**Fluentd，Kibana，Elasticsearch日志收集平台搭建**

# 安装

## 安装Fluentd

1. 安装前准备工作

由于Fluentd是分布式日志收集工具，因此官方建议在所有Fluentd部署NTP进行时钟同步。NTP部署方式如下：

安装NTP客户端：

#yum -y install ntp

配置文件，设定上层服务器：/etc/ntp.conf

如：server 172.24.2.214

启动ntp服务

#/etc/init.d/ntpd start

检查NTP服务状态如下，说明时钟同步已经完成：

# ntpstat

synchronised to NTP server (202.112.10.60) at stratum 3

time correct to within 62 ms

polling server every 1024 s

参考：http://linux.vbird.org/linux\_server/0440ntp.php

部署NTP后需要对节点机的一些参数进行调优，并重启机器生效：

修改主机文件句柄数

/etc/security/limits.conf

root soft nofile 65536

root hard nofile 65536

\* soft nofile 65536

\* hard nofile 65536

修改网络参数(sysctl命令修改 ，sysctl -p生效

sysctl net.ipv4.tcp\_tw\_recycle=1

sysctl net.ipv4.tcp\_tw\_reuse=1

sysctl net.ipv4.ip\_local\_port\_range=“10240 65535”

sysctl –p

参考：<http://docs.fluentd.org/v0.12/articles/before-install>

1. 安装RPM格式的td-agent

td-agent是Fluentd的易安装版本，该工具整合和一些相关的ruby依赖，使Fluentd可以一键安装（相当于devstack，由于Fluentd是基于Ruby内核的，td-agent安装时为我们一并安装了ruby的工具）。

当前Fluentd的稳定版本是v0.12（对应td-agent2）。

执行以下命令一键完成安装：

curl -L https://toolbelt.treasuredata.com/sh/install-redhat-td-agent2.sh | sh

上面的安装过程可能长时间没有反应，请等待。如果安装失败可以按照以下方案手动安装：

在/etc/yum.repos.d/目录创建一个新的源，如td.repo

源地址如下：

[treasuredata]

name=TreasureData

baseurl=http://packages.treasuredata.com/2/redhat/\$releasever/\$basearch

gpgcheck=1

gpgkey=https://packages.treasuredata.com/GPG-KEY-td-agent

更新YUM

#yum update

安装td-agent

#yum –y install td-agent

1. 启动服务

#/etc/init.d/td-agent start

#/etc/init.d/td-agent stop

#/etc/init.d/td-agent restart

#/etc/init.d/td-agent status

1. 配置文件和日志

/etc/td-agent/td-agent.conf

/var/log/td-agent.log

## 安装Elasticsearch

1. 安装JDK

下载RPM包

安装

#rpm –ivh xxxxx.rpm

配置环境变量

# vi /etc/profile

JAVA\_HOME=/usr/lib/jvm/xxx

JRE\_HOME=$JAVA\_HOME/jre

CLASS\_PATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

export JAVA\_HOME JRE\_HOME CLASS\_PATH PATH

刷新环境变量

source /etc/profile

1. 安装/配置Elasticsearch

下载RPM包安装

配置文件：

/etc/elasticsearch/elasticsearch.yml

/etc/sysconfig/elasticsearch(全局配置文件)

修改HOST地址（elasticsearch.yml）：

network.host: 172.24.2.214

启动服务

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable elasticsearch.service

sudo systemctl start elasticsearch.service

sudo systemctl stop elasticsearch .service

日志文件

/var/log/elasticsearch

## 安装Kibana

下载RPM包安装后，修改配置文件对接elasticsearch

配置文件地址：/etc/kibana/kibana.yml

elasticsearch.url: "http://172.24.2.214:9200"

server.host: "172.24.2.214"

#启动、停止服务

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable kibana.service

sudo systemctl start kibana.service

sudo systemctl stop kibana.service

#访问地址：

http://172.24.2.214:5601

# Fluentd配置

## 获取原始日志

安装日志格式化插件grok-parser，通过该插件对输出的日志进行初步的过滤格式化。

通过td-agent-gem命令安装：

#td-agent-gem install fluent-plugin-grok-parser –v 1.0.0

注意：v 0.12版本的fluentd只兼容到v 1.0.0版本的fluentd

Fluentd的插件（包括本身）以gem库的方式分发

gem库的一些相关操作

gem的lib目录，该目录下每个插件对应一个文件夹：

/opt/td-agent/embedded/lib/ruby/gems/2.1.0/gems/

列出所有gem安装包：

#td-agent-gem list

安装gem包：

# td-agent-gem install xxxxx –v version

卸载gem包：

# td-agent-gem uninstall xxxxx –v version

配置Fluentd输入源：

编辑/etc/td-agent/td-agent.conf，添加：

<source>

@type tail

path /opt/stack/logs/c-vol.log

pos\_file /var/log/td-agent/openstack/c\_vol.pos

format grok

<grok>

pattern %{TIMESTAMP\_ISO8601:Timestamp} %{NUMBER:Pid} %{LOGLEVEL:severity\_label} %{NOTSPACE:python\_module} \[%{NOTSPACE:request\_id} %{NOTSPACE:user\_id} %{NOTSPACE:tenant\_id}\] %{GREEDYDATA:logmsg}

</grok>

tag c\_vol

</source>

其中：

@type tail ：表示已增量方式从path定义的文件获取日志信息；

format grok ：通过grok方式格式化输入

<grok> pattern …… </grok> ：格式化使用的grok表达式

tag c\_vol ：该输入输出到名为c\_vol的输出

在编写grok表达式时需要一些在线调试工具，由于网络的原因通用的网址登陆不上，参考下面的帖子可以自己搭建一个grok表达式调试工具（<http://172.24.2.218:52900/>）。

帖子地址：<http://fengwan.blog.51cto.com/508652/1758845>

查询Fluentd插件的网站：https://rubygems.org/gems/

## 输出日志到Elasticsearch

对接Elasticsearch可以下载Fluentd输出插件：

td-agent-gem install fluent-plugin-elasticsearch

配置输出：

<match c\_vol>

@type elasticsearch

host 172.24.2.214

port 9200

index\_name c\_vol

type\_name c\_vol

logstash\_format true

logstash\_prefix c\_vol

flush\_interval 5s

</match>

其中：

<match c\_vol> ：该输出名称为c\_vol

@type elasticsearch ： 输出到elasticsearch

host 172.24.2.214 ： elasticsearch服务器地址

port 9200 ： elasticsearch开放端口

index\_name c\_vol ： elasticsearch侧创建的索引为c\_vol

关于多行日志的解析：

由于grok正则表达式默认只能解析单行日志，当出现错误时错误堆栈往往是以多行的形式打印的，因此想要解析多行日志，需要使用grok-parase的多行解析功能。配置如下：

<source>

@type tail

path /opt/stack/logs/c-vol.log

pos\_file /var/log/td-agent/openstack/c\_vol.pos

format multiline\_grok

grok\_pattern %{TIMESTAMP\_ISO8601:Timestamp} %{NUMBER:Pid} %{LOGLEVEL:severity\_label} %{NOTSPACE:python\_module} \[%{NOTSPACE:request\_id} %{NOTSPACE:user\_id} %{NOTSPACE:tenant\_id}\] %{GREEDYDATA:logmsg}

multiline\_start\_regexp /^(.\*) \[req(\S+) (\S+) (\S+)\] (.\*)/

tag c\_vol

multiline\_flush\_interval 5s

</source>

其中 :

multiline\_grok ：为多行解析器；

grok\_pattern ：不变；

multiline\_start\_regexp ：为首行的文本格式，grok插件，将该种格式之后的所有文本当做下一次解析的输入。

multiline\_flush\_interval ： fluentd开启此项时5s刷新一次缓存

# Elasticsearch相关Resful API接口

## 删除索引

curl -XDELETE 'localhost:9200/index\_name?pretty'

## Doc Values

字段的Doc Values是一种加快搜索速度的数据结构，该数据结构会在磁盘上建立索引，占用磁盘空间。默认情形下所有Field都是打开Doc Values能力的。

打开/Doc Values特性的API：

curl -XPUT 'localhost:9200/my\_index?pretty' -H 'Content-Type: application/json' -d'

{

"mappings": {

"my\_type": {

"properties": {

"my\_field1": {

"type": "keyword"

},

"my\_field1": {

"type": "keyword",

"doc\_values": false

}

}

}

}

}'

参考：

https://www.elastic.co/guide/en/elasticsearch/reference/5.1/doc-values.html

## 使字段可以排序

在Kibana对某个字段进行排序时有如下错误：

Discover: Fielddata is disabled on text fields by default. Set fielddata=true on [logmsg] in order to load fielddata in memory by uninverting the inverted index. Note that this can however use significant memory. Alternatively use a keyword field instead.

解决方案：

curl -XPUT 'http://172.24.2.214:9200/c\_vol-\*/\_mapping/severity\_label\_type?update\_all\_types&pretty' -H 'Content-Type: application/json' -d'

{

"properties": {

"severity\_label": {

"type": "text",

"fielddata": true

}

}

}'

参考：

https://www.elastic.co/guide/en/elasticsearch/reference/5.4/indices-put-mapping.html#merging-conflicts

<https://www.elastic.co/guide/en/elasticsearch/reference/5.1/fielddata.html>

## term方式和match方式

match方式会对搜索的关键词进行分词是模糊匹配。term并不进行分词是一种精确匹配方式，对于text类型在Fluentd中需要指定my\_field.keyword进行搜索才能得到结果。