**Morning：**

1、数据批量导入与查询

**-批量导入数据**

[root@kbn ~]# gzip -d accounts.json.gz logs.jsonl.gz

#解压gzip格式的压缩包

[root@kbn ~]# head -1 shakespeare.json

{"index":{"\_index":"shakespeare","\_type":"act","\_id":0}}

[root@kbn ~]# curl -XPOST http://192.168.1.11:9200/**\_bulk** \

> --data-binary @shakespeare.json

#\_bulk批量导入json数据，--data-binary转换为二进制进行POST传输

[root@kbn ~]# head -1 accounts.json

{"index":{"\_id":"1"}}

[root@kbn ~]# curl -XPOST http://192.168.1.11:9200/accounts/js/\_bulk\

> --data-binary @accounts.json

#导入时自定义索引名、类型名，默认分片5备份1

**-批量查询数据**

[root@kbn ~]# curl -XGET http://192.168.1.11:9200/**\_mget**?pretty -d '{

"docs": [

{

"\_index": "accounts",

"\_type": "js",

"\_id": 25

},

{

"\_index": "shakespeare",

"\_type": "line",

"\_id": 19

} ]

}'

#\_mget批量查询数据，?pretty结构化返回结果

**-map映射**

创建索引时，可以预先定义字段类型及相关属性。

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。

2、kibana的安装

**-准备工作**

虚拟机kibana：1.5G内存、2核CPU、20G硬盘，IP:192.168.1.10

设置/etc/hosts，可以ping通所有es

配置yum源

**-kibana的安装与配置**

[root@kibana ~]# yum -y install kibana

[root@kibana ~]# vim /opt/kibana/config/kibana.yml

server.port: 5601

server.host: "0.0.0.0"

elasticsearch.url: "http://es3:9200"

kibana.index: ".kibana" #默认索引名

kibana.defaultAppId: "discover" #kibana默认首页

elasticsearch.pingTimeout: 1500

elasticsearch.requestTimeout: 30000

elasticsearch.startupTimeout: 5000

#以上行去掉注释，修改ES.url

**-启动与验证**

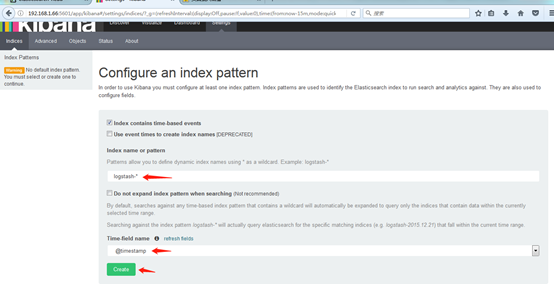
[root@kibana ~]# systemctl start kibana

[root@kibana ~]# firefox http://192.168.1.10:5601/status

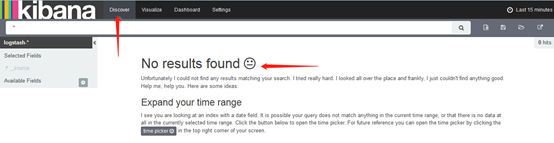
#Status：Green安装成功，默认创建.kibana索引(可通过ES.head查看)

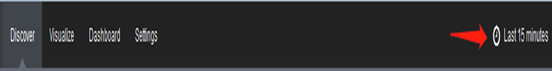
1. kibana的使用

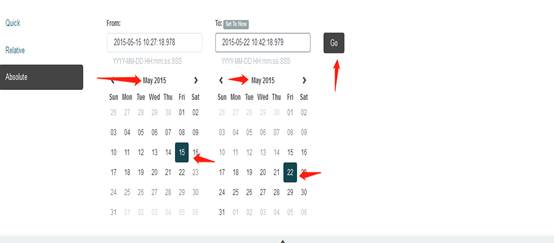
**步骤1：Setting导入索引**



**步骤2：Discover查看索引**



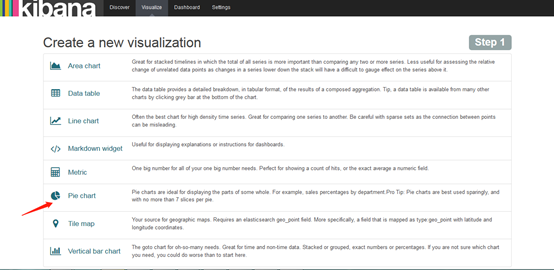




#由于Last 15 min没有数据，因此显示no found!

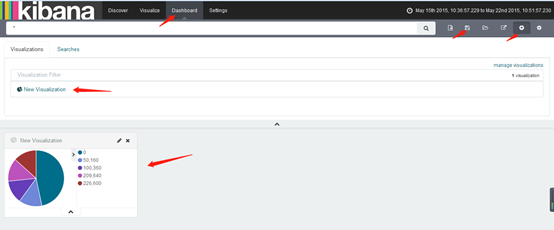
#可以选择数据的时间范围

**步骤3：Visualize制作图形**



#完成图形后，右上方save virsualization保存图形

**步骤4：Dashboard展示图形**



**Afternoon：**

1、Logstash介绍

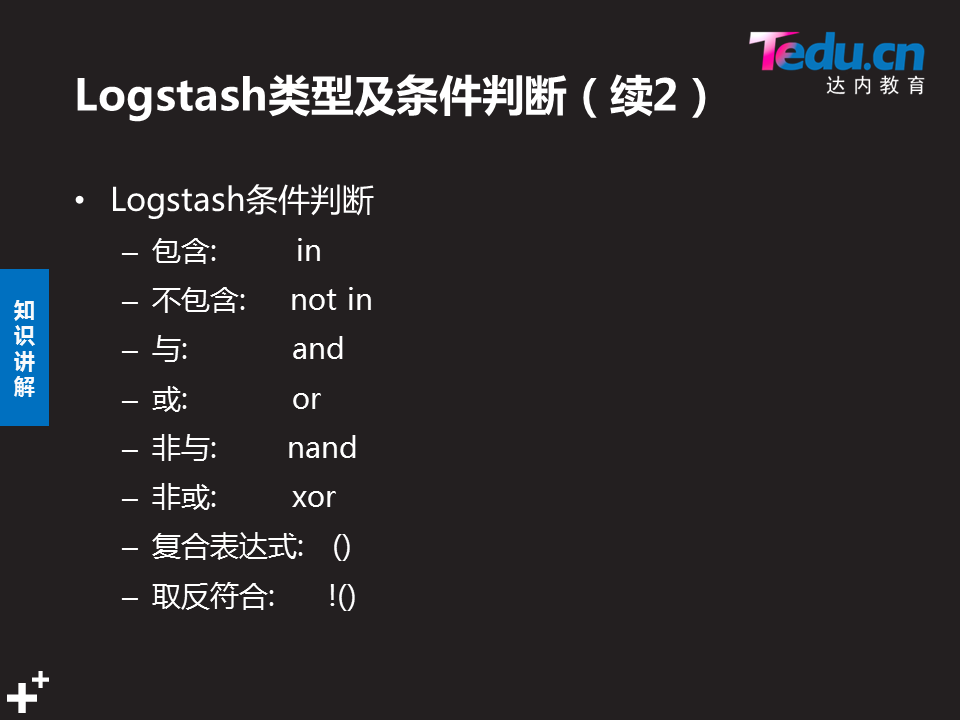
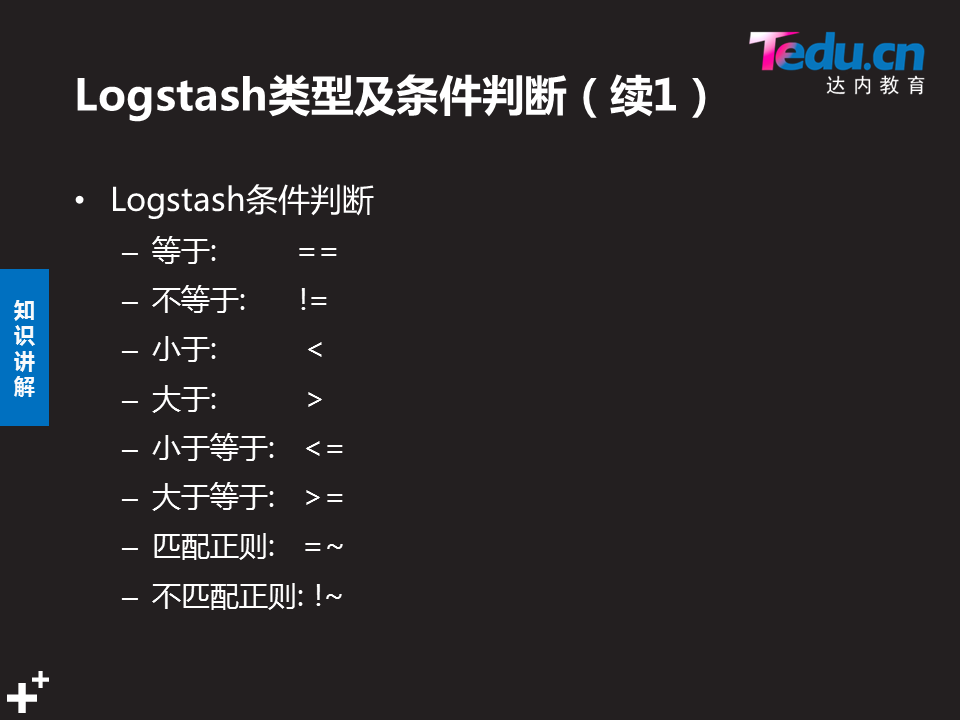
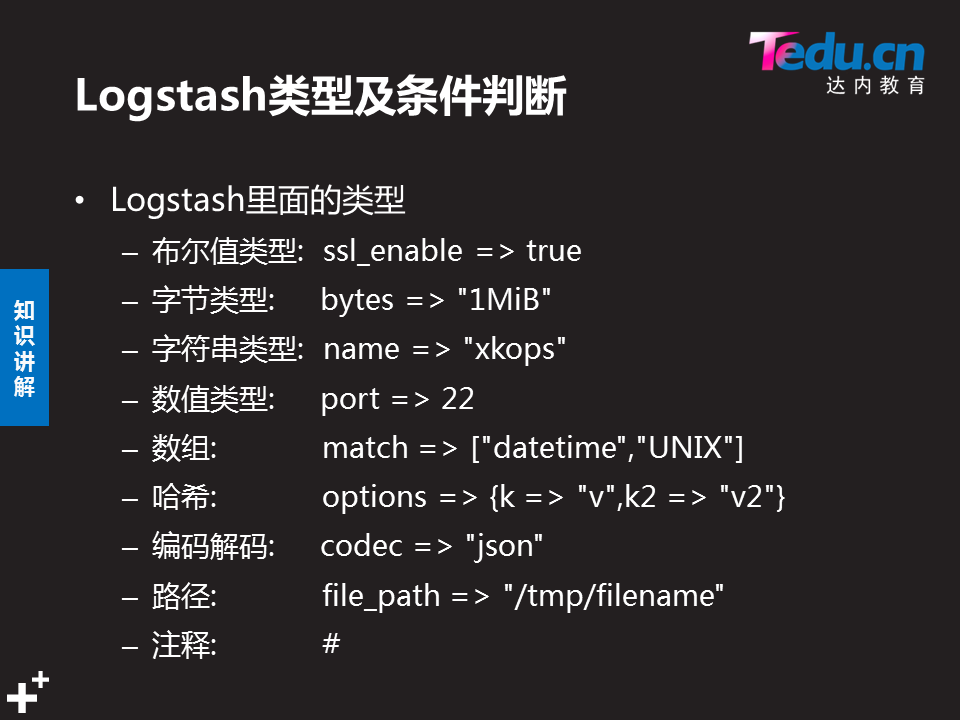
**-Logstash工作结构**

一个数据采集、加工处理及传输的工具。

Input{ } ==> filter{ } ==> output{ }

数据采集 加工处理 输出到ES

**-类型及条件判断**



2、Logstash安装与配置

**步骤1：准备工作**

1）虚拟机web：

-1.5G内存、2核CPU、20G硬盘、IP 192.168.1.100

-安装Apache

1. 虚拟机logstash：

-4G内存、2核CPU、20G硬盘、IP 192.168.1.20

-配置/etc/hosts(与es1~es5通讯)

-搭建yum源(es.repo)、安装java-1.8.0-openjdk、logstash

**步骤2：Logstash配置文件模板**

[root@logst ~]# vim /etc/logstash/logstash.conf #自定义配置文件

input{

stdin{}

}

filter{}

output{

stdout{}

}

[root@logst ~]# cd /opt/logstash/bin/

[root@logst ~]# ./logstash -f /etc/logstash/logstash.conf #启动

3、Logstash插件的使用

帮助文档：https://www.elastic.co/guide/en/logstash/current/index.html

[root@logst bin]# ./logstash-plugin list #查看所有插件

**-codec类插件**

1）json、rubydebug插件

[root@logst ~]# vim /etc/logstash/logstash.conf

input{

stdin{ codec => "json" } #使用json编码输入

}

filter{}

output{

stdout{ codec => "rubydebug" } #调试输出(易读)

}

**-input类插件**

1）file插件

[root@logst ~]# vim /etc/logstash/logstash.conf

input{

file{

path => ["/tmp/a.log", "/tmp/b.log"]

#导入文件位置

sincedb\_path => “/var/lib/logstash/since.db”

#自定义指针文件，默认为/root/.sincedbxxx

start\_position => “beginning” 或 “end”

#指针文件不存在时执行，从头读取，并重新生成指针文件

type => “test log” }

#给input的数据打标签

}

filter{}

output{

stdout{ codec => "rubydebug" }

}

[root@logst bin]# ./logstash -f /etc/logstash/logstash.conf

[root@logst ~]# echo A${RANDOM} >> /tmp/a.log

#验证本地数据采集

1. tcp/udp插件

[root@logst ~]# vim /etc/logstash/logstash.conf

input{

tcp {

mode => "server" 或

host => “0.0.0.0”

port => 8888

type => "tcp log" }

#作为server监听本机TCP 0.0.0.0:8888端口收到的数据

#mode、host有默认值，为了直观最好配置上去

udp {

port => 8888

type => "udp log" }

#默认host=0.0.0.0，监听本机UDP 0.0.0.0:8888端口收到的数据

... ...

文件描述符(文件句柄)：

-标准输入0、标准输出1、错误输出2

-/proc/$pid/fd，记录各个进程的文件描述符

-/proc/self/，记录进程本身的系统信息

-在当前进程定义文件描述符8：

exec 8<>/dev/tcp/192.168.1.20/8888

-关闭当前进程的文件描述符8：exec 8<&-

Bash的网络重定向：

-查看man bash帮助，/dev/tcp/host/port、/dev/udp/host/port

-网络重定向：echo abcd >/dev/tcp/192.168.1.20/8888

-使用文件描述符网络重定向：echo abcd >&8

[root@logst bin]# ./logstash -f /etc/logstash/logstash.conf

#开始监听0.0.0.0:8888端口

[root@kbn ~]# function send() {

> exec 8<>/dev/tcp/192.168.1.20/8888

> echo $1 >&8

> exec 8<&-

> }

[root@kbn ~]# send 123456

#验证网络数据采集

【拓展】自定义baidu:80文件句柄，获取页面内容

[root@kbn ~]# fun\_baidu() {

exec 99<>/dev/tcp/www.baidu.com/80

cat >&99 <<EOF

GET / HTTP/1.1 #此处请求头信息可通过curl -v获取

User-Agent: curl/7.29.0

Host: www.baidu.com

#空行\n表示请求头结束

EOF

cat >&1 <&99

exec 99<&-

}

[root@kbn ~]# fun\_baidu

1. syslog插件练习

[root@logst ~]# vim /etc/logstash/logstash.conf

input{

syslog {

port => 514

type => "syslog" }

#默认监听本机0.0.0.0:512端口(both TCP and UDP)收到的日志

... ...

验证：网络日志收集

[root@logst bin]# ./logstash -f /etc/logstash/logstash.conf

#开始监听0.0.0.0:514端口

[root@kb ~]# vim /etc/rsyslog.conf

authpriv.\* @192.168.1.20:514 #登陆安全信息日志

[root@kbn ~]# systemctl restart rsyslog

[root@kbn ~]# exit #断开ssh连接，查看Logstash信息

**-filter类插件**

1. grok插件练习

grok插件：使用正则把非结构化的数据结构化，几乎可应用于各类数据

**步骤1：**通读日志文件的字段含义

帮助文档：金步国作品集(各类软件中文手册)http://www.jinbuguo.com/

**步骤2：**将测试日志字段导入/tmp/a.log

测试技巧：sincedb\_path => "/dev/null"

**步骤3：**Logstash添加grok配置

[root@logst ~]# vim /etc/logstash/logstash.conf

input{

file{

path => "/tmp/a.log"

sincedb\_path => “/dev/null”

start\_position => “beginning”

type => “test log” }

}

filter{

grok{

match => [“message”, “(?<key1>reg1) (?<key2>reg2)”]

**#(1)自定义正则：**()组合、?固定、<key>组合名，reg正则

match => [“message”, “%{IP:key1} %{USER:key2}”]

**#(2)使用正则宏：**%{宏名:自定义名}

match => [“message”, “%{COMBINEDAPACHELOG}”]

**#(3)使用模块宏：**%{模块名}

}

output{

stdout{ codec => "rubydebug" }

}

Tips：

自定义正则：

match => ["message", "(?<ip>([12]?\d{1,2}\.){3}([12]?\d{1,2})) (?<identuser>\S+) (?<User>\S+) (?<time>\[.+\]) (?<head>\".+\") (?<return>\d+) (?<bytes>\d+) (?<ref>\"\S+\") (?<agent>\S+)"]

宏定义的文件位置：/opt/logstash/vendor/bundle/jruby/1.9/gems/logstash-patterns-core-2.0.5/patterns/grok-patterns

**步骤4：**验证grok处理结果

[root@logst bin]# ./logstash -f /etc/logstash/logstash.conf

"message" => “xxxx”

... ...

"key1" => "192.168.1.254",

“key2" => "-"

**-output类插件**

1. elasticsearch插件练习

[root@logst ~]# vim /etc/logstash/logstash.conf

input{... ...}

filter{... ...}

output{

if [type] == "testlog"{ #根据日志标签过滤输出

elasticsearch {

hosts => ["es1:9200", “es2:9200",”es3:9200”...]

#ES服务器地址:端口

index => "weblog-%{+YYYY.MM.dd}"

#在ES服务器创建的索引名

flush\_size => 2000

idle\_flush\_time => 10 }

#缓冲2000条数据或每10s提交一次给ES服务器

}

}

[root@logst bin]# ./logstash -f /etc/logstash/logstash.conf

[root@es1 ~]# http://192.168.1.11:9200/\_plugin/head/

#登陆ES.head插件查看weblog-2018.12.27索引

**-input.filebeat插件的使用**

filebeat插件：

辅助收集所有从web服务器发来的日志数据，减轻因部署logstash而带来的资源占用。

**步骤1：**web服务器安装filebeat

[root@web ~]# yum -y install filebeat

[root@web ~]# vim /etc/filebeat/filebeat.yml

paths:

- /var/log/httpd/access\_log

document\_type: apachelog

#elasticsearch:

#hosts: ["localhost:9200"]

logstash:

hosts: ["192.168.1.20:5044"]

[root@web ~]# systemctl start filebeat

**步骤2：**在logstash服务器配置beats插件

[root@logst ~]# vim /etc/logstash/logstash.conf

input{

beats { port => 5044 }

... ...

}

filter{

if [type] == "apachelog" {

grok{

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

}}

}

output{

stdout{ codec => "rubydebug" }

if [type] == "apachelog" {

elasticsearch{ ... ... }

}

}

[root@logst bin]# ./logstash -f /etc/logstash/logstash.conf

**步骤3：**ES.head查看结果

[root@es1 ~]# for i in `seq 30` ; do

> curl -so /dev/null 192.168.1.100

> done

#curl访问网页，获取access\_log

[root@es1 ~]# http://192.168.1.11:9200/\_plugin/head/

#查看apachelog-2018.12.27索引

**步骤4：**使用kibana绘图

[root@kbn ~]# firefox http://192.168.1.10:5601