ELK2

**kibana**

**数据的批量导入**

使用\_bulk批量导入：批量导入数据使用 POST 方式,数据格式为 json,url

编码使用 data-binary 小批量，小型软件 / url-percona

导入步骤：

将文件下载到本机------》在文件当前目录下Gzip -d 解压json.gz格式--------->库名。表名。数据格式------》curl -XPOST

<http://192.168.1.11/_bulk> --data-binary @accounts.json

注意：没有库和表时，数据无法导入，需要在导入时指定库名于表名，

curl -XPOST http://192.168.1.11/库名/表名/\_bulk --data-binary @jccounts.json/acconut/logs.jsonl

（没有库名表名指定库名表名，没有id系统随即生成一个导入，可以不加库名表名）

**数据批量查询使用 GET**

curl -XGET 'http://192.168.4.11:9200/\_mget?pretty' -d '{

"docs":[

{

"\_index": "accounts",

"\_type:": "act",

"\_id": 1

},

{

"\_index": "accounts",

"\_type:": "act",

"\_id": 2

}

{

"\_index": "shakespeare",

"\_type:": "scene",

"\_id": 1

}

]

}'

数据批量查询使用 GET• logstash 是什么

– logstash是一个数据采集、加工处理以及传输的工具

• logstash 特点:

– 所有类型的数据集中处理

– 丌同模式和格式数据的正常化

– 自定义日志格式的迅速扩展

– 为自定义数据源轻松添加插件

**Logstash:**

安装logstach----->新建一个虚拟机-------->安装java环境------->装logstach

• logstash 安装

– Logstash 依赖 java 环境,需要安装 java-1.8.0-

openjdk

– Logstash 没有默认的配置文件,需要手劢配置

– logstash 安装在 /opt/logstash 目录下

rpm -ivh logstash-2.3.4-1.noarch.rpm

#######################################

金步国作品集

Email(QQ)：70171448在QQ邮箱

学习并翻译中文手册------>

###########################################

Logstach的结构：

– { 数据源 } ==>

– input { } ==>(采购)

filter { } ==>（加工）

output { } ==>（处理）

{ ES }

---------------------------------------------------------------------------------------

logstash 里面的类型

– 布尔值类型: ssl\_enable => true

– 字节类型:

bytes => "1MiB"

– 字符串类型: name => "xkops"

– 数值类型: port => 22

– 数组: match => ["datetime","UNIX"]

– 哈希: options => {k => "v",k2 => "v2"}

– 编码解码: codec => "json"

– 路径: file\_path => "/tmp/filename"

– 注释: #

logstash 条件判断(同shell)

– 等于: ==

– 不等于: !=

– 小于: <

– 大于: >

– 小于等于: <=

– 大于等于: >=

– 匹配正则: =~

– 丌匹配正则: !~

logstash 条件判断

– 包含: in

– 丌包含: not in

– 不: and

– 戒: or

– 非不: nand

– 非戒: xor

– 复合表达式: ()

– 取反符合:!()

**logstash 的第一个配置文件**

------------------------------------------

**logstash 插件**

– 上页的配置文件使用了 logstash-input-stdin 和

logstash-output-stdout 两个插件,logstash 还有

filter 和 codec 类插件,查看插件的方式是

/opt/logstash/bin/logstash-plugin list

------------------------------------------------

– /etc/logstash/logstash.conf

input{

stdin{}

}

filter{ }

output{

stdout{}

}

• 启劢并验证

– logstash –f logstash.conf

或 /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf

在读取的时后，在/root/.sincedb\_db12342456的日志文件，作为历史记录进行读取。

file 模块

input{

stdin{ codec => "json" }

file {

id => "[ "/tmp/a.log","/etc/b.log" ]" ---->选择两个文件夹进行读取

sincedb\_path => "/var/lib/logstash/sincedb.log" ---->在指定的文件路径下读取（默认/root/,sincedb.....）

start\_position => "begining" ------->从开始的时后读取

Type => “filelog”

}

}

}

filter{}

output{

stdout{ codec => "rubydebug" }

}

#######################################

Tcp插件:

tcp {

port => 8888

host => "0.0.0.0"

mode => "server"

type => "tcplog"

}

两种模式：

第一种是server:监听端口，监听地址

第二种是client-----连接对方端口，对方的地址

UDP插件:

没有mode不分服务器于客户端

udp {

port => "9999"

}

测试相关连接

echo "test udp log" >/dev/udp/192.168.1.20/9999

echo "test tcp log" >/dev/tcp/192.168.1.20/8888

/opt/logstash/bin/logstash -f /etc/logstash/logstash.conf //开启监听

{

"message" => "test tcp log",

"@version" => "1",

"@timestamp" => "2018-07-30T08:46:23.980Z",

"host" => "192.168.1.5",

"port" => 52982, -----》客户端上的访问端口

"type" => "tcplog" -----》时别类型（由于访问的是8888端口故判断为tcplog）

}

{

"message" => "test udp log\n",

"@version" => "1",

"@timestamp" => "2018-07-30T08:47:18.732Z",

"host" => "192.168.1.5"

##########################################

Syslog系统日志插件

/etc/rsyslog.conf----日志配置文件

加入新的日志地址：

local0.info @@192.168.1.20:514

@@---TCP

@ ---udp

---------------------------------------------------------------------------------

在客户端当前目录下创建local0.info文件

重启rsyslog: systemctl restart rsyslog

logger -p local0.info -t "testlog" "hello world" "message" => "hello world\n",

查看监听：

"@version" => "1",

"@timestamp" => "2018-07-30T17:22:08.000Z",

"type" => "syslog",

"host" => "192.168.1.5",

"priority" => 134,

"timestamp" => "Jul 31 01:22:08",

"logsource" => "kibana",

"program" => "testlog",

"severity" => 6,

"facility" => 16,

"facility\_label" => "local0",

"severity\_label" => "Informational"

Gork插件

正则表达式分组匹配 (?<name>reg)

正则宏路径

/opt/logstash/vendor/bundle/jruby/1.9/gems/logstash-patterns-core-2.0.5/patterns

filter{

grok{

match => ["message", "%{COMBINEDAPACHELOG}"]

}

}

Grok插件是如何调试的，用那条命令可以进行监听显示，（出像未整理的数据）能不能在讲一下格式与内容。