ELK Stack SOP

Installation process

Create a user other than root user for the installation of elk stack. Below are the steps to create user.

# adduser elk

# passwd elk

Enter the password of your choice.

Change the ownership of /opt directory with the below command.

#mkdir –p /opt/elk

# chmod 775 /opt/elk -R

# chown elk:elk /opt/elk –R

**Filebeat**

* **Installation**

Use the below steps to install the filebeat on the application machine (From where the logs will be exported).

$ cd /opt

$ mkdir filebeat

$ cd filebeat$ curl -L -O <https://download.elastic.co/beats/filebeat/filebeat-1.1.2-x86_64.rpm>

$ rpm -ivh filebeat-1.1.2-x86\_64.rpm

* **Configuration**

$ vi /etc/filebeat/filebeat.yml

Configuration to read the file as input-

Log path and document type will be configuring in this file.

filebeat:

# List of prospectors to fetch data.

prospectors:

# Each - is a prospector. Below are the prospector specific configurations

-

# Paths that should be crawled and fetched. Glob based paths.

# For each file found under this path, a harvester is started.

paths:

- "/var/log/\*.log"

#- c:\programdata\elasticsearch\logs\\*

# Type of the files. Based on this the way the file is read is decided.

# The different types cannot be mixed in one prospector

#

# Possible options are:

# \* log: Reads every line of the log file (default)

# \* stdin: Reads the standard in

input\_type: log

document\_type: online

Configuration of filebeat output for logstash

output:

logstash:

hosts: ["10.5.205.233:5032"]

* **Start service**

Use the below command to start the service.

$ /opt/filebeat/bin/filebeat –c /etc/filebeat/filebeat.yml

* **Stop Service**

To stop the service first check the process id of that process with the ‘ps –ef’ command then kill the process using that process id.

$ ps –ef | grep filebeat

elk 16955 1 0 Mar02 ? 00:43:17 /usr/java/jdk1.8.0\_65//bin/java

$ kill -9 ‘process\_id’

* **Verify service**

Use the below command to verify the service is running.

$ ps –ef | grep filebeat

* **Running in background**

Use the below command to run the service in background.

$ nohup /opt/filebeat/bin/filebeat –c /etc/filebeat/filebeat.yml &

**Logstash**

* **Installation**

Download and install the public signing key:

$ rpm --import <https://packages.elastic.co/GPG-KEY-elasticsearch>

Add the following in your /etc/yum.repos.d/ directory in a file with a .repo suffix, for example logstash.repo

[logstash-2.1]

name=Logstash repository for 2.1.x packages

baseurl=http://packages.elastic.co/logstash/2.1/centos

gpgcheck=1

gpgkey=http://packages.elastic.co/GPG-KEY-elasticsearch

enabled=1

And your repository is ready for use. You can install it with:

$ yum install logstash

* **Configuration**

The configuration of the logstash file should be as below:

User have to configure the pattern as per the requirement.

input {

beat {

port => 5032

}

filter {

grok {

break\_on\_match => false

patterns\_dir => "./patterns"

}

output {

elasticsearch { hosts => ["10.5.205.233:9200","10.5.205.232:9201"]

index => "logstash-online"

}

stdout { codec => rubydebug }

}

* **Start service**

Use the below command to start the service.

$ /opt/logstash/bin/logstash –f /etc/logstash/conf.d/online\_logstash.conf

* **Stop Service**

To stop the service first check the process id of that process with the ‘ps –ef’ command then kill the process using that process id.

$ ps –ef | grep logstash

elk 16955 1 0 Mar02 ? 00:43:17 /usr/java/jdk1.8.0\_65//bin/java

$ Kill -9 “process id”

* **Verify service**

Use the below command to verify the service is running.

$ ps –ef | grep logstash

* **Running in background**
* Use the below command to run the service in background.

$ nohup /opt/logstash/bin/logstash –f /etc/logstash/conf.d/online\_logstash.conf &

**Elasticsearch**

* **Installation**

$ mkdir /opt/elasticsearch

Let’s download the Elasticsearch 2.2.0 tar as follows:

$ curl -L –O https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/tar/elasticsearch/2.2.0/elasticsearch-2.2.0.tar.gz

Then extract it as follows:

$ tar -xvf elasticsearch-2.2.0.tar.gz

* **Configuration**

$ vi /opt/elasticsearch/elasticsearch/config/elasticsearch.yml

Add the below lines in the elasticsearch.yml file:

# Set the bind address to a specific IP (IPv4 or IPv6):

network.host: 10.5.205.233

# Set a custom port for HTTP:

http.port: 9200

* **Start service**

Use the below command to start the service.

$ cd /opt/elasticsearch/elasticsearch/bin

$ ./elasticsearch

* **Stop Service**

To stop the service first check the process id of that process with the ‘ps –ef’ command then kill the process using that process id.

$ ps –ef | grep elasticsearch

elk 16955 1 0 Mar02 ? 00:43:17 /usr/java/jdk1.8.0\_65//bin/java

$ Kill -9 “process id”

* **Verify service**

Use the below command to verify the service is running.

$ ps –ef | grep elasticsearch

* **Running in background**

Use the below command to run the service in background.

$ nohup ./elasticsearch &

* **Cluster Configuration**

Please make the configuration of nodes, cluster name and other as we have in the attached file.



# ======================== Elasticsearch Configuration =========================

#

# NOTE: Elasticsearch comes with reasonable defaults for most settings.

# Before you set out to tweak and tune the configuration, make sure you

# understand what are you trying to accomplish and the consequences.

#

# The primary way of configuring a node is via this file. This template lists

# the most important settings you may want to configure for a production cluster.

#

# Please see the documentation for further information on configuration options:

# <http://www.elastic.co/guide/en/elasticsearch/reference/current/setup-configuration.html>

#

# ---------------------------------- Cluster -----------------------------------

#

# Use a descriptive name for your cluster:

#

cluster.name: online-prod

#

# ------------------------------------ Node ------------------------------------

#

# Use a descriptive name for the node:

#

node.name: node-2

node.master: true

node.data: true

#

# Add custom attributes to the node:

#

# node.rack: r1

#

# ----------------------------------- Paths ------------------------------------

#

# Path to directory where to store the data (separate multiple locations by comma):

#

path.data: /online01/data

#

# Path to log files:

#

path.logs: /online01/logs

#

# ----------------------------------- Memory -----------------------------------

#

# Lock the memory on startup:

#

# bootstrap.mlockall: true

#

# Make sure that the `ES\_HEAP\_SIZE` environment variable is set to about half the memory

# available on the system and that the owner of the process is allowed to use this limit.

#

# Elasticsearch performs poorly when the system is swapping the memory.

#

# ---------------------------------- Network -----------------------------------

#

# Set the bind address to a specific IP (IPv4 or IPv6):

#

network.host: 10.5.207.161

#

# Set a custom port for HTTP:

#

http.port: 9200

http.cors.allow-origin : "\*"

http.cors.allow-methods : OPTIONS, HEAD, GET, POST, PUT, DELETE

http.cors.allow-headers : X-Requested-With,X-Auth-Token,Content-Type, Content-Length

#

# For more information, see the documentation at:

# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.html>

#

# --------------------------------- Discovery ----------------------------------

#

# Pass an initial list of hosts to perform discovery when new node is started:

# The default list of hosts is ["127.0.0.1", "[::1]"]

#

# discovery.zen.ping.unicast.hosts: ["host1", "host2"]

#

discovery.zen.ping.multicast.enabled: false

discovery.zen.ping.unicast.hosts: ["10.5.207.160", "10.5.207.161", "10.5.207.162", "10.5.207.163", "10.5.207.164"]

# Prevent the "split brain" by configuring the majority of nodes (total number of nodes / 2 + 1):

#

discovery.zen.minimum\_master\_nodes: 1

**Kibana**

* **Installation**

Follow the below steps to install Kibana on server.

$ mkdir /opt/kibana

$ curl -L -O <https://download.elastic.co/kibana/kibana/kibana-4.4.1-linux-x64.tar.gz>

$ tar –xvf kibana-4.4.1-linux-x64.tar.gz

$ cd /opt/kibana/ kibana-4.4.1

* **Configuration**

$ vi /opt/kibana/config/kibana.yml

Configure the below lines in kibana.yml file

# Kibana is served by a back end server. This controls which port to use.

server.port: 5601

# The host to bind the server to.

server.host: "10.5.205.233"

* **Start service**

Use the below command to start the service.

$ cd /opt/kibana/kibana-4.4.1/bin

$ ./kibana

* **Stop Service**

To stop the service first check the process id of that process with the ‘ps –ef’ command then kill the process using that process id.

$ ps –ef | grep cli

elk 16955 1 0 Mar02 ? 00:43:17 /usr/java/jdk1.8.0\_65//bin/java

$ kill -9 “process id”

* **Verify service**

Use the below command to verify the service is running.

$ps –ef | grep cli

* **Running in background**

Use the below command to run the service in background.

$ nohup ./kibana &

**ElastAlerts**

* **Installation**

$ mkdir –p /opt/elastalert

$ git clone <https://github.com/Yelp/elastalert.git>

$ python setup.py install

$ pip install -r requirements.txt

* **Configuration**

Seting up Elasticsearch

$ elastalert-create-index

To create the rule, edit the below file

$ vi example\_rules/example\_frequency.yaml

# Elasticsearch host

es\_host: 10.5.205.233

# Elasticsearch port

es\_port: 9200

es\_username: elk

es\_password: elk@123#

name: Online\_1 rule

type: any

index: logstash-\*

num\_events: 1

timeframe:

hours: 1

filter:

- term:

App\_name: TELEMEDIA

and

Transaction\_status.raw: Failure

alert:

- "email"

email: [abc@airtel.com](mailto:abc@airtel.com)

Test the rule with below command

$ elastalert-test-rule example\_rules/example\_frequency.yaml

* **Start service**

Use the below command to start the service.

$ /usr/bin/elastalert --verbose --rule /root/elastalert/elastalert/example\_rules/example\_frequency.yaml --config /root/elastalert/elastalert/config.yaml.example

* **Stop Service**

To stop the service first check the process id of that process with the ‘ps –ef’ command then kill the process using that process id.

$ ps –ef | grep elastalert

elk 16955 1 0 Mar02 ? 00:43:17 /usr/java/jdk1.8.0\_65//bin/java

$ kill -9 “process\_id”

* **Verify service**

Use the below command to verify the service is running.

$ ps –ef | grep elastalert

* **Running in background**

Use the below command to run the service in background.

$ /usr/bin/elastalert --verbose --rule /root/elastalert/elastalert/example\_rules/example\_frequency.yaml --config /root/elastalert/elastalert/config.yaml.example &

**Additional Components:**

**Run Jmeter**

To run the jmeter, we need the GUI access which we can take with the help of vnc server. These are the below commands use to start the vnc server.

First we will set the password with the **vncpasswd** command.

#vncpasswd

Password:

Confirm

Then start vncserver with the vncserver command.

#vncserver

Then take the console on your machine vnc viewer client.

To run the jmeter go to the bin directory and run the jmeter.jar.

# cd /opt/jmeter/bin

# java –jar jmeter.jar

**Nmon**

Run the nmon command to monitor the server performance.

# nmon –f 15 –c 240 /data1/

**Curator**

* **Installation**

The curator installs with the help of pip command. Please find the below command to install it.

# pip install elasticsearch-curator

* **Setting the Cronjob**

We need to set the cronjob to configure the curator to delete the indices older than the 15 days. Below is the string we configured in the crontab file.

20 00 \* \* \* /usr/bin/curator --host 10.5.205.233 delete indices --older-than 15 --time-unit days --timestring '%Y.%m.%d'