一、

解压logstash-7.2.1.zip

二、

/usr/app/logstash-7.2.1/bin/logstash-plugin install --version 5.0.0 logstash-codec-frame

/usr/app/logstash-7.2.1/bin/logstash-plugin install --version 6.4.3 logstash-filter-hashtree

三、对logstash进行测试：

/usr/app/logstash-7.2.1/bin/logstash -e 'input{stdin{}}output{stdout{codec=>rubydebug}}'

输入：hello world

输出：

{

"@timestamp" => 2019-08-21T05:53:43.671Z,

"host" => "pad",

"message" => "hello world",

"@version" => "1"

}

三、测试是否能写入elasticsearch：

1.docker start elasticsearch kibana

2. cp /usr/app/logstash-7.2.1/config/logstash.yml /usr/app/logstash-7.2.1/config/logstash.yml.bak

3.把环境变量写入/etc/profile

export SSL\_VERIFY=false

export HOSTNAME=pad

export TERM=xterm

export ELASTIC\_CONTAINER=true

export SSL\_CERT=/etc/certs/cert.pem

export LC\_ALL=en\_US.UTF-8

export COLLECTD\_PORT=25826

export HASHTREE\_METHOD=SHA1

export HASHTREE\_FILE=/usr/share/logstash/data/filter-hashtree

export SSL\_KEY=/etc/certs/key.pem

export SSL\_EXTRA\_CHAIN\_CERTS=/etc/certs/ca.pem

export HASHTREE\_TARGET=fingerprint

export ELASTICSEARCH\_HOST=elasticsearch

export PATH=/usr/share/logstash/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

export LANG=en\_US.UTF-8

export SYSLOG\_TLS\_PORT=6514

export SYSLOG\_UDP\_PORT=8514

export SHLVL=1

export HOME=/usr/app/logstash-7.2.1/

export COLLECTD\_BUFFER\_SIZE=1452

export ELASTICSEARCH\_PORT=9200

export HL7\_CHARSET=ISO-8859-1

export SYSLOG\_TCP\_PORT=8514

export HASHTREE\_PREVIOUS=fingerprint\_previous

source /etc/profile

还要把elasticsearch解析为本地ip。

4. vi logstash.conf

写入：

input{

stdin{}

}

output {

elasticsearch {

hosts => ["${ELASTICSEARCH\_HOST}:${ELASTICSEARCH\_PORT}"]

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

}

stdout{ codec => rubydebug }

}

5.启动logstash

/usr/app/logstash-7.2.1/bin/logstash -f /usr/app/logstash-7.2.1/config/logstash.conf

写入Hello World

登录kibanna：<http://172.16.0.19:5601>

可以发现有新的index-pattern。

同时/var/local/dcm4chee-arc/elasticsearch/nodes/0/indices多了东西。

进行测试：

curl -XDELETE 'http://localhost:9200/\_all'

curl -XGET localhost:9200/\_search

{"took":0,"timed\_out":false,"\_shards":{"total":0,"successful":0,"skipped":0,"failed":0},"hits":{"total":{"value":0,"relation":"eq"},"max\_score":0.0,"hits":[]}}[ro

在logstash的终端再输入一次Hello World

curl -XGET localhost:9200/\_search

{"took":10,"timed\_out":false,"\_shards":{"total":1,"successful":1,"skipped":0,"failed":0},"hits":{"total":{"value":1,"relation":"eq"},"max\_score":1.0,"hits":[{"\_index":"logstash-%{type}-2019.08.21","\_type":"\_doc","\_id":"\_xH9smwBZRWLLQLtqnbI","\_score":1.0,"\_source":{"host":"pad","message":"Hello World","@version":"1","@timestamp":"2019-08-21T07:04:24.232Z"}}]}}

表明logstash成功把数据传输到elasticsearch。

[root@pad translog]# pwd

/var/local/dcm4chee-arc/elasticsearch/nodes/0/indices/d8WQn5q2RQe\_Jb4XlvLkTA/0/translog

[root@pad translog]# cat translog-2.tlog

?translogMm82WIv\_QtyQQ7vy5cVbtg

表明logstash可以成功接入elasticsearch，则docker端的elasticsearch和kibana都是没问题的。

四、把容器里面的/etc/logstash.yml的内容写入到/usr/app/logstash-7.2.1/config/logstash.conf，在output那里加上stdout{ codec => rubydebug }。

/usr/app/logstash-7.2.1/bin/logstash -f /usr/app/logstash-7.2.1/config/logstash.conf

报错：

[2019-08-21T15:12:57,240][ERROR][logstash.agent ] Failed to execute action {:action=>LogStash::PipelineAction::Create/pipeline\_id:main, :exception=>"LogStash::ConfigurationError", :message=>"Cannot evaluate `${SSL\_KEY\_PASSWORD}`. Replacement variable `SSL\_KEY\_PASSWORD` is not defined in a Logstash secret store or as an Environment entry and there is no default value given.", :backtrace=>["/usr/app/logstash-7.2.1/logstash-core/lib/logstash/util/substitution\_variables.rb:50:in `block in replace\_placeholders'", "org/jruby/RubyString.java:3060:in `gsub'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/util/substitution\_variables.rb:35:in `replace\_placeholders'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/util/substitution\_variables.rb:23:in `deep\_replace'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/config/mixin.rb:83:in `block in config\_init'", "org/jruby/RubyHash.java:1419:in `each'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/config/mixin.rb:82:in `config\_init'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/inputs/base.rb:60:in `initialize'", "/usr/app/logstash-7.2.1/vendor/bundle/jruby/2.5.0/gems/logstash-input-tcp-6.0.2-java/lib/logstash/inputs/tcp.rb:122:in `initialize'", "org/logstash/plugins/PluginFactoryExt.java:258:in `plugin'", "org/logstash/plugins/PluginFactoryExt.java:120:in `buildInput'", "org/logstash/execution/JavaBasePipelineExt.java:50:in `initialize'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/java\_pipeline.rb:24:in `initialize'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/pipeline\_action/create.rb:36:in `execute'", "/usr/app/logstash-7.2.1/logstash-core/lib/logstash/agent.rb:325:in `block in converge\_state'"]}

然后自动退出。

把logstash.conf中的

tcp {

port => "${SYSLOG\_TLS\_PORT}"

codec => "frame"

type => "audit"

ssl\_enable => true

ssl\_cert => "${SSL\_CERT}"

ssl\_extra\_chain\_certs => "${SSL\_EXTRA\_CHAIN\_CERTS}"

ssl\_key => "${SSL\_KEY}"

ssl\_key\_passphrase => "${SSL\_KEY\_PASSWORD}"

ssl\_verify => "${SSL\_VERIFY}"

}

内容删掉。

再启动logstash：

/usr/app/logstash-7.2.1/bin/logstash -f /usr/app/logstash-7.2.1/config/logstash.conf

可以成功启动：

[2019-08-21T15:25:50,258][INFO ][logstash.javapipeline ] Starting pipeline {:pipeline\_id=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max\_inflight"=>125, :thread=>"#<Thread:0x2179cbf0 run>"}

[2019-08-21T15:25:52,218][INFO ][logstash.javapipeline ] Pipeline started {"pipeline.id"=>"main"}

[2019-08-21T15:25:52,245][INFO ][logstash.inputs.gelf ] Starting gelf listener (udp) ... {:address=>"0.0.0.0:12201"}

[2019-08-21T15:25:52,258][INFO ][logstash.inputs.tcp ] Starting tcp input listener {:address=>"0.0.0.0:8514", :ssl\_enable=>"false"}

[2019-08-21T15:25:52,625][INFO ][logstash.inputs.udp ] Starting UDP listener {:address=>"0.0.0.0:8514"}

[2019-08-21T15:25:52,671][INFO ][logstash.inputs.udp ] Starting UDP listener {:address=>"0.0.0.0:25826"}

[2019-08-21T15:25:52,878][INFO ][logstash.agent ] Pipelines running {:count=>1, :running\_pipelines=>[:main], :non\_running\_pipelines=>[]}

[2019-08-21T15:25:53,195][INFO ][logstash.inputs.udp ] UDP listener started {:address=>"0.0.0.0:8514", :receive\_buffer\_bytes=>"106496", :queue\_size=>"2000"}

[2019-08-21T15:25:53,253][INFO ][logstash.inputs.udp ] UDP listener started {:address=>"0.0.0.0:25826", :receive\_buffer\_bytes=>"106496", :queue\_size=>"2000"}

[2019-08-21T15:25:54,235][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}

五、启动dcm4chee，试图把日志传送到elk服务器的8514端口：

No such file or directory - /usr/share/logstash/data/filter-hashtree

[2019-08-21T15:49:38,751][ERROR][org.logstash.Logstash ] java.lang.IllegalStateException: Logstash stopped processing because of an error: (SystemExit) exit

[2019-08-21T15:49:38,754][ERROR][logstash.inputs.gelf ] Caught exception while reading from UDP socket {:exception=>#<Errno::EBADF: Bad file descriptor - No message available>}

[2019-08-21T15:49:38,761][ERROR][logstash.inputs.gelf ] Caught exception while reading from UDP socket {:exception=>#<IOError: closed stream>}

[2019-08-21T15:49:38,761][ERROR][logstash.inputs.gelf ] Caught exception while reading from UDP socket {:exception=>#<IOError: closed stream>}

发现报错。

解决办法：

mkdir /usr/app/logstash-7.2.1/data/filter-hashtree

把/etc/profile的HASHTREE\_FILE=/usr/share/logstash/data/filter-hashtree改成：

HASHTREE\_FILE=/usr/app/logstash-7.2.1/data/filter-hashtree

重新启动：

/usr/app/logstash-7.2.1/bin/logstash -f /usr/app/logstash-7.2.1/config/logstash.conf

报错：

org.jruby.exceptions.SystemCallError: (EISDIR) Is a directory - /usr/app/logstash-7.2.1/data/filter-hashtree

解决办法：

rm -rf /usr/app/logstash-7.2.1/data/filter-hashtree

touch /usr/app/logstash-7.2.1/data/filter-hashtree

chown -R logstash:logstash /usr/app/logstash-7.2.1/

重新启动：

/usr/app/logstash-7.2.1/bin/logstash -f /usr/app/logstash-7.2.1/config/logstash.conf

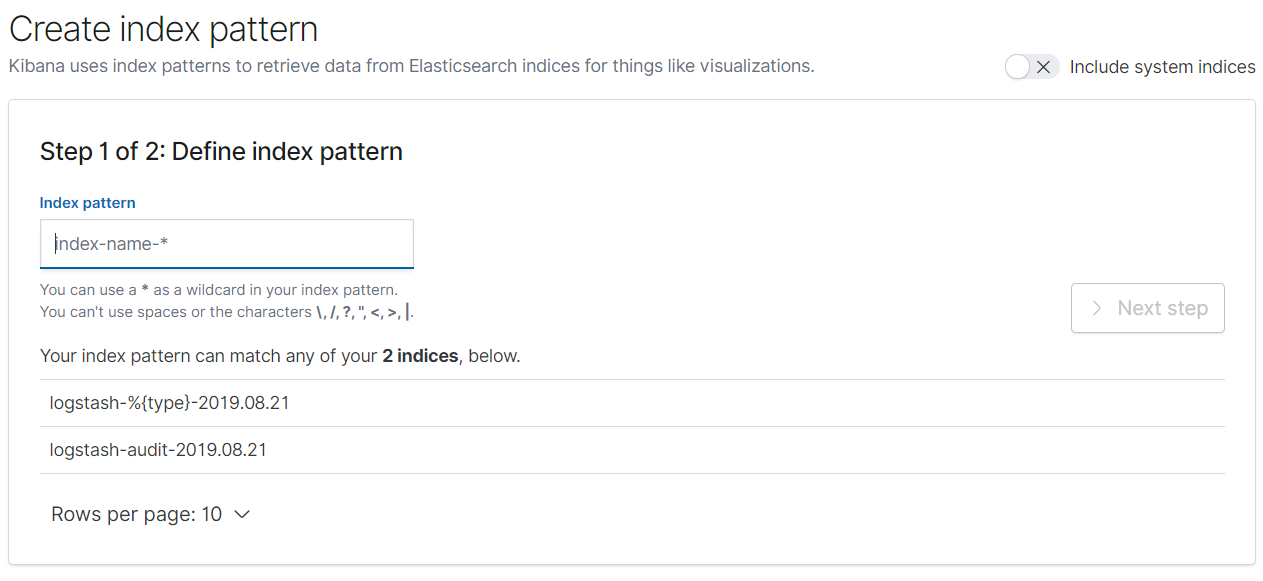
此时stdout没有报错，

ls /var/local/dcm4chee-arc/elasticsearch/nodes/0/indices

B\_WIlfvMTguRKFs7k\_rugA d8WQn5q2RQe\_Jb4XlvLkTA

多了东西。

Kibana上有了index pattern：



一开始添加新的index pattern不生效，而重启kibana和elasticsearch后生效了。

如有以下信息，可以对参数进行筛选：

