ELK 日志分析平台部署文档

计划分为四部分
1,ELK 介绍及参考网站
2,ELK 小企业部署
3,ELK+redis 中型企业部署
4,flume+kafka+zookeeper+ELK+HDFS+LVS 中大型企业集群环境部署

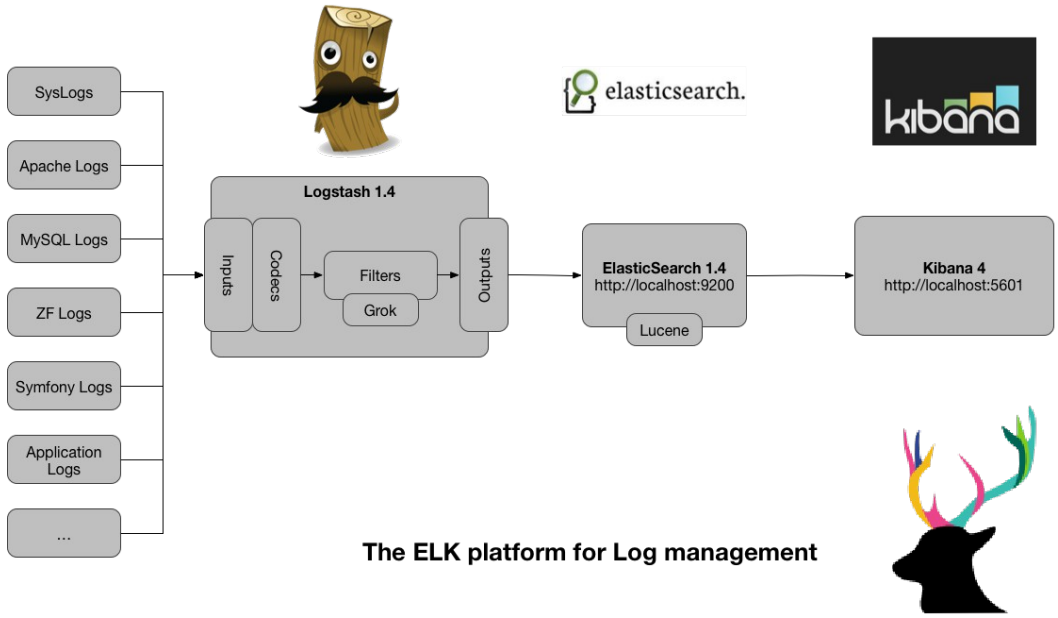
第一部分：ELK 介绍及参考网站

简要介绍

Elasticsearch + Logstash + Kibana（ELK）是一套开源的日志管理方案，分析网站的访问情况时我们一般会借助 Google/百度/CNZZ 等方式嵌入 JS 做数据统计，但是当网站访问异常或者被攻击时我们需要在后台分析如 Nginx 的具体日志，而 Nginx 日志分割/GoAccess/Awstats 都是相对简单的单节点解决方案，针对分布式集群或者数据量级较大时会显得心有余而力不足，而 ELK 的出现可以使我们从容面对新的挑战。

- Logstash：负责日志的收集，处理和储存
- Elasticsearch：负责日志检索和分析
- Kibana：负责日志的可视化
- Redis：在 logstash 和 Elasticsearch 中间作为消息队列，以减轻 es 集群的压力
- Flume：和 logstash 功能一样，一个分布式的日志收集工具
- Kafka：分布式消息队列，替代 redis，毕竟 Redis 作为消息队列并不是它的强项
- Zookeeper：分布式程序协调服务，是 Google 的 Chubby 一个开源的实现，kafka 需要用到
- HDFS：分布式文件系统，将日志存储在这里供 mapreduce 分析，当然也可以用 storm 来实时分析
- LVS：负载均衡，避免 Elasticsearch/Kibana 单点

看下面的图，或许更好理解每个组组件的职责：



参考网站

https://wsgzao.github.io/post/elk/

http://www.chenshake.com/centos-install-7-x-elk-elasticsearchlogstashkibana/

http://blog.chinaunix.net/xmlrpc.php?r=blog/article&uid=17291169&id=4898582

一个老外的 ELK 视频，视频地址 http://yunpan.cn/cd5feBr4diFDn 访问密码 019a

官方网站

ELK：https://www.elastic.co/

ELKstack 中文指南：http://kibana.logstash.es/content/index.html

Redis：http://redis.io/

Flume：https://flume.apache.org/

Kafka：http://kafka.apache.org/

Zookeeper：https://zookeeper.apache.org/

author：云尘　　网站：www.jikecloud.com

LVS：http://zh.linuxvirtualserver.org/

第二部分：ELK 小企业部署

基本环境配置

- 1，JDK

http://www.oracle.com/technetwork/java/javase/downloads/index.html

下载解压到一个目录

修改/etc/profile

```
JAVA_HOME=/home/app/soft/jdk1.8.0_73
PATH=$JAVA_HOME/bin:$ECLIPSE_HOME:$MAVEN_HOME/bin:$PATH
```

当然也可以直接 yum 按照

yum install java-1.7.0-openjdk

- 2，设置 FQDN

创建 SSL 证书时需要用到 FQDN

```
#修改 hostname
cat /etc/hostname
elk

#修改 hosts
cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.3.178 xdev.ooxx.com xdev

#刷新环境
hostname -F /etc/hostname

#复查结果
```

```
hostname -f
elk.ooxx.com

hostname
elk
```

服务端：Elasticsearch

- 1，下载安装

```
wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.1.noarch.rpm
yum localinstall elasticsearch-1.7.1.noarch.rpm
```

- 2，启动相关服务

```
service elasticsearch start
service elasticsearch status
```

- 3，查看 Elasticsearch 的配置文件

```
rpm -qc elasticsearch

/etc/elasticsearch/elasticsearch.yml
/etc/elasticsearch/logging.yml
/etc/init.d/elasticsearch
/etc/sysconfig/elasticsearch
/usr/lib/sysctl.d/elasticsearch.conf
/usr/lib/systemd/system/elasticsearch.service
/usr/lib/tmpfiles.d/elasticsearch.conf
```

- 4，查看端口使用情况

```
netstat -nltp

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9200            0.0.0.0:*               LISTEN      1765/java
tcp        0      0 0.0.0.0:9300            0.0.0.0:*               LISTEN      1765/java
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1509/sshd
tcp        0      0 :::22                   :::*                    LISTEN      1509/sshd
```

- #测试访问

curl -X GET http://localhost:9200/

服务端：Kibana

- 下载 tar 包

wget https://download.elastic.co/kibana/kibana/kibana-4.1.1-linux-x64.tar.gz

解压

tar zxf kibana-4.1.1-linux-x64.tar.gz -C /usr/local/
cd /usr/local/
mv kibana-4.1.1-linux-x64 kibana

- 创建 kibana 服务

vi /etc/rc.d/init.d/kibana

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          kibana
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Runs kibana daemon
# Description: Runs the kibana daemon as a non-root user
### END INIT INFO
# Process name
NAME=kibana
DESC="Kibana4"
PROG="/etc/init.d/kibana"
# Configure location of Kibana bin
KIBANA_BIN=/usr/local/kibana/bin
# PID Info
PID_FOLDER=/var/run/kibana/
PID_FILE=/var/run/kibana/$NAME.pid
LOCK_FILE=/var/lock/subsys/$NAME
PATH=/bin:/usr/bin:/sbin:/usr/sbin:$KIBANA_BIN
DAEMON=$KIBANA_BIN/$NAME
# Configure User to run daemon process
DAEMON_USER=root
# Configure logging location
KIBANA_LOG=/var/log/kibana.log
# Begin Script
RETVAL=0
if [ `id -u` -ne 0 ]; then
        echo "You need root privileges to run this script"
        exit 1
```

```
fi
# Function library
. /etc/init.d/functions

start() {
        echo -n "Starting $DESC : "
pid=`pidofproc -p $PID_FILE kibana`
        if [ -n "$pid" ] ; then
                echo "Already running."
                exit 0
        else
        # Start Daemon
if [ ! -d "$PID_FOLDER" ] ; then
                        mkdir $PID_FOLDER
                fi
daemon --user=$DAEMON_USER --pidfile=$PID_FILE $DAEMON 1>"$KIBANA_LOG" 2>&1
&
                sleep 2
                pidofproc node > $PID_FILE
                RETVAL=$?
                [[ $? -eq 0 ]] && success || failure
echo
                [ $RETVAL = 0 ] && touch $LOCK_FILE
                return $RETVAL
        fi
}
reload()
{
    echo "Reload command is not implemented for this service."
    return $RETVAL
}
stop() {
        echo -n "Stopping $DESC : "
        killproc -p $PID_FILE $DAEMON
        RETVAL=$?
echo
        [ $RETVAL = 0 ] && rm -f $PID_FILE $LOCK_FILE
}

case "$1" in
  start)
        start
;;
  stop)
        stop
        ;;
  status)
        status -p $PID_FILE $DAEMON
        RETVAL=$?
        ;;
```

```
  restart)
        stop
        start
        ;;
  reload)
reload
;;
  *)
# Invalid Arguments, print the following message.
        echo "Usage: $0 {start|stop|status|restart}" >&2
exit 2
        ;;
esac
```

- 修改启动权限

```
chmod +x /etc/rc.d/init.d/kibana
```

- 启动 kibana 服务

```
service kibana start
service kibana status
```

- 查看端口

```
netstat -nltp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9200            0.0.0.0:*               LISTEN      1765/java
tcp        0      0 0.0.0.0:9300            0.0.0.0:*               LISTEN      1765/java
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1509/sshd
tcp        0      0 0.0.0.0:5601            0.0.0.0:*               LISTEN      1876/node
tcp        0      0 :::22                   :::*                    LISTEN      1509/sshd
```

服务端：Logstash

- 下载 rpm 包

wget https://download.elastic.co/logstash/logstash/packages/centos/logstash-1.5.4-1.noarch.rpm
yum localinstall logstash-1.5.4-1.noarch.rpm

- 设置 ssl，之前设置的 FQDN 是 xdev.ooxx.com

cd /etc/pki/tls

openssl req -subj '/CN=xdev.ooxx.com/' -x509 -days 3650 -batch -nodes -newkey rsa:2048 -keyout private/logstash-forwarder.key -out certs/logstash-forwarder.crt

- 创建一个 01-logstash-initial.conf 文件

```
cat > /etc/logstash/conf.d/01-logstash-initial.conf << EOF
input {
  lumberjack {
    port => 5000
    type => "logs"
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %
{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
output {
  elasticsearch { host => xdev.ooxx.com }
  stdout { codec => rubydebug }
}
EOF
```

- 启动 logstash 服务

service logstash start
service logstash status

- 查看 5000 端口

```
netstat -nltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9200            0.0.0.0:*               LISTEN      1765/java
tcp        0      0 0.0.0.0:9300            0.0.0.0:*               LISTEN      1765/java
tcp        0      0 0.0.0.0:9301            0.0.0.0:*               LISTEN      2309/java
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1509/sshd
tcp        0      0 0.0.0.0:5601            0.0.0.0:*               LISTEN      1876/node
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN      2309/java
tcp        0      0 :::22                   :::*                    LISTEN      1509/sshd
```

客户端：Logstash Forwarder

- #登陆到客户端，安装 Logstash Forwarder

```
wget https://download.elastic.co/logstash-forwarder/binaries/logstash-forwarder-0.4.0-1.x86_64.rpm
yum localinstall logstash-forwarder-0.4.0-1.x86_64.rpm
```

- 查看 logstash-forwarder 的配置文件位置

```
rpm -qc logstash-forwarder
/etc/logstash-forwarder.conf
```

- 备份配置文件

```
cp /etc/logstash-forwarder.conf /etc/logstash-forwarder.conf.save
```

- 编辑 /etc/logstash-forwarder.conf，需要根据实际情况进行修改

```
cat > /etc/logstash-forwarder.conf << EOF
{
  "network": {
    "servers": [ "xdev.ooxx.com:5000" ],
    "ssl ca": "/etc/pki/tls/certs/logstash-forwarder.crt",
    "timeout": 15
  },
  "files": [
    {
      "paths": [
        "/var/log/messages",
        "/var/log/secure"
      ],
      "fields": { "type": "syslog" }
    }
  ]
}
EOF
```

- 启动服务

service logstash-forwarder start
service logstash-forwarder status

- 访问 Kibana，Time-field name 选择 @timestamp

http://localhost:5601/

- 增加节点和客户端配置一样，注意同步证书

/etc/pki/tls/certs/logstash-forwarder.crt

到这里就可以访问测试，这里收集了系统日志

下面加入 nginx 日志的收集

配置 Nginx 日志策略

- #修改客户端 Logstash Forwarder 配置

```
vi /etc/logstash-forwarder.conf
{
  "network": {
    "servers": [ "xdev.ooxx.com:5000" ],
    "ssl ca": "/etc/pki/tls/certs/logstash-forwarder.crt",
    "timeout": 15
  },
  "files": [
    {
      "paths": [
        "/var/log/messages",
        "/var/log/secure"
      ],
      "fields": { "type": "syslog" }
    }, {
      "paths": [
        "/app/local/nginx/logs/access.log"
      ],
      "fields": { "type": "nginx" }
    }
  ]
}
```

- logstash 服务端增加 patterns

mkdir /opt/logstash/patterns

vi /opt/logstash/patterns/nginx

```
NGUSERNAME [a-zA-Z\.\@\-\+_%]+
NGUSER %{NGUSERNAME}
NGINXACCESS %{IPORHOST:remote_addr} - - \[%{HTTPDATE:time_local}\] "%
{WORD:method} %{URIPATH:path}(?:%{URIPARAM:param})? HTTP/%
{NUMBER:httpversion}" %{INT:status} %{INT:body_bytes_sent} %{QS:http_referer} %
{QS:http_user_agent}
```

- 官网 pattern 的 debug 在线工具

https://grokdebug.herokuapp.com/

- 修改 logstash 权限

```
chown -R logstash:logstash /opt/logstash/patterns
```

- 修改服务端 logstash 配置

vi /etc/logstash/conf.d/01-logstash-initial.conf

```
input {
  lumberjack {
    port => 5000
    type => "logs"
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %
{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
  if [type] == "nginx" {
    grok {
      match => { "message" => "%{NGINXACCESS}" }
    }
  }
}
output {
  elasticsearch { host => xdev.ooxx.com }
  stdout { codec => rubydebug }
}
```

OK，重启 logstash 以及 Logstash Forwarder，就可以看到数据了

后面加入图两张

其他问题

- 修改 kibana，编辑 kibana.yaml

vi /usr/local/kibana/config/kibana.yml

- 安装 es 的管理插件

es 官方提供一个用于管理 es 的插件，可清晰直观看到 es 集群的状态，以及对集群的操作管理，安装方法如下：

/usr/local/elasticsearch/bin/plugin -i mobz/elasticsearch-head

安装好之后，访问方式为： http://192.168.3.178:9200/_plugin/head

- 增加 elasticsearch 的 JVM 内存

```
#修改 elasticsearch.in.sh
vi /usr/share/elasticsearch/bin/elasticsearch.in.sh

if [ "x$ES_MIN_MEM" = "x" ]; then
    ES_MIN_MEM=1g
fi
if [ "x$ES_MAX_MEM" = "x" ]; then
    ES_MAX_MEM=1g
```