**15Architecture\_03分布式ELK平台+ES集群安装+扩展插件**

**一 分布式ELK平台-ELK简介**

**1.1 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

ELK分别代表[重要性依次降低]

Elasticsearch: 负责日志检索和存储[面向文档数据库,相当于LNMP的mariadb]

Logstash: 负责日志的收集和分析\处理[相当于LNMP的Python]

Kibana: 负责日志的可视化[提供可视网页服务,相当于LNMP的nginx]

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与LNMP作用类比

L N M P

K E L

**1.2 ELK能做什么**

ELK组件在海量日志系统的运维中,可以用于解决:

分布式日志数据集中式查询和管理

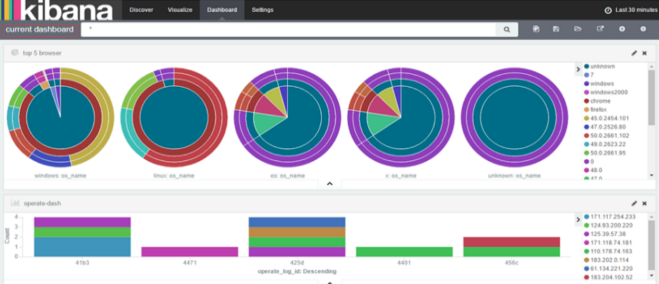
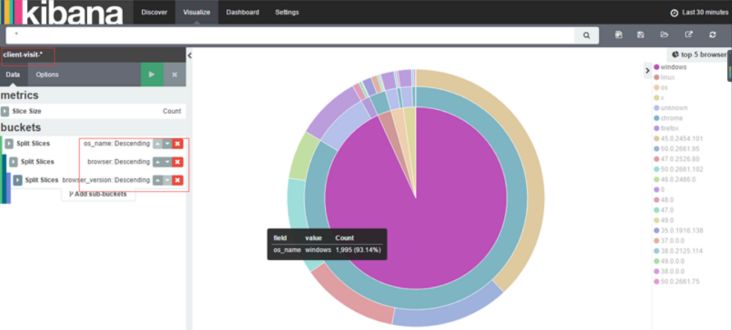
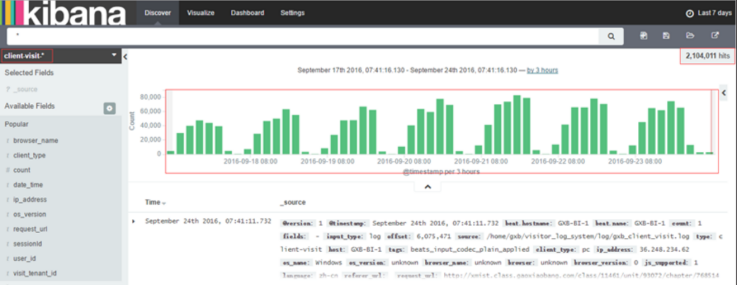
系统监控,包含系统硬件和应用各个组件的监控

故障排查

安全信息和事件管理

报表功能

**1.3 ELK什么样子**



**分布式ELK平台-Elasticsearch**

**2.1 Elasticsearch部分**

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.

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\分片和复制

接口友好,支持JSON

Elasticsearch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的服务器节点

Cluster: 由多个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

Peplicas: 索引的拷贝

**2.2 SQL与NoSQL**

ES于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,类比传统型关系型数据库

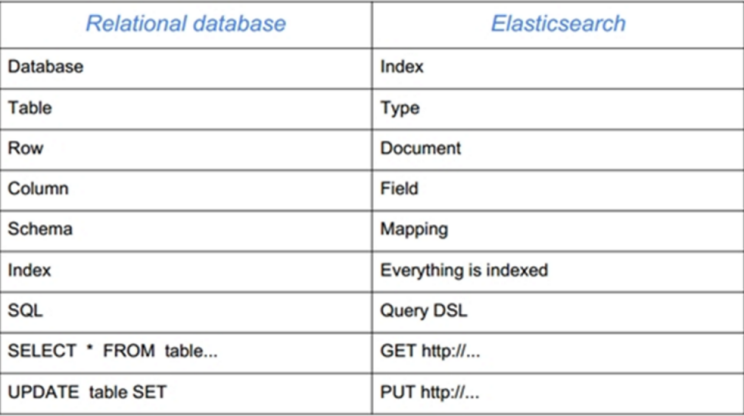
DB -> Databases -> Tables -> Rows -> Columns

关系性 数据库 表 行 列

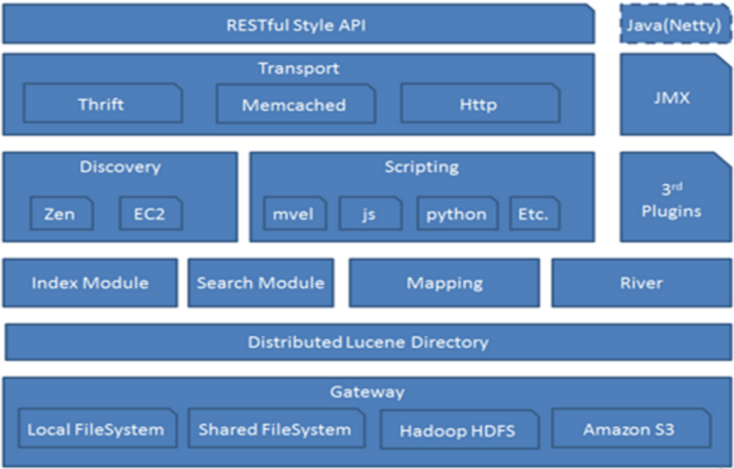
ES -> Indices -> Types -> Documents -> Fields

ES 索引 类型 文档 域(字段)

ES与关系型数据库的对比



**2.3 Elasticsearch架构图**



**三 ES 集群安装-ES基础**

**3.1 ES集群安装**

**3.1.1 安装第一台ES服务器**

room9pc01 ~]# base-vm es51

room9pc01 ~]# virsh start es51

room9pc01 ~]# virsh console es51

[root@localhost ~]# eip 51

[root@localhost ~]# hostnamectl set-hostname es51

[root@localhost ~]# reboot

room9pc01 ~]# ssh -X [root@192.168.1.51](mailto:root@192.168.1.51)

**3.1.2 设置主机名称和IP对应关系**

配置/etc/hosts: 192.168.1.51 es1

es51 ~]# vim /etc/hosts

192.168.1.51 es51

**3.1.3 解决依赖关系**

ES要求至少Java 7

一般推荐使用openJDK 1.8

配置号安装源后,我们先解决依赖关系

es51 ~]# yum -y install java-1.8.0-openjdk

**3.1.4 安装软件包**

room9pc01 ~]# cp -r /linux-soft/04/elk/\* /var/ftp/elk/

room9pc01 ~]# createrepo /var/ftp/elk

es51 ~]# vim /etc/yum.repos.d/local.repo

[elk]

name=elk

baseurl="ftp://192.168.1.254/elk"

enabled=1

gpgcheck=0

es51 ~]# yum -y reinstall elasticsearch

**3.1.5 修改配置文件**

es51 ~]# vim **/etc/elasticsearch/elasticsearch.yml**

54 nework.host:0.0.0.0 #解除注释并修改

**3.1.6 启动服务**

systemctl restart\enable elasticsearch

**3.1.7 检查服务**

es51 ~]# ss -antulp | grep 9200\9300

curl <http://192.168.1.51:9200/>

**3.2 ES集群配置**

**3.2.1 创建虚拟机**

room9pc01 ~]# base-vm es52 es53 es54 es55

**3.1.2 修改IP,hostname,修改yum源**

es52\53\54\55 ~]# yum -y install java-1.8.0-openjdk

es52\53\54\55 ~]# yum -y install elasticsearch

**3.1.3 修改配置文件/etc/elasticsearch/elasticsearch.yml**

es51\52\53\54\55 ~]# vim /etc/elasticsearch/elasticsearch.yml

17 cluster.name: nsd1906 #设置集群名称,注意空格

23 node.name: es51\es52\es53\es54\es55 #设置本机主机名,,注意空格

54 network.host: 0.0.0.0 #配置网络主机,注意空格

68 discovery.zen.ping.unicast.hosts: ["es51", "es52", "es53", "es54", "es55"] #配置集群节点主机名,注意空格

**3.1.4 启动服务,检查端口**

es51\52\53\54\55 ~]# systemctl restart elasticsearch

es51\52\53\54\55 ~]# systemctl enable elasticsearch

es51\52\53\54\55 ~]# ss -antulp | grep :9200

es51\52\53\54\55 ~]# ss -antulp | grep :9300

**3.1.5 配置/etc/hosts文件**

es51 ~]# vim /etc/hosts

192.168.1.51 es51

192.168.1.52 es52

192.168.1.53 es53

192.168.1.54 es54

192.168.1.55 es55

es51 ~]# for i in {52..55}

> do scp /etc/hosts root@192.168.1.$i:/etc/hosts

> done

**3.1.6 网页验证**

http://192.168.1.51:9200/\_cluster/health?pretty

{

"cluster\_name" : "nsd1906",

"status" : "green",

... ...

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

}

**重启所有主机时,需要先重启声明的节点主机,再启动其他节点主机**

**四 ES 集群安装-ES插件的使用**

**4.1 ES插件的使用**

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

可用通过它来查看ES集群的各种状态,如CPU\内存使用情况,索引数据,索引 情况,http连接数等

查看安装的插件:

/usr/share/elasticsearch/bin/plugin list **[install remove list]**

安装插件:

/usr/share/elasticsearch/bin/plugin install <ftp://192.168.1.254/head.zip>

/usr/share/elasticsearch/bin/plugin install [file:///tmp/kopf.zip](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,使用file://的方式指定路径

es51 ~]# cd /usr/share/elasticsearch/bin/

es51 bin]# ./plugin install <ftp://192.168.1.254/elk/elasticsearch-head-master.zip>

es51 bin]# ./plugin install <ftp://192.168.1.254/elk/elasticsearch-kopf-master.zip>

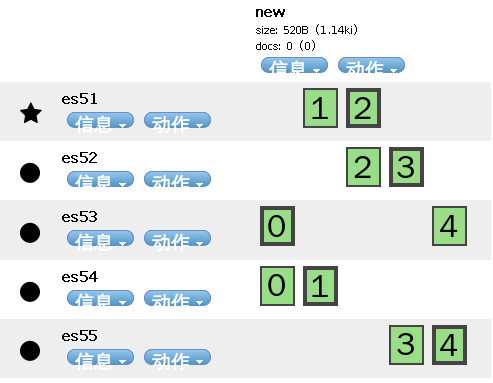
es51 bin]# ./plugin install <ftp://192.168.1.254/elk/bigdesk-master.zip>

**4.2 ES HEAD插件**

<http://192.168.1.51:9200/_plugin/head/>

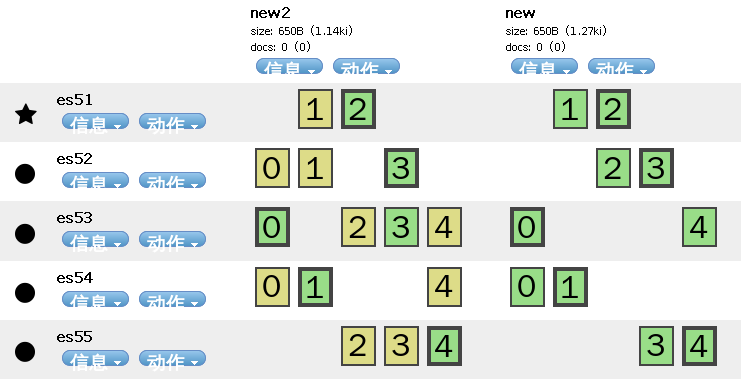
索引->新建索引->索引名称:new,分片数:5,副本数:1->OK

概览->查看new索引的数据分片信息,厚框为原始数据,薄框为副本数据



索引->新建索引->索引名称:new2,分片数:5,副本数:2->OK

概览->查看new索引的数据分片信息,厚框为原始数据,薄框为副本数据



动作->删除->输入”删除”->确定,删除索引

**4.3 ES KOPF插件**

<http://192.168.1.51:9200/_plugin/kopf>

**4.4 ES bigdesk插件**

<http://192.168.1.51:9200/_plugin/bigdesk/>

带勋章的为master主机,由系统自动选择,该主机down后,系统再次选择master主机

**五 RESTful API**

**5.1 http协议简介**

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,CONNECT

ES常用:

PUT->增, DELETE->删, POST->改, GET->查

**5.2 系统命令curl**

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,是一款很强大的http命令行工具.支持多种请求模式,自定义请求头等强大功能,是一款中和工具

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

es51 ~]# curl -XPOST -A "Windows IE/99.0" http://139.159.158.210/info.php

**5.3 RESTful API调用**

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

多索引进行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使用JSON格式

JSON

javascript object notation,JavaScript对象表示法.是一种基于文本独立于语言的轻量级数据交换格式.

JSON格式:

数组 [“aa”,”bb”,”cc”]

键值对 {key:value}

**5.4 RESTful API调用-简单使用**

\_cat API查询集群状态,节点信息

?V 显示详细信息

?help 显示帮助信息

es51 ~]# curl <http://192.168.1.51:9200/_cat/>

es51 ~]# curl <http://192.168.1.51:9200/_cat/shards?v>

es51 ~]# curl <http://192.168.1.51:9200/_cat/shards?help>

查询节点详细信息

es51 ~]# curl <http://192.168.1.51:9200/_cat/nodes?v>

查询节点索引信息

es51 ~]# curl <http://192.168.1.51:9200/_cat/indices?v>

**5.5 RESTful API增加**

创建一个索引tedu,并设置分片数量为5,副本数量1

es51 ~]# curl -XPUT 'http://192.168.1.51:9200/tedu1/' -d ‘

> { #-d表示数据,放在’’中

> "settings":{

> "index":{

> "number\_of\_shards":5,

> "number\_of\_replicas":2

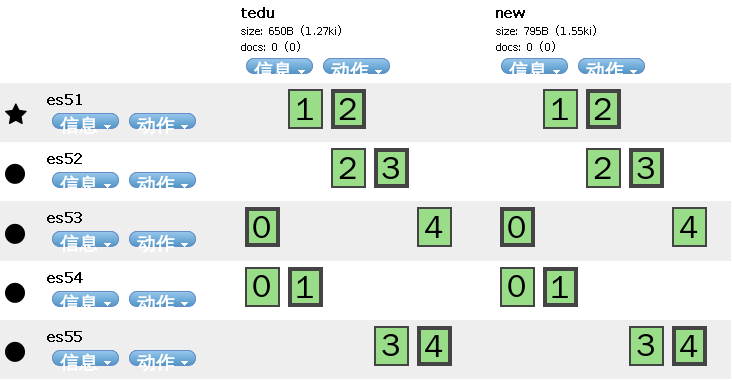
> }

> }

> }'

<http://192.168.1.51:9200/_plugin/head/>

概览



**5.6 RESTful API插入数据**

(增)加数据,使用PUT方法

调用方式: 数据库地址/索引/类型/ID值

es51 ~]# curl -XPUT 'http://192.168.1.51:9200/tedu/teacher/1' -d ‘

> {

> "职业":"诗人",

> "名字":"李白",

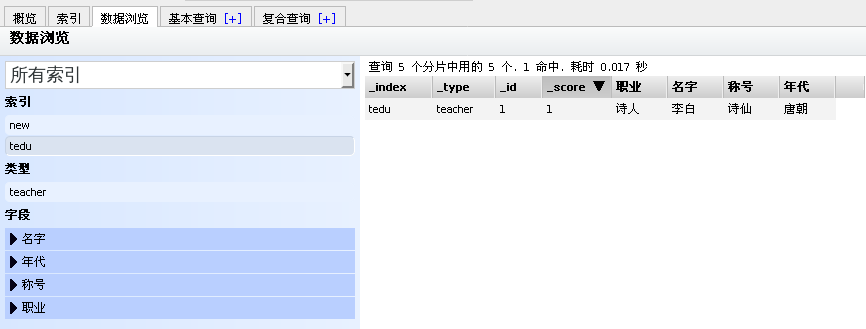
> "称号":"诗仙",

> "年代":"唐朝"

> }'

<http://192.168.1.51:9200/_plugin/head/>

数据浏览->tedu



**5.7 POST修改数据**

修改数据,使用POST方法

在修改数据的时候必须调用\_update关键字

调用方式: 数据库地址/索引/类型/ID值/\_update

es51 ~]# curl -XPOST \

> 'http://192.168.1.51:9200/tedu/teahcer/1/\_update' -d '

> {

> "doc":{

> "年代":"宋代"

> }

> }'

**5.8 (查)查询/(删)删除**

查询使用GET方法,GET为默认方法

查询显示结果时候可以用pretty规范显示格式

多条查询需要使用\_mget关键字配合JSON调用

es51 ~]# curl -XGET 'http://192.168.1.51:9200/tedu/teacher/1'

(删)删除

删除时可以是文档,也可以是库,但不能是类型

es51 ~]# curl -XDELETE 'http://192.168.1.51:9200/tedu/teacher/1'

数据浏览中1不见了

es51 ~]# curl -XDELETE 'http://192.168.1.51:9200/\*' #删根

**5.9 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

es51 ~]# cd elk

es51 elk]# gzip -d logs.jsonl.gz

es51 elk]# curl -XPOST \

> 'http://192.168.1.51:9200/\_bulk' --data-binary @logs.jsonl

es51 elk]# gzip -d accounts.json.gz

es51 elk]# curl -XPOST \

> 'http://192.168.1.51:9200/tedu/teacher/\_bulk'

> --data-binary @accounts.json

es51 elk]# gzip -d shakespeare.json.gz

es51 elk]# curl -XPOST \

> 'http://192.168.1.51:9200/\_bulk' \

> --data-binary @shakespeare.json

**5.10 map映射**

mapping

映射: 创建索引的时候,可以预先定义字段的类型及相关属性

作用: 这样会让索引建立得更加的细致和完善

分类: 静态映射和动态映射

动态映射: 自动根据数据进行相应的映射

静态映射: 自定义字段映射数据类型

**百度搜索金步国,翻译好的各种官方手册**

**www.jinbuguo.com**