



## 使用filebeat收集java多行日志

### 收集java多行日志

```
1 vim /etc/filebeat/filebeat.yml
2 filebeat.inputs:
3   - type: log
4     enabled: true
```

```
5  paths:
6    - /var/log/elasticsearch/elasticsearch.log
7
8  multiline.pattern: ^\[
9  multiline.negate: true
10  multiline.match: after
11
12  output.elasticsearch:
13    hosts: ["10.0.0.51:9200"]
14    index: "es-%[agent.version]}-%{+yyyy.MM}"
15
16  setup.ilm.enabled: false
17  setup.template.enabled: false
18
19  logging.level: info
20  logging.to_files: true
```

## 使用Redis作为EBK缓存

### 1.修改nginx修改为json格式

```
1  systemctl stop nginx
2  > /var/log/nginx/access.log
3  vim /etc/nginx/nginx.conf
4  access_log /var/log/nginx/access.log json;
5  systemctl restart nginx
6  curl 127.0.0.1
7  cat /var/log/nginx/access.log
```

### 2.安装部署redis 两台都装

```
1  yum install redis -y
2  sed -i 's#^bind 127.0.0.1#bind 127.0.0.1 10.0.0.51#'
   /etc/redis.conf
```

```
3 systemctl restart redis
4 redis-cli -h 10.0.0.51
```

### 3.修改filebeat配置文件

```
1 cat > /etc/filebeat/filebeat.yml << 'EOF'
2 filebeat.inputs:
3 - type: log
4   enabled: true
5   paths:
6   - /var/log/nginx/access.log
7   json.keys_under_root: true
8   json.overwrite_keys: true
9   tags: ["access"]
10
11 - type: log
12   enabled: true
13   paths:
14   - /var/log/nginx/error.log
15   tags: ["error"]
16
17 output.redis:
18   hosts: ["10.0.0.51"]
19   keys:
20   - key: "nginx_access"
21     when.contains:
22       tags: "access"
23   - key: "nginx_error"
24     when.contains:
25       tags: "error"
26
27 setup.ilm.enabled: false
28 setup.template.enabled: false
29
30 logging.level: info
31 logging.to_files: true
```

```
32 EOF
33 systemctl restart filebeat
```

## 4.生成测试数据

```
1 for i in {0..100};do curl -s 127.0.0.1;done
```

## 5.查看redis数据

### #查看有多少KEY

```
1 10.0.0.51:6379> keys *
2 1) "nginx_error"
3 2) "nginx_access"
```

### #查看数据类型

```
1 10.0.0.51:6379> TYPE nginx_access
2 list
```

### #查看列表有多长

```
1 10.0.0.51:6379> LLEN nginx_access
2 (integer) 1001
```

### #查看列表元素

```
1 10.0.0.51:6379> LRANGE nginx_access 0 10
```

## 6.安装logstash

把两个rpm、包拉进来

```
1 rpm -ivh jdk-8u181-linux-x64.rpm
2 rpm -ivh logstash-7.9.1.rpm
```

## 7.编写logstash配置文件

```
1 cat >/etc/logstash/conf.d/redis.conf << 'EOF'
2 input {
3   redis {
```

```
4  host => "10.0.0.51"
5  port => "6379"
6  db => "0"
7  key => "nginx_access"
8  data_type => "list"
9  }
10 redis {
11  host => "10.0.0.51"
12  port => "6379"
13  db => "0"
14  key => "nginx_error"
15  data_type => "list"
16  }
17 }
18
19 output {
20  stdout {}
21  if "access" in [tags] {
22  elasticsearch {
23  hosts => "http://10.0.0.51:9200"
24  manage_template => false
25  index => "nginx_access-%{+yyyy.MM}"
26  }
27  }
28  if "error" in [tags] {
29  elasticsearch {
30  hosts => "http://10.0.0.51:9200"
31  manage_template => false
32  index => "nginx_error-%{+yyyy.MM}"
33  }
34  }
35  }
36  EOF
```

## 8.前台启动logstash测试 时间会很长

```
1 /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/redis.conf
```

## 9.测试成功后使用systemd启动

```
1 systemctl start logstash.service
2 systemctl status logstash.service
```

## 10.优化配置文件

### #优化filebeat配置文件

```
1 cat > /etc/filebeat/filebeat.yml << 'EOF'
2 filebeat.inputs:
3   - type: log
4     enabled: true
5     paths:
6       - /var/log/nginx/access.log
7     json.keys_under_root: true
8     json.overwrite_keys: true
9     tags: ["access"]
10
11   - type: log
12     enabled: true
13     paths:
14       - /var/log/nginx/error.log
15     tags: ["error"]
16
17 output.redis:
18   hosts: ["10.0.0.51"]
19   key: "nginx"
20
21 setup.ilm.enabled: false
22 setup.template.enabled: false
23
24 logging.level: info
25 logging.to_files: true
```

```
26 EOF
27 systemctl restart filebeat
```

## #优化logstash配置文件

```
1 cat >/etc/logstash/conf.d/redis.conf << 'EOF'
2 input {
3   redis {
4     host => "10.0.0.51"
5     port => "6379"
6     db => "0"
7     key => "nginx"
8     data_type => "list"
9   }
10 }
11
12 output {
13   stdout {}
14   if "access" in [tags] {
15     elasticsearch {
16       hosts => "http://10.0.0.51:9200"
17       manage_template => false
18       index => "nginx_access-%{+yyyy.MM}"
19     }
20   }
21   if "error" in [tags] {
22     elasticsearch {
23       hosts => "http://10.0.0.51:9200"
24       manage_template => false
25       index => "nginx_error-%{+yyyy.MM}"
26     }
27   }
28 }
29 EOF
```

## 11.多个redis备份

### #filebeat读取多个redis

```
1 cat > /etc/filebeat/filebeat.yml << 'EOF'
2 filebeat.inputs:
3   - type: log
4     enabled: true
5     paths:
6       - /var/log/nginx/access.log
7     json.keys_under_root: true
8     json.overwrite_keys: true
9     tags: ["access"]
10
11  - type: log
12    enabled: true
13    paths:
14      - /var/log/nginx/error.log
15    tags: ["error"]
16
17  output.redis:
18    hosts: ["10.0.0.51","10.0.0.7"]
19    key: "nginx"
20
21  setup.ilm.enabled: false
22  setup.template.enabled: false
23
24  logging.level: info
25  logging.to_files: true
26 EOF
```

### #logstash读取多个redis

```
1 cat >/etc/logstash/conf.d/redis.conf << 'EOF'
2 input {
3   redis {
4     host => "10.0.0.51"
5     port => "6379"
```



```
6  db => "0"
7  key => "nginx"
8  data_type => "list"
9  }
10 }
11
12 input {
13   redis {
14     host => "10.0.0.7"
15     port => "6379"
16     db => "0"
17     key => "nginx"
18     data_type => "list"
19   }
20 }
21 output {
22   stdout {}
23   if "access" in [tags] {
24     elasticsearch {
25       hosts => "http://10.0.0.51:9200"
26       manage_template => false
27       index => "nginx_access-%{+yyyy.MM}"
28     }
29   }
30   if "error" in [tags] {
31     elasticsearch {
32       hosts => "http://10.0.0.51:9200"
33       manage_template => false
34       index => "nginx_error-%{+yyyy.MM}"
35     }
36   }
37 }
38 EOF
```

# 使用kafka作为缓存

## 1.配置hosts和密钥 三台机子都操作

```
1 cat >/etc/hosts<<EOF
2 10.0.0.51 db-51
3 10.0.0.52 db-52
4 10.0.0.53 db-53
5 EOF
1 ssh-keygen
2 ssh-copy-id 10.0.0.52
3 ssh-copy-id 10.0.0.53
```

## 2.安装配置zookeeper

### #db51操作

```
1 cd /data/soft
2 tar xzf zookeeper-3.4.11.tar.gz -C /opt/
3 ln -s /opt/zookeeper-3.4.11/ /opt/zookeeper
4 mkdir -p /data/zookeeper
5 cp /opt/zookeeper/conf/zoo_sample.cfg /opt/zookeeper/conf/zoo.cfg
6 cat >/opt/zookeeper/conf/zoo.cfg<<EOF
7 tickTime=2000
8 initLimit=10
9 syncLimit=5
10 dataDir=/data/zookeeper
11 clientPort=2181
12 server.1=10.0.0.51:2888:3888
13 server.2=10.0.0.52:2888:3888
14 server.3=10.0.0.53:2888:3888
15 EOF
1 echo "1" > /data/zookeeper/myid
2 cat /data/zookeeper/myid
3 scp -r /opt/zookeeper* 10.0.0.52:/opt/
4 scp -r /opt/zookeeper* 10.0.0.53:/opt/
```

### #db52操作

```
1 mkdir -p /data/zookeeper
2 echo "2" > /data/zookeeper/myid
3 cat /data/zookeeper/myid
```

### #db53操作

```
1 mkdir -p /data/zookeeper
2 echo "3" > /data/zookeeper/myid
3 cat /data/zookeeper/myid
```

## 3.所有节点启动zookeeper

```
1 /opt/zookeeper/bin/zkServer.sh start
```

## 4.每个节点都检查

```
1 /opt/zookeeper/bin/zkServer.sh status
```

## 5.测试zookeeper

在一个节点上执行,创建一个频道

```
1 /opt/zookeeper/bin/zkCli.sh -server 10.0.0.51:2181
2 create /test "hello"
```

在其他节点上看能否接收到

```
1 /opt/zookeeper/bin/zkCli.sh -server 10.0.0.52:2181
2 get /test
```

## 6.安装部署kafka

### #db51操作

```
1 cd /data/soft/
2 tar xzf kafka_2.11-1.0.0.tgz -C /opt/
3 ln -s /opt/kafka_2.11-1.0.0/ /opt/kafka
4 mkdir /opt/kafka/logs
5 cat >/opt/kafka/config/server.properties<<EOF
```

```

6 broker.id=1
7 listeners=PLAINTEXT://10.0.0.51:9092
8 num.network.threads=3
9 num.io.threads=8
10 socket.send.buffer.bytes=102400
11 socket.receive.buffer.bytes=102400
12 socket.request.max.bytes=104857600
13 log.dirs=/opt/kafka/logs
14 num.partitions=1
15 num.recovery.threads.per.data.dir=1
16 offsets.topic.replication.factor=1
17 transaction.state.log.replication.factor=1
18 transaction.state.log.min.isr=1
19 log.retention.hours=24
20 log.segment.bytes=1073741824
21 log.retention.check.interval.ms=300000
22 zookeeper.connect=10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181
23 zookeeper.connection.timeout.ms=6000
24 group.initial.rebalance.delay.ms=0
25 EOF

1 scp -r /opt/kafka* 10.0.0.52:/opt/
2 scp -r /opt/kafka* 10.0.0.53:/opt/

```

### #db52操作

```

1 sed -i "s#10.0.0.51:9092#10.0.0.52:9092#g" /opt/kafka/config/server.properties
2 sed -i "s#broker.id=1#broker.id=2#g" /opt/kafka/config/server.properties

```

### #db53操作

```

1 sed -i "s#10.0.0.51:9092#10.0.0.53:9092#g" /opt/kafka/config/server.properties
2 sed -i "s#broker.id=1#broker.id=3#g" /opt/kafka/config/server.properties

```

## 7.前台启动测试

```
1 /opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/server.properties
```

## 8.验证进程

```
1 jps
```

## 9.测试创建topic

```
1 /opt/kafka/bin/kafka-topics.sh --create --zookeeper 10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181 --partitions 3 --replication-factor 3 --topic kafkatest
```

## 10.测试获取toppid

```
1 /opt/kafka/bin/kafka-topics.sh --describe --zookeeper 10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181 --topic kafkatest
```

## 11.测试删除topic

```
1 /opt/kafka/bin/kafka-topics.sh --delete --zookeeper 10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181 --topic kafkatest
```

## 12.kafka测试命令发送消息

### #1.创建命令

```
1 /opt/kafka/bin/kafka-topics.sh --create --zookeeper 10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181 --partitions 3 --replication-factor 3 --topic messagetest
```

### #2.测试发送消息

```
1 /opt/kafka/bin/kafka-console-producer.sh --broker-list 10.0.0.51:9092,10.0.0.52:9092,10.0.0.53:9092 --topic messagetest
```

### #3.其他节点测试接收

```
1 /opt/kafka/bin/kafka-console-consumer.sh --zookeeper 10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181 --topic messagetest --from-beginning
```

### #4.测试获取所有的频道

```
1 /opt/kafka/bin/kafka-topics.sh --list --zookeeper
10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181
```

### 13.测试成功之后,可以放在后台启动

```
1 /opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/s
server.properties
```

### 14.修改filebeat配置文件

```
1 cat >/etc/filebeat/filebeat.yml << 'EOF'
2 filebeat.inputs:
3 - type: log
4   enabled: true
5   paths:
6   - /var/log/nginx/access.log
7   tags: ["access"]
8
9 - type: log
10  enabled: true
11  paths:
12  - /var/log/nginx/error.log
13  tags: ["error"]
14
15 output.kafka:
16   hosts: ["10.0.0.51:9092", "10.0.0.52:9092", "10.0.0.53:9092"]
17   topic: 'filebeat'
18
19 setup.ilm.enabled: false
20 setup.template.enabled: false
21 EOF
```

### 15.修改logstash配置文件

```
1 cat >/etc/logstash/conf.d/kafka.conf <<EOF
2 input {
3   kafka{
```

```

4  bootstrap_servers=>["10.0.0.51:9092,10.0.0.52:9092,10.0.0.53:90
92"]
5  topics=>["filebeat"]
6  #group_id=>"logstash"
7  codec => "json"
8  }
9  }
10
11 output {
12   stdout {}
13   if "access" in [tags] {
14     elasticsearch {
15       hosts => "http://10.0.0.51:9200"
16       manage_template => false
17       index => "nginx_access-%{+yyyy.MM}"
18     }
19   }
20   if "error" in [tags] {
21     elasticsearch {
22       hosts => "http://10.0.0.51:9200"
23       manage_template => false
24       index => "nginx_error-%{+yyyy.MM}"
25     }
26   }
27 }
28 EOF

```

## 16.启动logstash并测试

### #1.前台启动

```

1 /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/kafka.conf

```

### #2.后台启动

```

1 systemctl start logstash

```

