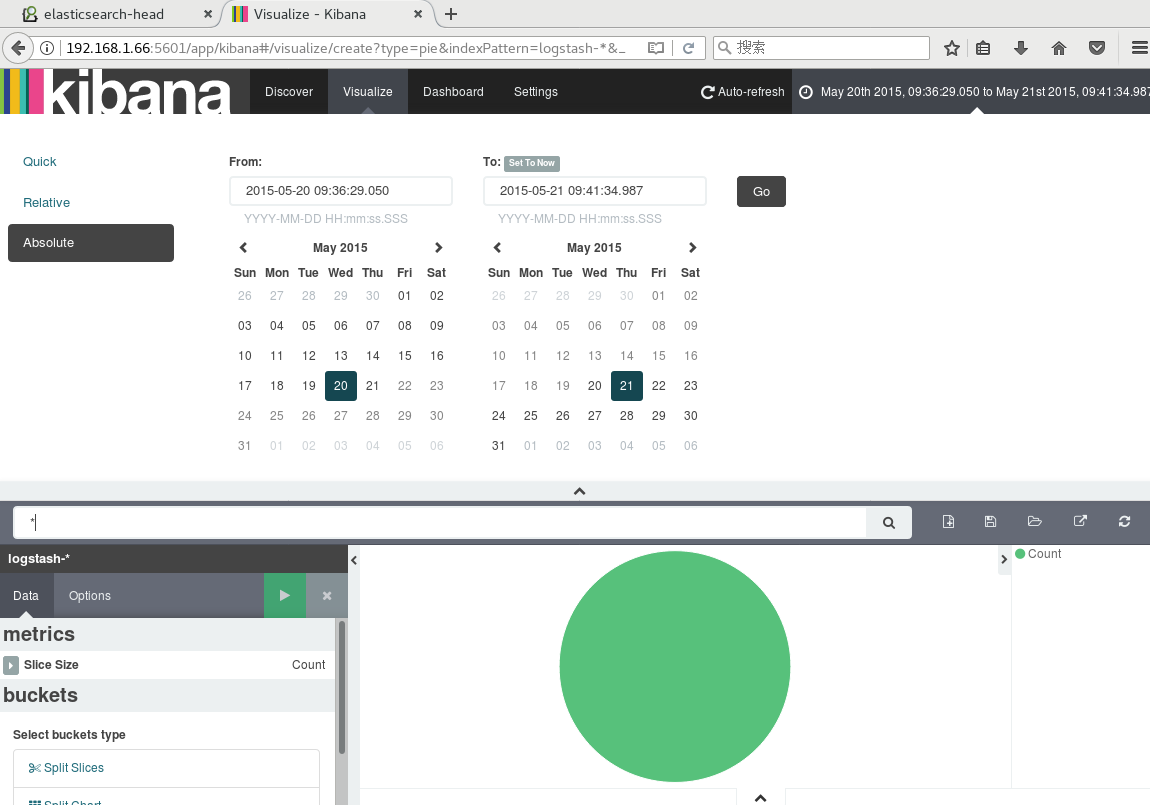
**}’**

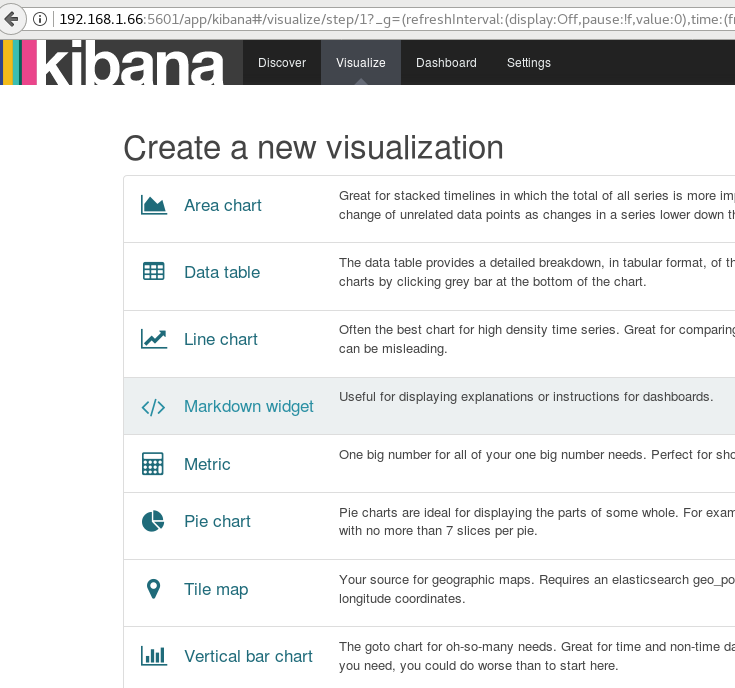
**– web访问kibana**

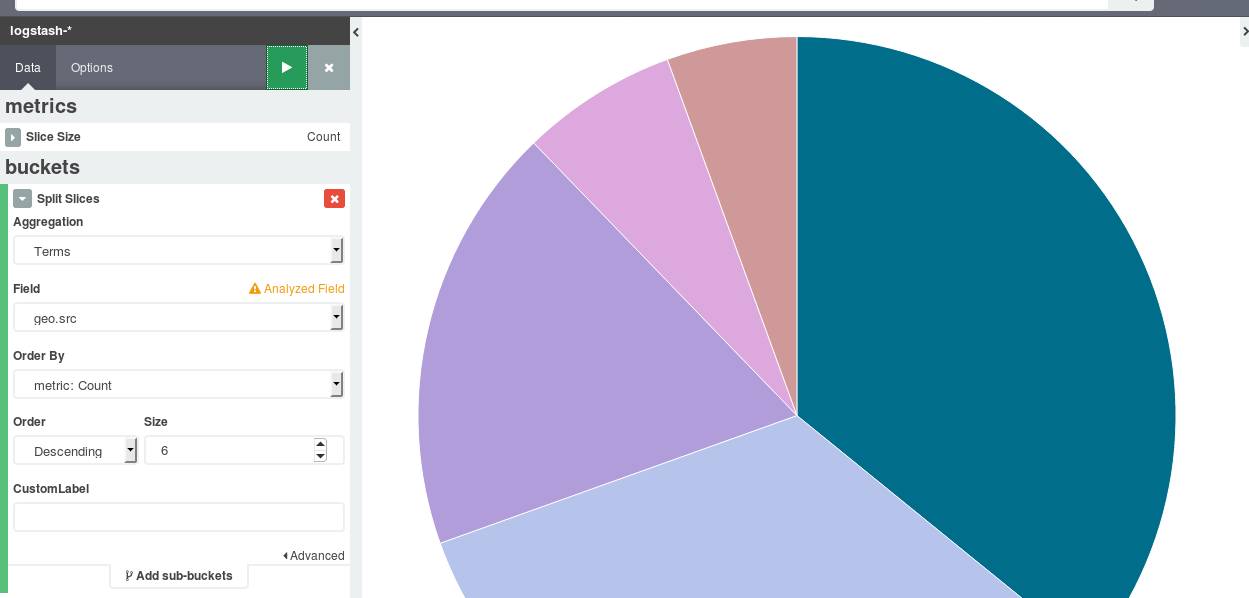
**http://192.168.1.66:5601/**

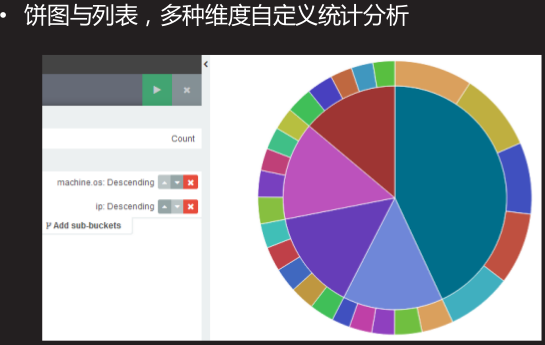
****

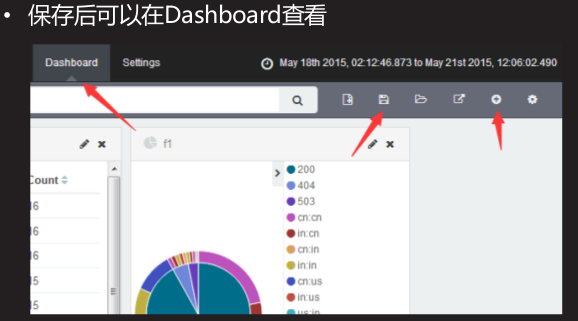
****

****

****

****

****

****