Set up Rsyslog client and server

on all vms you need to create files that rsyslog need to write it and get permission to user syslog in group adm -> syslog:adm

for simple config on each machine :

On client

## To create specific file to send all user command in it with specific syntax

- vim /etc/rsyslog.d/bash.conf :

```
local6.*    /var/log/commands.log
```

## In /etc/profile.d/test.sh : ------> in all vms that rsyslog in it

I. export PROMPT_COMMAND='RETRN_VAL=$?;logger -p local6.debug "$(whoami) [$$] [$PWD]: $(history 1 | sed "s/^[ ]*[0-9]\+[ ]*//") [$RETRN_VAL][$USER]"'

II. export PROMPT_COMMAND='RETRN_VAL=$?;logger -p local6.debug "[$$] command:[$(history 1 | sed "s/^[ ]*[0-9]\+[ ]*//")] => pwd: [$PWD] -> user: [$USER] [$RETRN_VAL]"' (final figure)

III. export PROMPT_COMMAND='RETRN_VAL=$?;logger -p local6.debug "[$$] command:[$(history 1 | sed "s/^[ ]*[0-9]\+[ ]*//"|sed y/\"/"_"/)] => pwd: [$PWD] -> user: [$USER] [$RETRN_VAL]"' (for fix bog -> echo “hi” (double quotes in echo command))

- vim /etc/rsyslog.d/client.conf :

```
# to connect client to server
*.*         @@192.168.119.92:10514
```

- TLS mode in rsyslog client to connect to rsyslog server :

  * apt install gnutls-bin | yum install -y gnutls-utils

- vim /etc/rsyslog.d/rsyslog-tls.conf :

```
# load package gnutls for this line with apt | yum (*)
```

```
$DefaultNetstreamDriver gtls

# certs
$DefaultNetstreamDriverCAFile /etc/ssl/rsyslog/CA.pem
$DefaultNetstreamDriverCertFile /etc/ssl/rsyslog/client-cert.pem
$DefaultNetstreamDriverKeyFile /etc/ssl/rsyslog/client-key.pem
$ActionSendStreamDriverAuthMode anon   # to authentication client and server
$ActionSendStreamDriverMode 1
$ActionSendStreamDriverPermittedPeer *     # send log to specific rsyslog server -> ip:port
*.* @@server:port         # usually is 6514 | 10514  (@@ -> tcp | @ udp)
```

On server

# Install packages you need :

```
apt install gnutls-bin rsyslog-elasticsearch
apt-get install rsyslog-gnutls
yum install -y gnutls-utils
```

** Note : in server vm you add ssh program name to config ssh for create log :

```
PROMPT_COMMAND='history -a >(tee -a ~/.bash_history | logger -t "$USER[$$]
$SSH_CONNECTION")'
```

- vim /etc/rsyslog.d/bash.conf :

```
local6.*    /var/log/commands.log
```

- vim /etc/rsyslog.d/server.conf :

```
# To create server in rsyslog
# Listen for TCP
$ModLoad imtcp
# Listen on port 514
$InputTCPServerRun 10514
$template RemoteServer, "/var/log/%HOSTNAME%/%SYSLOGFACILITY-TEXT%.log"
*.* ?RemoteServer
```

- TLS mode in rsyslog client to connect to rsyslog server :

* apt install gnutls-bin rsyslog-imptcp | yum install -y gnutls-utils

- vim /etc/rsyslog.d/rsyslog-tls.conf

```
# Add
$ModLoad imptcp -> need to install imptcp package
$ModLoad imtcp

$DefaultNetstreamDriver gtls

# certs
$DefaultNetstreamDriverCAFile /etc/ssl/rsyslog/CA.pem
$DefaultNetstreamDriverCertFile /etc/ssl/rsyslog/server-cert.pem
$DefaultNetstreamDriverKeyFile /etc/ssl/rsyslog/server-key.pem

#authentication
$InputTCPServerStreamDriverAuthMode anon
$InputTCPServerStreamDriverMode 1
$InputTCPServerStreamDriverPermittedPeer *
$InputPTCPServerRun 10514
```

** Note : if your client not connected to server , just check iptables rule to set tls port (like 10514) and in both server and client set * in flag { InputTCPServerStreamDriverPermittedPeer (in server) | ActionSendStreamDriverPermittedPeer (in client) }

- Install on client and server Vms for using elasticsearch

```
apt install rsyslog-elasticsearch
```

- vim /etc/rsyslog.d/00-elasticsearch.conf :

```
#set $.user=getenv("USER");

module(load="omelasticsearch")

template(name="plain-syslog" type="list" option.json="on") {
   constant(value="{")
   constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat="rfc3339")
   constant(value="\",\"host\":\"")        property(name="hostname")
   constant(value="\",\"severity-num\":")  property(name="syslogseverity")
   constant(value=",\"facility-num\":")    property(name="syslogfacility")
   constant(value=",\"severity\":\"")      property(name="syslogseverity-text")
   constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
#   constant(value="\",\"syslogtag\":\"")   property(name="syslogtag")
   constant(value="\",\"message\":\"")     property(name="msg")
   constant(value="\",\"name\":\"")        property(name="programname")
```

```
            #    constant(value="\",\"usenam\":\"")    property(name="$.user")
            #    constant(value="\",\"usenam\":\"")
            #    constant(value=`echo $USER`)
                constant(value="\"}")
            }

            template(name="rsyslog-index" type="string"
    string="rsyslog-%$YEAR%.%$MONTH%.%$DAY%")

            action(type="omelasticsearch"
              server="192.168.119.94"
              serverport="9200"
              template="plain-syslog"
              searchIndex="rsyslog-index"
              dynSearchIndex="on"
              bulkmode="on"
              maxbytes="100m"
              queue.type="linkedlist"
              queue.size="5000"
              queue.dequeuebatchsize="300"
              action.resumeretrycount="-1"
              errorfile="/var/log/omelasticsearch.log")
```

- After than you config rsyslog server you can define rule with iptables :

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 10514 -j ACCEPT
```

- If you need to see ssh and ansible and scp commands that run in all vms you can config rsyslog in specific logfile :

- vim /etc/rsyslog.d/ssh.conf :

```
        local3.*    /var/log/sshd.log
        if $programname == 'sshd' then /var/log/sshd.log
```

and in /etc/ssh/sshd_config uncomment this sec :

```
# Logging
#SyslogFacility AUTH
SyslogFacility local3
```

```
      #LogLevel INFO
      LogLevel DEBUG3 (debug3 is the all of logs in ssh service like sessions and commands that run in there )
```

On another server create elk stack without logstash (with optional ssl)

- vim $pwd/docker-compose.yml :

```
version: "2.2"

services:
  create_certs:
    container_name: create_certs
    image: docker.elastic.co/elasticsearch/elasticsearch:7.10.2
    command: >
      bash -c '
        if [[ ! -f /certs/bundle.zip ]]; then
          bin/elasticsearch-certutil cert --silent --pem --in config/certificates/instances.yml -out /certs/bundle.zip;
          unzip /certs/bundle.zip -d /certs;
        fi;
        chown -R 1000:0 /certs
      '
    user: "0"
    working_dir: /usr/share/elasticsearch
    volumes: ['certs:/certs', 'cert-elasticsearch:/usr/share/elasticsearch/config/certificates']
    networks:
      - elastic

  elasticsearch:
    container_name: elasticsearch
    image: docker.elastic.co/elasticsearch/elasticsearch:7.10.2
    restart: unless-stopped
    ports:
      - "9200:9200"
      - "9300:9300"
    networks:
      - elastic
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro
      - certs:/usr/share/elasticsearch/config/certificates
    environment:
      - discovery.type=single-node
      - ES_JAVA_OPTS=-Xms512m -Xmx1g
      - ELASTIC_PASSWORD=elastic
      - xpack.license.self_generated.type=trial
      - xpack.security.enabled=true
      - xpack.security.http.ssl.enabled=true
      - xpack.security.http.ssl.key=/usr/share/elasticsearch/config/certificates/es/es.key
      - xpack.security.http.ssl.certificate_authorities=/usr/share/elasticsearch/config/certificates/ca/ca.crt
      - xpack.security.http.ssl.certificate=/usr/share/elasticsearch/config/certificates/es/es.crt
      - xpack.security.transport.ssl.enabled=true
      - xpack.security.transport.ssl.verification_mode=certificate
      -
xpack.security.transport.ssl.certificate_authorities=/usr/share/elasticsearch/config/certificates/ca/ca.crt
      - xpack.security.transport.ssl.certificate=/usr/share/elasticsearch/config/certificates/es/es.crt
      - xpack.security.transport.ssl.key=/usr/share/elasticsearch/config/certificates/es/es.key

    healthcheck:
      test: curl --cacert /usr/share/elasticsearch/config/certificates/ca/ca.crt -s https://localhost:9200
>/dev/null; if [[ $$? == 52 ]]; then echo 0; else echo 1; fi
      interval: 30s
      timeout: 10s
      retries: 5
    #cpu_shares: 10
    #cpu_quota: 50000
    cpuset: 0,1,2
    mem_limit: 1g
    memswap_limit: 2g
    mem_reservation: 512m
    #shm_size: 64M

  kibana:
    container_name: kibana
    image: kibana:7.10.1
    depends_on: {"elasticsearch": {"condition": "service_healthy"}}
    restart: unless-stopped
    ports:
      - "5601:5601"
    networks:
      - elastic
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro
      - /srv/db/monitoring/kibana/config/kibana.yml:/usr/share/kibana/config/kibana.yml
      - certs:/usr/share/elasticsearch/config/certificates
    environment:
      SERVERNAME: localhost
      ELASTICSEARCH_URL: https://elasticsearch:9200
      ELASTICSEARCH_HOSTS: https://elasticsearch:9200
      ELASTICSEARCH_USERNAME: elastic
      ELASTICSEARCH_PASSWORD: elastic
      ELASTICSEARCH_SSL_CERTIFICATEAUTHORITIES: /usr/share/elasticsearch/config/certificates/ca/ca.crt
      SERVER_SSL_ENABLED: "true"
      SERVER_SSL_KEY: /usr/share/elasticsearch/config/certificates/kibana/kibana.key
      SERVER_SSL_CERTIFICATE: /usr/share/elasticsearch/config/certificates/kibana/kibana.crt
    cpuset: 0,1,2
    mem_limit: 1g
    memswap_limit: 2g
    mem_reservation: 512m
    #shm_size: 64M

volumes: {"certs", "cert-elasticsearch"}
networks:
  elastic:
    driver: bridge
```

- NOTE : in cert-elasticsearch volume :

```
touch indices.yml :
            instances:
               - name: es
                 dns:
                     - es
                     - elasticsearch
                     - 192.168.119.107
                     - localhost
                 ip:
                     - 127.0.0.1

               - name: kibana
                 dns:
```

```
    - kib
    - kibana
    - 192.168.119.107
    - localhost
ip:
  - 127.0.0.1

  - name: vm-1
    dns:
        - vm-1
        - 192.168.119.104
    ip:
       - 192.168.119.104
```

On server with specific index for one programname like ssh (conditional rsyslog’s logs)

vim /etc/rsyslog.d/00-elasticsearch.conf :

```
          module(load="omelasticsearch")
          module(load="mmnormalize")

          #set $.my_user=getenv("USER");

          template(name="getuser" type="string" string="%msg:R,ERE,2,DFLT:(user:)(\\\"|[^\"]*)->--end%")

          #template(name="getuser" type="string" string="%msg::$USER%")
          set $!my_user = exec_template("getuser");


template(name="plain-syslog" type="list" option.json="on") {
             constant(value="{")
              constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat="rfc3339")
              constant(value="\",\"host\":\"")        property(name="hostname")
              constant(value="\",\"severity-num\":")  property(name="syslogseverity")
              constant(value=",\"facility-num\":")    property(name="syslogfacility")
              constant(value=",\"severity\":\"")      property(name="syslogseverity-text")
              constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
          #    constant(value="\",\"syslogtag\":\"")   property(name="syslogtag")
              constant(value="\",\"message\":\"")     property(name="msg")
          constant(value="\",\"name\":\"")       property(name="programname")
          constant(value="\",\"username-system\":\"")    property(name="$!my_user")
        #    constant(value="\",\"username-rsyslog\":\"")
        #    constant(value=`echo $user`)
          constant(value="\"}")
}

template(name="rsyslog-index" type="string" string="rsyslog2-%$YEAR%.%$MONTH%.%$DAY%")

#action(type="omelasticsearch"
#  server="192.168.119.94"
#  serverport="9200"
#  template="plain-syslog"
#  searchIndex="rsyslog-index"
#  dynSearchIndex="on"
#  bulkmode="on"
#  maxbytes="100m"
#  uid=`echo $ES_USER`
#  pwd=`echo $ES_PASSWORD`
#  queue.type="linkedlist"
#  queue.size="5000"
#  queue.dequeuebatchsize="300"
#  action.resumeretrycount="-1"
#  errorfile="/var/log/omelasticsearch.log")


template(name="extract" type="string" string="%msg:R,ERE,2,DFLT:(user:)(\\\"|[^\"]*)->--end%")
#template(name="getuser" type="string" string="%msg::$USER%")
set $!my_user = exec_template("extract");


template(name="ssh-syslog" type="list" option.json="on") {
  constant(value="{")
  constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat="rfc3339")
  constant(value="\",\"host\":\"")        property(name="hostname")
  constant(value="\",\"severity-num\":")  property(name="syslogseverity")
  constant(value=",\"facility-num\":")    property(name="syslogfacility")
  constant(value=",\"severity\":\"")      property(name="syslogseverity-text")
  constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
#    constant(value="\",\"syslogtag\":\"")   property(name="syslogtag")
  constant(value="\",\"message\":\"")     property(name="msg")
  constant(value="\",\"name\":\"")        property(name="programname")
  constant(value="\",\"username-system\":\"")    property(name="$!my_user")
#    constant(value="\",\"username-rsyslog\":\"")
#    constant(value=`echo $user`)
    constant(value="\"}")
}

template(name="ssh-index" type="string" string="ssh-%$YEAR%.%$MONTH%.%$DAY%")

if $programname == 'sshd' then{
action(type="omelasticsearch"
  server="192.168.119.94"
  serverport="9200"
  template="ssh-syslog"
  searchIndex="ssh-index"
  dynSearchIndex="on"
  bulkmode="on"
  maxbytes="100m"
  uid=`echo $ES_USER`
  queue.type="linkedlist"
  queue.size="5000"
  queue.dequeuebatchsize="300"
  action.resumeretrycount="-1"
  errorfile="/var/log/omelasticsearch-ssh.log")
}else{
action(type="omelasticsearch"
  server="192.168.119.94"
  serverport="9200"
  template="plain-syslog"
  searchIndex="rsyslog-index"
  dynSearchIndex="on"
  bulkmode="on"
  maxbytes="100m"
  uid=`echo $ES_USER`
  queue.type="linkedlist"
  queue.size="5000"
  queue.dequeuebatchsize="300"
  action.resumeretrycount="-1"
  errorfile="/var/log/omelasticsearch.log")
}
```

## Note : after rsyslog config you should define env in bashrc :

Vim /etc/bash.bashrc :

```
        HISTTIMEFORMAT="%Y-%m-%d:%H-%M-%S: user:$USER "
```

```
export HISTTIMEFORMAT

source /etc/bash.bashrc
```

Write Template in rsyslog

in /etc/rsyslog.d/00-elasticsearch :

```
module(load="omelasticsearch")
module(load="mmnormalize")

#set $.my_user=getenv("USER");

template(name="getuser" type="string" string="%msg:R,ERE,2,DFLT:(user: )(\\\"|[^\"]*) --end%")
#template(name="getuser" type="string" string="%msg::$USER%")
set $!my_user = exec_template("getuser");


template(name="getpwd" type="string" string="%msg:R,ERE,2,DFLT:(pwd: )(\\\"|[^\"]*)->--end%")
set $!my_pwd = exec_template("getpwd");


template(name="getcommand" type="string"
string="%msg:R,ERE,2,DFLT:(command:)(\\\"|[^\"]*)=>--end%") set $!my_command =
exec_template("getcommand");

template(name="plain-syslog" type="list" option.json="on") {
   constant(value="{")
   constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat="rfc3339")
   constant(value="\",\"host\":\"")        property(name="hostname")
   constant(value="\",\"severity-num\":")  property(name="syslogseverity")
   constant(value=",\"facility-num\":")    property(name="syslogfacility")
   constant(value=",\"severity\":\"")      property(name="syslogseverity-text")
   constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
#    constant(value="\",\"syslogtag\":\"")   property(name="syslogtag")
   constant(value="\",\"message\":\"")     property(name="msg")
   constant(value="\",\"name\":\"")        property(name="programname")
   constant(value="\",\"username-system\":\"")    property(name="$!my_user")
   constant(value="\",\"pwd\":\"")    property(name="$!my_pwd")
   constant(value="\",\"command\":\"")    property(name="$!my_command")
#    constant(value="\",\"username-rsyslog\":\"")
#    constant(value=`echo $user`)
   constant(value="\"}")
}

template(name="rsyslog-index" type="string" string="rsyslog2-%$YEAR%.%$MONTH%.%$DAY%")

action(type="omelasticsearch"
  server="192.168.119.107"
  serverport="9200"
  template="plain-syslog"
  searchIndex="rsyslog-index"
  dynSearchIndex="on"
  bulkmode="on"
  maxbytes="100m"
  uid=`echo $ES_USER`
  queue.type="linkedlist"
  queue.size="5000"
  queue.dequeuebatchsize="300"
  action.resumeretrycount="-1"
  errorfile="/var/log/omelasticsearch.log")
```

Https connection between elasticsearch and rsyslog server

در مسیر /var/lib/docker/volume/certs/_data/ گواهی های تولید شده توسط الاستیک در داکر کامپوز را از ماشین دریافت نموده و در
مسیر /etc/local/share/ca-certificate در ماشین rsyslog server قرار میدهیم و سپس دستور زیر را اجرا میکنیم :

```
update-ca-certificates
```

سپس در کانفیگ ماژول ارسال کننده لاگ ها به الاستیک (/etc/rsyslog.d/00-elasticsearch.conf) در تابع اجرایی action گزینه
های زیر را اضافه مینماییم :

```
usehttps="on"
cert="/usr/local/share/ca-certificates/ca.crt"
pwd=<ES_PASSWORD>
uid=<ES_USERNAME>
server=<dns_name or ip that set in instances.yml in elasticsearch docker-compose>
```

Ssh filtering in rsyslog and send to elasticsearch index

in /etc/rsyslog.d/00-elasticsearch.conf :

```
if $programname == 'sshd' then
{
     if $msg contains 'Starting' then {
        set $!start = $msg;
    } else if $msg contains 'debug3: mm_audit_run_command' then {
        set $!run = $msg;
    } else if $msg contains 'Disconnected' then {
        set $!disconnect = $msg;
    } else if $msg contains 'Failed password' then {
        set $!failed = $msg;
    }
}
```

and in template plain-rsyslog in this file add below line :

```
template(name="plain-syslog" type="list" option.json="on") {
    ...
    .
    .
    .
    ...
    constant(value="\",\"start-session\":\"")    property(name="$!start")
    constant(value="\",\"disconnect-session\":\"")    property(name="$!disconnect")
    constant(value="\",\"run_command\":\"")    property(name="$!run")
    constant(value="\",\"failed\":\"")    property(name="$!failed")

}
```