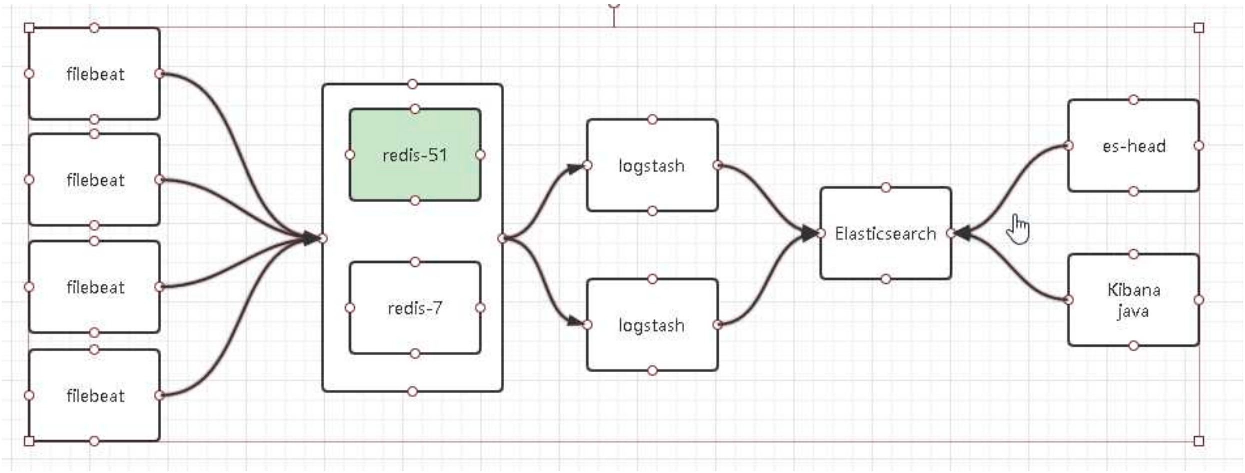

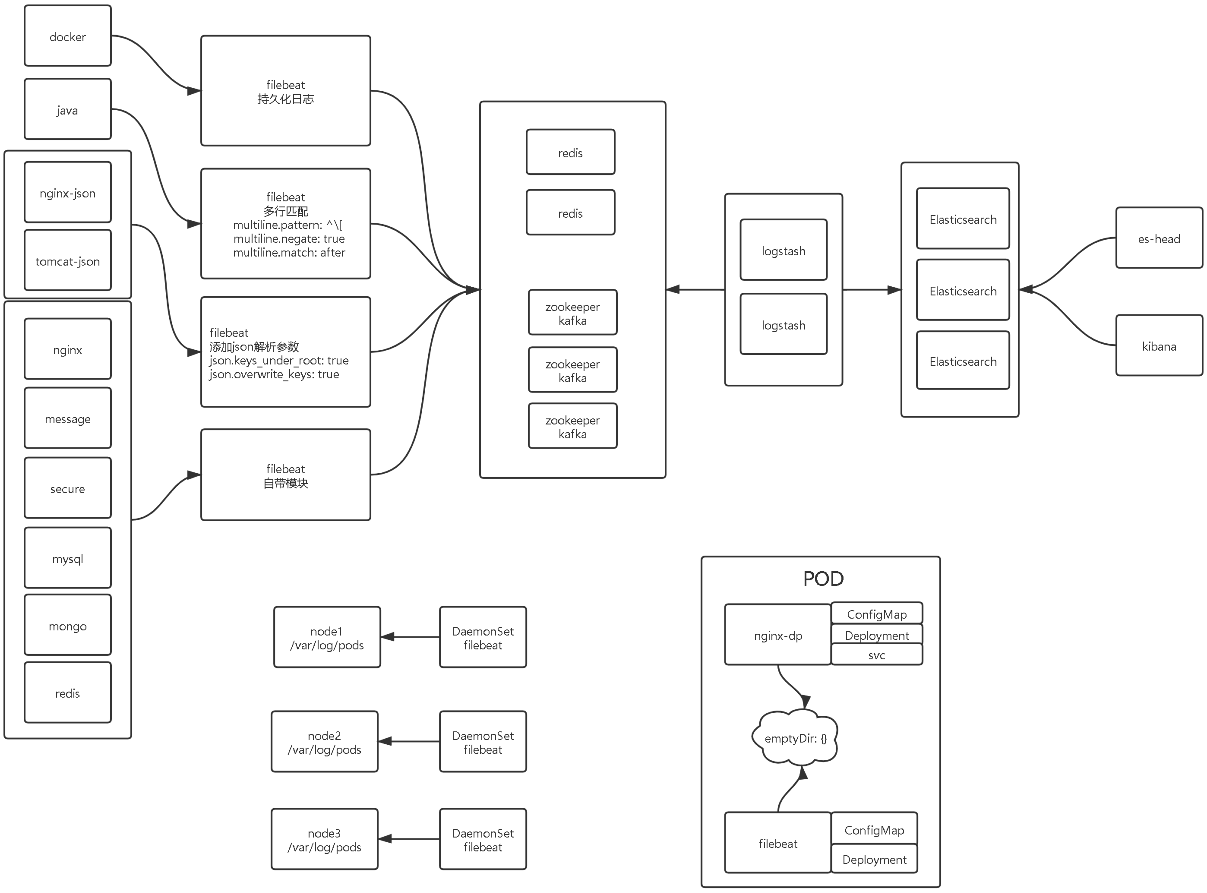

使用filebeat收集java多行日志

收集java多行日志

```
vim /etc/filebeat/filebeat.yml
filebeat.inputs:
- type: log
 enabled: true
```

```
paths:
- /var/log/elasticsearch/elasticsearch.log

multiline.pattern: ^\[
multiline.negate: true
 multiline.match: after

output.elasticsearch:
 hosts: ["10.0.0.51:9200"]
 index: "es-%{[agent.version]}-%{+yyyy.MM}"

setup.ilm.enabled: false
setup.template.enabled: false

logging.level: info
logging.to_files: true
```

使用Redis作为EBK缓存

1.修改nginx修改为json格式

```
systemctl stop nginx
> /var/log/nginx/access.log
vim /etc/nginx/nginx.conf
access_log /var/log/nginx/access.log json;
systemctl restart nginx
curl 127.0.0.1
cat /var/log/nginx/access.log
```

2.安装部署redis　两台都装

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

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

3.修改filebeat配置文件

```
cat > /etc/filebeat/filebeat.yml << 'EOF'
filebeat.inputs:
- type: log
 enabled: true
 paths:
 - /var/log/nginx/access.log
 json.keys_under_root: true
 json.overwrite_keys: true
 tags: ["access"]

- type: log
 enabled: true
 paths:
 - /var/log/nginx/error.log
 tags: ["error"]

output.redis:
 hosts: ["10.0.0.51"]
 keys:
 - key: "nginx_access"
 when.contains:
 tags: "access"
 - key: "nginx_error"
 when.contains:
 tags: "error"

setup.ilm.enabled: false
setup.template.enabled: false

logging.level: info
logging.to_files: true
```

```
EOF
systemctl restart filebeat
```

4.生成测试数据

```
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

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

#查看数据类型

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

#查看列表有多长

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

#查看列表元素

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

6.安装logstash

把两个rpm、包拉进来

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

7.编写logstash配置文件

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

```
host => "10.0.0.51"
port => "6379"
db => "0"
key => "nginx_access"
data_type => "list"
}
 redis {
 host => "10.0.0.51"
 port => "6379"
 db => "0"
 key => "nginx_error"
 data_type => "list"
 }
}

output {
 stdout {}
 if "access" in [tags] {
 elasticsearch {
 hosts => "http://10.0.0.51:9200"
 manage_template => false
 index => "nginx_access-%{+yyyy.MM}"
 }
 }
 if "error" in [tags] {
 elasticsearch {
 hosts => "http://10.0.0.51:9200"
 manage_template => false
 index => "nginx_error-%{+yyyy.MM}"
 }
 }
}
EOF
```

8.前台启动logstash测试　时间会很长

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

9.测试成功后使用systemd启动

```
systemctl start logstash.service
systemctl status logstash.service
```

10.优化配置文件

#优化filebeat配置文件

```
cat > /etc/filebeat/filebeat.yml << 'EOF'
filebeat.inputs:
- type: log
 enabled: true
 paths:
 - /var/log/nginx/access.log
 json.keys_under_root: true
 json.overwrite_keys: true
 tags: ["access"]

- type: log
 enabled: true
 paths:
 - /var/log/nginx/error.log
 tags: ["error"]

output.redis:
 hosts: ["10.0.0.51"]
 key: "nginx"

setup.ilm.enabled: false
setup.template.enabled: false

logging.level: info
logging.to_files: true
```

```
EOF
systemctl restart filebeat
```

#优化logstash配置文件

```
cat >/etc/logstash/conf.d/redis.conf << 'EOF'
input {
 redis {
 host => "10.0.0.51"
 port => "6379"
 db => "0"
 key => "nginx"
 data_type => "list"
 }
}

output {
 stdout {}
 if "access" in [tags] {
 elasticsearch {
 hosts => "http://10.0.0.51:9200"
 manage_template => false
 index => "nginx_access-%{+yyyy.MM}"
 }
 }
 if "error" in [tags] {
 elasticsearch {
 hosts => "http://10.0.0.51:9200"
 manage_template => false
 index => "nginx_error-%{+yyyy.MM}"
 }
 }
}
EOF
```

11.多个redis备份

#filebeat读取多个redis

```
cat > /etc/filebeat/filebeat.yml << 'EOF'
filebeat.inputs:
- type: log
 enabled: true
 paths:
 - /var/log/nginx/access.log
 json.keys_under_root: true
 json.overwrite_keys: true
 tags: ["access"]

- type: log
 enabled: true
 paths:
 - /var/log/nginx/error.log
 tags: ["error"]

output.redis:
 hosts: ["10.0.0.51","10.0.0.7"]
 key: "nginx"

setup.ilm.enabled: false
setup.template.enabled: false

logging.level: info
logging.to_files: true
EOF
```

#logstash读取多个redis

```
cat >/etc/logstash/conf.d/redis.conf << 'EOF'
input {
 redis {
 host => "10.0.0.51"
 port => "6379"
 db => "0"
 key => "nginx"
 data_type => "list"
 }
 }

 input {
  redis {
  host => "10.0.0.7"
  port => "6379"
  db => "0"
  key => "nginx"
  data_type => "list"
  }
 }
 output {
  stdout {}
  if "access" in [tags] {
  elasticsearch {
  hosts => "http://10.0.0.51:9200"
  manage_template => false
  index => "nginx_access-%{+yyyy.MM}"
  }
  }
  if "error" in [tags] {
  elasticsearch {
  hosts => "http://10.0.0.51:9200"
  manage_template => false
  index => "nginx_error-%{+yyyy.MM}"
  }
  }
 }
 EOF
```

使用kafka作为缓存

1.配置hosts和密钥 三台机子都操作

```
cat >/etc/hosts<<EOF
10.0.0.51 db-51
10.0.0.52 db-52
10.0.0.53 db-53
EOF
```

```
ssh-keygen
ssh-copy-id 10.0.0.52
ssh-copy-id 10.0.0.53
```

2.安装配置zookeeper

#db51操作

```
cd /data/soft
tar zxf zookeeper-3.4.11.tar.gz -C /opt/
ln -s /opt/zookeeper-3.4.11/ /opt/zookeeper
mkdir -p /data/zookeeper
cp /opt/zookeeper/conf/zoo_sample.cfg /opt/zookeeper/conf/zoo.cfg
cat >/opt/zookeeper/conf/zoo.cfg<<EOF
tickTime=2000
initLimit=10
syncLimit=5
dataDir=/data/zookeeper
clientPort=2181
server.1=10.0.0.51:2888:3888
server.2=10.0.0.52:2888:3888
server.3=10.0.0.53:2888:3888
EOF
```

```
echo "1" > /data/zookeeper/myid
cat /data/zookeeper/myid
scp -r /opt/zookeeper* 10.0.0.52:/opt/
scp -r /opt/zookeeper* 10.0.0.53:/opt/
```

#db52操作

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

#db53操作

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

3.所有节点启动zookeeper

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

4.每个节点都检查

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

5.测试zookeeper

在一个节点上执行,创建一个频道

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

在其他节点上看能否接收到

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

6.安装部署kafka

#db51操作

```
cd /data/soft/
tar zxf kafka_2.11-1.0.0.tgz -C /opt/
ln -s /opt/kafka_2.11-1.0.0/ /opt/kafka
mkdir /opt/kafka/logs
cat >/opt/kafka/config/server.properties<<EOF
```

```
broker.id=1
listeners=PLAINTEXT://10.0.0.51:9092
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
log.dirs=/opt/kafka/logs
num.partitions=1
num.recovery.threads.per.data.dir=1
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
log.retention.hours=24
log.segment.bytes=1073741824
log.retention.check.interval.ms=300000
zookeeper.connect=10.0.0.51:2181,10.0.0.52:2181,10.0.0.53:2181
zookeeper.connection.timeout.ms=6000
group.initial.rebalance.delay.ms=0
EOF
```

```
scp -r /opt/kafka* 10.0.0.52:/opt/
scp -r /opt/kafka* 10.0.0.53:/opt/
```

#db52操作

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

#db53操作

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

7.前台启动测试

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

8.验证进程

```
jps
```

9.测试创建topic

```
/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

```
/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

```
/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

```
/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

```
/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

```
/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

```
/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

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

14.修改filebeat配置文件

```
cat >/etc/filebeat/filebeat.yml << 'EOF'
filebeat.inputs:
- type: log
 enabled: true
 paths:
 - /var/log/nginx/access.log
 tags: ["access"]

- type: log
 enabled: true
 paths:
 - /var/log/nginx/error.log
 tags: ["error"]

output.kafka:
 hosts: ["10.0.0.51:9092", "10.0.0.52:9092", "10.0.0.53:9092"]
 topic: 'filebeat'

setup.ilm.enabled: false
setup.template.enabled: false
EOF
```

15.修改logstash配置文件

```
cat >/etc/logstash/conf.d/kafka.conf <<EOF
input {
 kafka{
  bootstrap_servers=>["10.0.0.51:9092,10.0.0.52:9092,10.0.0.53:9092"]
  topics=>["filebeat"]
  #group_id=>"logstash"
  codec => "json"
  }
 }

output {
 stdout {}
 if "access" in [tags] {
 elasticsearch {
 hosts => "http://10.0.0.51:9200"
 manage_template => false
 index => "nginx_access-%{+yyyy.MM}"
 }
 }
 if "error" in [tags] {
 elasticsearch {
 hosts => "http://10.0.0.51:9200"
 manage_template => false
 index => "nginx_error-%{+yyyy.MM}"
 }
 }
}
EOF
```

16.启动logstash并测试

#1.前台启动

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

#2.后台启动

```
systemctl start logstash
```