Log Monitoring using ELK Stack

(Log Monitoring using the ELK Stack)

The ELK Stack is an open-source log analytics solution with three software components: Elasticsearch, Logstash, and Kibana. Working together, these technologies allow DevOps and SecOps teams to collect, aggregate, analyze, and visualize log data in the cloud, supporting critical functions like application monitoring and security analytics.

What is difference b/w Logging tools like log4j, slf4j and etc.. and ELK stack?

Ans) Logging tools generate the Log messages and writes them to log files where as ELK stack is Log Monitoring Tool who actually monitors the log messages by collecting from different log files having indexing and Visualization

ELK stack is the combination of 3 tools

It is a NO SQL DB s/w which can store the Application generated log messages by collecting from original log files (text files) :: It is a pipe line tool that takes log messages from different sources (like Apps)

E--> Elestic Search

::

L ---> Logstash

K ---> Kibana

and exports them to different targets (likestic serving the indexing

:: It is a Visualization UI Layer, which helps the developer to monitor application logs browser env.. having GUI screens

What are different Log monitoring Tools available in the market? Ans) ELK stack, Splunk and etc..

Java /Non-Java App (S)

(any Technology App)

ELK stack is open source stack becoz

Elasticsearch, LogStash and kibana tools are

open source tools

splunk> Splunk

Datadog

DATADOG

graylog Graylog

LogsHero Ltd.

logz.io

Su mo

Sumo Logic

Fluentd

Fluentr

uses logging api of certain domain

log file(s)

LogStash

(b)

(c)

Elestic Search

(a)

(NO SQL DB)

note:: ELK statck very useful to collect the log messages from different log files same or different apps of any technology and

Logstash

. . . . . .

to summarize and aggragate those log message ..

CloudPhysics

(a) Ap

generates the log messages using its logging api like log4j/slf4j and etc.. and writes them to log file(s)

(b) Logstash tool takes the log messages from the log file s and process them having indexing

(c) Logstash processed messages will be stored in Elestic search Non SQL DB s/w

(d) these stored messages will be passed to kibana

(d)

ELK stack can be used in java, .NET, php, js, python and etc.. technology Apps to enable analytics on the log messages.

Grafana Loki

Grafanc

Kibana

ELK is not alternate to any logging api like log4j,slf4j and etc.. Infact it compliments those logging apis by collecting their log messages for aggregation and analyzation

Dynatrace

to

tool for UI env.. to see and use the log messages

ELK stack tools are top level tool enable the analytics on the

log messages that are generated by any technology app using specific logging api.

note: Java App generated log4j/slf4j based log messages can be analyzed displayed for end users using the support of ELK

(DevOps)

Dataset

ELK Stack Use Cases

When deployed together, Elasticsearch, Logstash and Kibana work as a search and analytics engine, allowing IT operations teams to:

->Aggregate collected log data from a variety of sources using Logstash. ->Transform, process, and enrich log data using Logstash and Elasticsearch.

->Index and search log data using Elasticsearch.

->Explore and analyze log data, and produce data visualizations using Kibana.

Common use cases for the ELK stack include:

Log Analysis and Monitoring: ELK is frequently used for centralizing and analyzing log data from various sources such as servers, applications, and network devices. It allows organizations to gain insights into system health, performance, and potential issues, helping to troubleshoot problems and maintain system reliability.

Application Performance Monitoring (APM): With the Elastic APM module, the ELK Stack can be used to monitor the performance of applications. It enables developers and operations teams to track and analyze application behavior, identify bottlenecks, and optimize code performance.

Security Monitoring: ELK can be utilized to monitor security events, collecting and analyzing security-related data from multiple sources to detect and respond to security threats, suspicious activities, and cyberattacks in real-time.

Business Intelligence and Data Analytics: Organizations can use ELK to analyze large datasets, generate insights, and build interactive dashboards and visualizations for business intelligence purposes. This information helps in data-driven decision-making and identifying trends and patterns.

Infrastructure Monitoring: Many teams use ELK for monitoring the performance and health of IT infrastructure, including servers, databases, networks, and cloud resources. It enables proactive monitoring and alerting based on predefined thresholds.

DevOps Monitoring: For DevOps teams, ELK can be used to consolidate logs and metrics from various development and deployment tools, facilitating collaboration, and providing insights into the software development lifecycle.

Compliance and Auditing: The ELK Stack can help organizations meet regulatory compliance requirements by collecting and retaining relevant data for audit trails and reporting, although retaining data in Elasticsearch can be costly

When and where should i use ELK stack?

Ans) use ELK stack on the top of existing logging apis if the app is generating huge amount of log messages from different instance of the app by taking the support huge variety no.of log files. ELK stack is useful

to collect and consolidate the huge no.of log messages coming from the muliple log files in a single place.

cental logging using ELK Statck Tools (Performing Log Monitoring)

Example App on

of

=>The 3 softwares of ELK stack can be downloaded in the form of zip files

step1) Download and keep all the 3 tools ELK stack ready

a) Kibana download URL

x

https://www.elastic.co/downloads/kibana

Choose platform:

Windows

✓ Windows

sha asc

b) elestic search download

elastic.co/downloads/elasticsearch

kibana-7.17.0-windows-x86_64

Download Elasticsearch

1 Download and unzip Elasticsearch

→

C

=>Since the 8.x versions of ELK stack softwares are having few technical issues, So better to use the 7.x versions of the ELK stack

Choose platform:

Windows

www. Downloads >

Name

Type

✓ Today

elasticsearch-7.17.0-windows-x86_64 (4)

✓ **Windows**

sha asc

c) Log stash

elastic.co/downloads/logstash

1 Download and unzip Logstash

Choose platform:

Windows

✓ Today

✓ **Windows**

shaasc

www.logstash-7.17.0-windows-x86_64

step2) Try to start kibana and elestic search servers by extracting them

(a) edit kibana.yml file from <kibana_home>\config fodler

kibana.yml

 # elasticsearch.hosts: ["http://localhost:9200"] (this is nothing but (uncomment this line)

⇧

linking elastic search port number server with kibana server)

of Elastic Search

bin

b) start elastic search server

i) use <elstaticsearch_home>\elasticsearch.bat file (Run as adminstrator) (takes few minutes to start)

ii) see the confirmation screen in the browser

http://localhost:9200

(upto 7 version, no need of giving usename and password)

← C localhost:9200

{

}

"name": "DESKTOP-QENT2RN",

"cluster_name": "elasticsearch",

"cluster_uuid": "4JWUkwiRSgik4fGmL5bsjg",

▼ "version": {

"number": "8.10.2",

"build_flavor": "default",

"build_type": "zip",

"build_hash": "6d20dd8ce62365be9b1aca96427de4622e970e9e",

"build_date":"2023-09-19T08:16:24.564900370Z",

"build_snapshot": false,

"lucene_version": "9.7.0",

"minimum_wire_compatibility_version": "7.17.0",

"minimum_index_compatibility_version": "7.0.0"

},

"tagline": "You Know, for Search"

ii) start the kibana Server

=>use <kibana_home>\bin\kibana.bat (Run as admin) (takes few minutes time to start)

=> observe the kibana home page

oralhost:3300

←

elastic

http://localhost:5601

Frowne integrations Integration

x

+

localhost:5501/app/integrations/browse

Integrations Browse Integrations

Q Search Elast

Integrations

Choose an integration to start collecting and analyzing your data.

Browse integrations

Installed integrations

Web site crawler

Add search to your website with the App Search web crawler.

Elastic APM Monitor, detect and diagnose complex performance issues from your application.

step3) Develope the regualar Java App (s)

and run that application for multiple times...

//SelectTest.java

App1

package com.nt.service;

import java.sql.Connection;

====

having slf4j log messages..

Create the following env.. variable before starting the kibana server

Edit System Variable

Variable name:

JAVA_HOME

Variable value:

C:\Program Files\Java\jdk-17

Browse Directory...

Browse File...

Edit System Variable

Variable name:

LS_JAVA_HOME

Variable value:

C:\Program Files\Java\jdk-17

Browse Directory...

Browse File...

log4j.properties

 # For All log messages

log4j.rootLogger=ALL,R

log4j.appender.R=org.apache.log4j.FileAppender

log4j.appender.R.File= D:/jdbc_elk_emp.txt

log4j.appender.R.Append=true

log4j.appender.R.Immediate Flush=true

log4j.appender.R.layout=org.apache.log4j.PatternLayout

log4j.appender.R.layout.ConversionPattern=%p- %t- %r-%c-%m-%d-% -%n

import java.sql.DriverManager;

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class SelectTest {
private static final String GET_ALL_EMPS="SELECT EMPNO,ENAME,JOB,SAL, DEPTNO FROM EMP";
//create Logger object
private static Logger logger-LoggerFactory.getLogger(SelectTest.class);
public static void main(String[] args) {
logger.debug("start of main(-) method");
Connection con=null;
PreparedStatement ps=null;
ResultSet rs=null;
try {
logger.debug("start of try block");
//Load jdbc driver class
Class.forName("oracle.jdbc.driver.OracleDriver");
logger.info("JDBC driver class is loaded");
//establish the connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
```

App2

====

DeptSelectTest.java

OK

OK

==========

```
//DeptSelectTest.java
package com.nt.main;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

Mezmo

elastic Elastic NV

Management

LogDNA

Logmatic

SigNoz Signoz, Inc.

Alerting

>

>

papertrail Paper trail

GoAcces GoAccess

SOLARWINDS

SolarWinds

Logentries

log Syslog-ng

ENTITY

Bulletlog

```
logger.info("Connection is established");
//create PreparedStatement object
ps=con.prepareStatement(GET_ALL_EMPS);
logger.info("PreparedStatement obj is created");
//execute the Query
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class DeptSelectTest {
rs=ps.executeQuery();
logger.info("JDBC ResultSet object is created");
//process the ResultSet
int count=0;
while(rs.next()) {
logger.debug("start of the main(-) method");
count++;
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4)+" "+rs.getInt(5));
private static final String GET_EMPS_QUERY="SELECT DEPTNO,DNAME, LOC FROM DEPT";
private static Logger logger = LoggerFactory.getLogger(DeptSelectTest.class);
public static void main(String[] args) {
try( //establish the connection
}//while
logger.info("JDBC ResultSEt object processed");
catch(SQLException se) {
```

```
Connection con= DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "tiger");
//create PreparedStatement
if(count==0) {
System.out.println("No records found in the db table");
logger.error("problem in closing the ResultSet obj"); se.printStackTrace();
PreparedStatement ps=con.prepareStatement(GET_EMPS_QUERY);
//execute the Query
logger.warn("ResultSet object should not be the empty object in any giv
}
}
ResultSet rs=ps.executeQuery();
logger.debug("end of try block");
X{
try {
}//try
if(ps!=null)
logger.info("Connection is established");
catch(SQLException se) {
ps.close();
logger.debug("Prepared Statement obj is ready");
logger.error("DB problem :"+se.getMessage());
logger.info("PreparedStatement obj is closed");
logger.debug("Select Query is executed and the ResultSet obj is generated");
se.printStackTrace();
}
//process the ResultSet object
}
catch(SQLException se) {
while(rs.next()) {
catch(ClassNotFoundException cnf) {
se.printStackTrace();
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
logger.error("Problem in Loadidng JDBC driver class");
logger.error("Problem in closing the PreparedStatement object");
}
cnf.printStackTrace();
}
```

```
}
try {
logger.warn("for Salary rs.getFloat(-) is used rahter rs.getDouble(-)"); logger.debug("ResutSet of is Processed");
catch(Exception e) {
if(con!=null)
logger.info("Connection, PreparedStatement, ResultSet objs are closed automatically at end of try with Resource");
logger.error("unknown problem in the Application"+e.getMessage());
con.close();
}
e.printStackTrace();
logger.info("Connection obj is closed");
}
catch(SQLException se) {
}
finally {
logger.error("Known DB problem ::" +se.getMessage());
catch(SQLException se) {
//close jdbc objs
se.printStackTrace();
se.printStackTrace();
logger.debug(" staart of finally block");
}
logger.error("Problem in closing the JDBC connection object");
try {
 }
catch (Exception e) {
if(rs!=null)
logger.debug("end of the finally block");
logger.error("Unknow Problem ::"+e.getMessage());
rs.close();
}//finally
e.printStackTrace();
logger.info("ResultSet obj is closed");
}
}
logger.debug("end of main(-) method");
```

}//main

logger.debug("end of the main(-) method");

}//main

}//class

names and

}//class log4j.properties

step4) Develop the logstash configuration file specifying the input log file

in <logstash_home>\bin directory

outputЯban console

logstash-jdbc.conf

FileAppender and PatternLayout log4j.rootLogger=ALL, R

log4j.appender.R=org.apache.log4j.FileAppender

Input section: defines where Logstash reads the data from

log4j.appender.R.File=d:/jdbc_elk_dept.txt

log4j.appender.R.Append=true

log4j.appender.R.Immediate Flush=true

log4j.appender.R.layout=org.apache.log4j.PatternLayout

'yyy-MM-dd} - %L - %F - [%M] %

log4j.appender.R.layout.Conversion Pattern=[%p] - %c - %t - %r - %m %d{yyyy-MM-dd} - %L - %F - [%M] %n

input {

file {

path => "d:/jdbc_elk_emp.txt" # Path to the input log file

start_position => "beginning" # Read the file from the beginning

}

file {

path => "d:/jdbc_elk_dept.txt" # Path to the input log file

start_position => "beginning" # Read the file from the beginning

}

}

Output section: defines where Logstash sends the processed data

output {

stdout {

codec => rubydebug # Print processed data to console in a readable format

}

elasticsearch {

hosts =>["http://localhost:9200"] # Elasticsearch host

any <file name>.<ext> can be taken as the log stash configuration file

}

step5) execute the configuration file (This links our log file to logstash software then to elastic search)

```
D:\ELKStack1\logstash-7.17.0Soft\logstash-7.17.0\bin>logstash -f logstash-jdbc.conf
```

step5) Open kibana console , to create dashboard and to discover log messages through indexing

a) Create DashBoard by choosing the indexing name

(any name)

menu ---> dash board ---> create dashboard ---> create index ---> index name :: logstash* ---> choose the primary

filed (@timestamp) ---> add/remove the fields ---> .....

b) discover log messages

Search

menu ---> discover ---> select index (logstash*) ----> refresh ---> (to see all the messages)

Discover

=

+ Add filter

logstash* ✓

000

37 hits

Q Search field names

Filter by type

0

✓ Available fields

Popular

t message

t_id

t_index

#_score

t_type

apply more filters

D

Discover

25

20

15

5

0

19:45:00

19:50:00

KQL 聞く

Last 30 minutes

19:55:00

Options New Open

Share Inspect

Save

Show dates

C Refresh

20:00:00

20:05:00

20:10:00

Aug 5, 2023 @ 19:44:47.601 - Aug 5, 2023 @ 20:14:47.601

Time ↓

Document

>

Aug 5, 2023 @ 19:50:27.134

Chart options

```
@timestamp: Aug 5, 2023 @ 19:50:27.134 @version: 1 host: DESKTOP-QENT2RN message:
2023-08-04T20:22:50.497+05:30 INFO 25720 [http-nio-9001-exec-9]
com.nt.ms.PaymentOperationsController: At beging of doPayment() method path:
d:/elk_log.txt _id: OmMSxokB91_hcVD8LNj2 _index: logstash-2023.08.05-000001
```

Activate WindowS

message:problem

select the filed and write

KQL

Last 1 hour

condition + content to search

=

+ Add filter

in the log messages

logstash* ✓

4 hits

19:40

19:45

Options

New

Open

Share Inspect

Save

Show dates C Refresh

19:50

Chart options

19:55

20:00

20:05

20:10

20:15

Aug 5, 2023 @ 19:16:08.033 - Aug 5, 2023 @ 20:16:08.033

Q Search field names

4

3

2

Filter by type 0

1

0

✓ Available fields

9

19:20

19:25

19:30

19:35

Popular

t message

t_id

Time↓

Document

>

Aug 5, 2023 @ 19:38:03.169

t_index

#_score

t_type

message: 2023-08-04T20:22:54.903+05:30 ERROR 25720 --- [http-nio-9001-exec-4] com.nt.ms.PaymentOperationsController Problem is billamoint generated @timestamp: Aug 5, 2023 @ 19:38:03.169 @version: 1 host: DESKTOP-QENT2RN path: d:/elk_log.txt

_id: 6GMGxokB9i_hcVD809cI _index: logstash-2023.08.05-000001

score:

Activate Window type: -doc

More samples on KQL (Kibana Query Language)

This ELK stack can be applied on single App and on multiple Apps that are using multiple log files

note:: Since the ELK stack allows seamlesss integration with any domain based application development, we can use ELK stack along with other logging and tracing mechansims

with slf4j, log4j and etc ...

問く @timestamp < now-10m AND message:INFO

what is indexing in Logstash

In Logstash, "indexing" generally refers to the process of sending and storing data in an Elasticsearch index. An index in Elasticsearch is similar to a database in relational database systems,

and it's where Logstash pushes data for quick retrieval and analysis.

How Indexing Works in Logstash

1. Data Collection: Logstash collects data from various sources (logs, metrics, etc.) as specified in

the input section of its configuration file.

2. Data Processing: The data passes through filter plugins for parsing, enriching, and transforming (e.g., using grok for pattern matching, date for timestamp processing, etc.).

3. Data Indexing: In the output section of ↓ configuration, you specify where the processed data should go. By default, if you output to Elasticsearch, Logstash will index the data by creating a document in a specified Elasticsearch index.

plaintext

output {

elasticsearch {

hosts =>["http://localhost:9200"]

index => "your_index_name"

}

}

Index name where data is stored

Copy code

4. Indexing Format: Logstash can dynamically create indices based on timestamp patterns (e.g.,

daily indices like logs-YYYY.MM.DD) or other fields in the data, making it easier to search and manage logs based on time or other criteria.

5. Index Storage in Elasticsearch: Each new document sent to Elasticsearch from Logstash is

indexed and stored in the specified index. Elasticsearch uses inverted indices, making searches and retrievals extremely fast.

Why Indexing is Important

• Searchability: Indexed data is easily searchable, allowing users to query specific fields, patterns,

or time ranges.

• Analytics: Indexing enables efficient data aggregation and analysis, making it powerful for log analysis, error

tracking, and real-time monitoring.

• Scalability: By creating multiple indices (e.g., daily indices), you can manage and scale large volumes of log data more effectively.

In summary, "indexing" in Logstash is the process of preparing data and sending it to an Elasticsearch index, enabling fast search and analysis in the Elasticsearch-Logstash-Kibana (ELK)

stack.

FAQS

====

Q) what is ELK stack

Q) What is difference b/w Logging tools and Log monitoring tool?

Q) What is the difference b/w ELK stack and Log4j/SLF4J?

Q) List out Various Log Monitoring Tools?

Q) What is the role of each Tool in ELK stack?

Q) Why the Logstash tool is called Pipeline tool?

Q) What is indexing in the ELK stack?

Q) Explain various components of Logstash configuration file?

Q) How to map Elastic Serach Server with Kibana Server

Q) What is KQL (Kibana Query Language)

Q) List out all

the fields that can be used in KQL Queries

Q) List out all the fields that can be used in indexing