JSON语句格式：{hash，集值对，对象} ， [数组]

------配置文件时yml -----输出格式：.json

ELK：类似于lamp，ELK是一系列的软件服务的组合。A--E，M---E，P---L

L A M P

K E L

Kebana(网页)------->Logstarch----->ELasticsearch（数据库）

ELK 其实幵丌是一款软件,而是一整套解决方案,是

三个软件产品的首字母缩写

– 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

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

Elasticsearch

用Java开发的,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（Nosql）

• 主要特点

– 实时分析

– 分布式实时文件存储,幵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

没有什么

-是一种面向文档的数据库。

没有提供授权和认证特性

没有典型意义的事务.

– 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)

在Elastic中，增删改查又http://

**安装elastic**

elasticsearch 单机安装

1 修改 /etc/hosts

2 安装 yum install -y java-1.8.0-openjdk

3 安装 yum install -y elasticsearch

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

network.host: 0.0.0.0

启动服务 systemctl start elasticsearch

浏览器访问验证 <http://192.168.1.11:9200/>

**搭建建集群：**

按照正常单机安装每一台机器

在配置文件/etc/elasticsearch/elasticsearch.yml中改为：

Es1为主节点：

cluster.name: my-es -----集群名字

node.name: es1 ------本机名字

network.host: 0.0.0.0 ----本机IP

discovery.zen.ping.unicast.hosts: ["es1", "es2","es3"](参与的节点)

Es2，3同理

集群名字要相同才能加入集群中来

cluster.name: my-es -----集群名字

node.name: es2 ------本机名字

network.host: 0.0.0.0 ----本机IP

discovery.zen.ping.unicast.hosts: ["es1", "es2","es3"](参与的节点)

依次重启机器，主节点先起来即可加入

(配置文件格式非常合理)各个节点都可以读写（自动转发给mater修改完后又返回给节点）

Elasticsearch的增删改查：

• 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 （在登陆时也使用post）

– GET --- 查

Curl 命令：

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,可以说是一款徆强大的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（全部都用大写）

– -A 修改请求 agent(useragent:显示客户端的user信息)

– -X 设置请求方法

– -i 显示返回头信息

在浏览器中主要使用get . post需要写一表单

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(ftp共享路径，也可以在github>上网络安装)

这里必须使用 url 的方式进行安装,如果文件在本地,也需要使用 file:// 的方式指定路径,例如文件在

/tmp/xxx 下面,要写成 file:///tmp/xxx 删除使用 remove 指令

在es1上安装插件

- bigdesk

- head

- kopf

访问插件 head

http://192.168.1.11:9200/\_plugin/head

访问插件 kopf

http://192.168.1.11:9200/\_plugin/kopf

访问插件 bigdesk

http://192.168.1.11:9200/\_plugin/bigdesk

分布式集群，优点：I/O强大，数据存储空间大

在head中可以新建索引：设置分片数，分片数越多，存储空间越小，东西越不一丢失。

拿空间，换安全。

**查询方式：**

1. 浏览器查询
2. curl 查询 curl <http://es1:9200/_cat/health?v>

– v 参数显示详细信息

– help 显示帮助信息

– nodes 查询节点状态信息 <http://192.168.4.15:9200/_cat/nodes?v>

– 索引信息 <http://192.168.4.15:9200/_cat/indices?v>

– 创建一个索引,幵设置分片数量不副本数量

curl -XPUT 'http://192.168.4.13:9200/tarena/' -d（发送一个数据json索引） '{

"settings":{

"index":{

"number\_of\_shards": 5,（分五个片）

"number\_of\_replicas": 1 (复制一份)

}

}

}'

**增加数据 PUT**

插入一条数据

curl -XPUT 'http://192.168.4.11:9200/tarena/teacher/1（id=1）' -d '{ （类型不需要提前声明）

"title": "阶段1",

"name":{"first": "小逗比", "last": "牛犇"},

"age": 25

}’

创建中注意：

\_index \_type \_id \_score title name.first name.last age

Index | type | id 必须要唯一才能创建，否则会进行覆盖之前相似的/ 类似于复合主键

更改数据 POST

curl -XPOST "http://192.168.1.15:9200/nsd1803/teacher/3/\_update" -d \

'{

"doc": { "age":18 }

}'

查询与删除数据

curl -XGET "http://192.168.1.14:9200/nsd1803/teacher/1?pretty"

curl -XDELETE "http://192.168.1.14:9200/nsd1803/teacher/1?pretty"

curl -XDELETE "http://192.168.1.14:9200/nsd1803/ 删除索引

curl -XDELETE "http://192.168.1.14:9200/删根。。

**Kibana数据可视化平台工具**

重要性：

E > L > K

Kibana（就是看一下日志，解耦性好，可以放入docker）

Kibana坏掉，日志无法看，可以修好就可以

Log坏掉，数据不更新了，修好然后可以通过日志补回来

Elasticsearch坏掉：数据丢失