Spring boot Central Loggging/Distributed Logging using ELK Stack

---

ELK stack is the combination of 3 tools

E--> Elestic Search L ---> Logstash

K ---> Kibana

:: It is a NO SQL DB s/w which can store the Application generated log messages

:: It is a pipe line tool that takes log messages from different sources (like Apps)

and exports them to different targets (likestic seahving the indexing

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

(Best)

What is the difference b/w performing loggingpions using ELK statck and using Sleuth and Zipkin?

JAVA

messages

=>ELK stack supports seamless integration with spring/spring boot applications (all kinds of applications) i.e with out adding any starters related to ELK stack to spring/spring boot Apps we can store and process the log messages generated by the applications or microservices .ELK stack can be applied on java,.net, python and etc.. projects => Sleuth & zipkin does not support this Seamless integration i.e we need to add the related starters to spring/spring boot applications Sleuth &zipkin can not be added in other than java projects

=> ELK statck can do only logging operations where as slueth and zipkin can do both tracing and logging opittings --> to get the flow of execution

=>ELK stack can be applied in both JAva and non- java apps where as slueth and zipkin can be used only in Java Apps

=>log4j, sl4fj, logback and etc.. are called logging tools to generate the log messages => ELK stack, sleuth, zipkin and etc.. are called Log monitoring tools becoz theytake the log message from the log file(s) of the App(s) and provide env.. to monitor the log messages in GUI env...

=> ELK stack(Open Source), splunk(commercial) and etc,, are called Log Monitorig tools i.e using these tools we can not generate the log message but we can monitor the generated log messages

=> In MicroServices archicture we have different Ms Projects

In

representing different services and each project generates its own log file using own logging api. To monitor all those log files related log message in single GUI env we need to use Log Monitoring tools like ELK Stack, Splunk

=> Spring boot 2.x, we need to take sleuth,zipkin tools for log monitoring activities ..which is removed in spring boot 3.x.. So in spring boot 3.x take the support of ELK stack for the same (In fact, ELK statck can be used in all versions of Spring Boot)

=> ELK Stack can be used in all the versions of spring /spring boot Projects.. where as sleuth, zipkin can be used only in spring boot 2.x version projects

Java /Non-Java App (s)

(any Technology App)

uses logging api of

certain domain

log file (s)

LogStash

(b)

(c)

(a)

(pipeline)

Elestic Search (NO SQL DB) (processor)

note:: ELK stack tool is no way related to certain language or technology or framework or their versions i.e it can be used any where

What is distributed Logging? (or) What is Distributed Log Monitoring?

Ans) when multiple Apps/comps/microservices are in communication there will be multiple log files.. In order gather the log messages of the all the log files and to format and index them having better presentation take the support of Distributed Logging tools (or) Log Monitoring tools

eg:: ELK statck tools, Zipkin, sleuth tools Splunk

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

(b) Logstash tool takes the log messages from the log file 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

tool for UI env.. to see and use the log messages (Log Monitoring)

Example App on MS intra Communication based cental logging using ELK Statck Tools

of

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

choose 7.17 version

b) elestic search download

C

elastic.co/downloads/elasticsearch

Download Elasticsearch

→ C

1 Download and unzip Elasticsearch

Choose platform:

Windows

Windows

sha asc

c) Log stash

==========

elastic.co/downloads/logstash

1 Download and unzip Logstash

Choose platform:

Windows

www.> Downloads >

Name

choose 7.17 version

✓ Today

Windows

sha asc

choose 7.17 version

Type

(d)

Kibana

(Gives UI for the messages)

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 server with kibana server)

step3)

ELK Stack Pros

1. Free to Get Started

MocroService#1

log file

MocroService#2

log file

MocroService#4

MocroService#3

log file

log file

Distributed Logging Tools (ELK or zipkin -sieuth or Splunk)

Gives better Log messages presentation

One of the key reasons for the growth in popularity of the ELK stack is its low financial barrier to entry. All of the software components of ELK are free and open-source tools - that means no up- front purchases are required and there are no ongoing software licensing fees.

2. Multiple Hosting Options

When it comes to deploying an ELK stack, organizations have multiple hosting options to choose from. For organizations with the right capabilities and resources, an ELK stack can be installed on a local server and managed in-house. Alternatively, organizations can choose to deploy their ELK stack as a managed service with products like Amazon OpenSearch by partnering with a specialist MSP.

3. Centralized Logging Capabilities

One of the most important features of the ELK stack is that it offers centralized logging capabilities, allowing users to aggregate logs from increasingly complex cloud environments into a single searchable index. This capability makes it possible to correlate log and event data from multiple sources, enabling use cases like security monitoring and root cause analysis.

4. Real-Time Data Analysis & Visualization

With Kibana, ELK stack users can create data visualizations and build custom dashboards using real- time data from Elasticsearch. The ability to visualize data in real time decreases time-to-insights, supporting a variety of use cases and driving organizational agility and informed decision-making. Installing Kibana is easy, and teams can build a Kibana dashboard using third party log analytics solutions that are more cost-effective and easy-to-use than Elasticsearch.

5. Official Clients in Multiple Programming Languages

Some ELK stack users have multiple languages in their codebase and wish to use Elasticsearch from all of them. To support this requirement, developers at Elastic have released official clients for Elasticsearch in at least 10 programming languages, including JavaScript, Go, Python, .NET, and Perl. In addition, the Elasticsearch open source community has contributed clients in various languages. Elastic provides support for all of its official clients, fixing bugs and responding to support queries as needed.

b) start elastic search server

`i)`

bin

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

localhost:9200

X

Browse integrations - Integration X

← →

C

localhost:9200

▼ {

"name": "DESKTOP-QENT2RN",

"cluster_name": "elasticsearch",

"cluster_uuid": "MxoCCjzMS2eXjwVwHI6tVQ",

▼ "version": {

"number": "7.17.0",

"build_flavor": "default",

"build_type": "zip",

"build_hash": "bee86328705acaa9a6daede7140defd4d9ec56bd"

"build_date":"2022-01-28T08:36:04.875279988Z",

"build_snapshot": false,

"lucene_version": "8.11.1",

"minimum_wire_compatibility_version": "6.8.0",

"minimum_index_compatibility_version": "6.0.0-beta1"

"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

http://localhost:5601

==>SLF4J/Log4j,Logback and etc... are called Logging tools are called logging tools becoz these tools generate the log messages to log files

==> ELK stack,splunk, sleuth and zipkin are called Log monitoring tools becoz using these tools we gather the generated log messages for better use and monitoring of log messages

localhost:9200

elastic

X

Browse integrations - Integration X

+

localhost:5601/app/integrations/browse

Integrations Browse integrations

Q Search Elastic

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.

Develop multiple microservices in communication along with eureka server (Take separate log file in each MS / Eureka Server Project)

a) create spring MS or Rest Service Project (web starter, lombok starter)

b) add the following entries in application.properties

application.properties

 # MS port number

server.port=9001

 # Logging file name

logging.file.name=d:/shopping_elk_log1.txt

> BootMsProj 12-Billing ServiceMS-ELKStack [boot]

> BootMsProj12-EurekaServer-ELKStack [boot]

> BootMsProj12-ShoppingServiceMS-ELKStack [boot]

Taking multiple MS

in communication

#specify the logger level as DEBUG for "com.nt" (root) package logging.level.com.nt=DEBUG

 # Application name

spring.application.name=ELKTestApp

c) develop the Restcontroller linking with slf4j

As taken for the given MS Project, enable logging on all 3 projects having 3 separate log files

|--->d:/billing_elk_log2.txt

|--->d:/shopping_elk_log1.txt

|--->d:/eureka_elk_log3.txt

=>@Slf4j annotation gives Logger obj having name log dynamically

=> The equalent code is

```
@RestController
@slf4j
@RequestMapping("/shopping-api")
public class ShoppingServiceOperationsController { @Autowired
private BillingService_ConsumingClient clientComp;
@GetMapping("/info")
public ResponseEntity<String> shoppingInfo(){ log.info("start of shoppingInfo() -- ShoppingMs"); // use the provider MS
String billingInfo=clientComp.getBillingInfo();
log.info("Dest MS is invoked thriugh client comp");
// get shopping MS result
```

logger level priorities

TRACE<DEBUG<INFO<WARN<ERROR

String shoppingInfo="The Shopping BillAmount is::"+new Random().nextInt(20000);

// combine the final result

String finalResult=shoppingInfo+"..."+billingInfo;

log.info("Final result is prepared and send ");

return new ResponseEntity<String>(finalResult, HttpStatus.OK);

}

d) Run MS application /Rest service Application and check the log file

Discover

Elastic

C

A Not secure

Eureka

192.168.1.236:6061/shopping-api/info

Logger log=LoggerFactory.getLogge<classname>.class); (slf4j code)

note:: While working LBC, FeignClient we need to take multiple instance of provider MS having different port numbers.. they fail when we take the single instance

192.168.1.236:6061/shopping-a

+

ŭ

The Shopping BillAmount is::11207...we accept UPI Payments,DebitCards,Credit Card and etc.. from the instance-->Billing-Service: 7ef5161611814b0df519389ffc954efd....8082 names and output kiban console

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

in <logstash_home>\bin directory

logstash- Ms. conf

input {

file {

path => "d:/eureka_elk_log2.txt"

start_position => "beginning"

}

file {

path => "d:/shopping_elk_log3.txt"

start_position => "beginning"

}

file {

path => "d:/billing_elk_log1.txt"

start_position => "beginning"

}

}

output {

elasticsearch {

hosts =>["localhost:9200"]

}

stdout { codec => rubydebug}

}

file name pattern logstash-<name>.conf

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

D:\ELKStack1\Logstash-7.17.05oft\logstash-7.17.8\bin>logstash -f logstash- Ms.conf

TAVA HAME J.

„ŠJI NILA A

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

a) Create DashBoard by choosing the indexing name

visualize dashboard

(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

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

Discover

固<

Search

+ Add filter

logstash*

Q Search field names

□□□ 37 hits

25

20

15

Filter by type

0

✓ Available fields

Popular

t message

t_id

t_index

#_score

t_type

apply more filters

D

Discover

message:problem

=

+ Add filter

logstash*

Q Search field names

Filter by type 0

✓ Available fields

9

Popular

t message

t_id

t_index

#_score

t_type

0

9

19:45:00

19:50:00

Options

New

Open

Share Inspect

Save

KQL

聞く Last 30 minutes

Show dates C Refresh

19:55:00

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

000 €

4 hits

3

2

1

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

KQL

聞く

Last 1 hour

Activate Windows

Options New Open Share Inspect

Save

Show dates

C Refresh

Chart options

19:20

19:25

19:30

19:35

19:40

19:45

19:50

19:55

20:00

20:05

20:10

20:15

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

Time↓

Document

>

Aug 5, 2023 @ 19:38:03.169

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

O

More samples on KQL (Kibana Query Language)

@timestamp < now-10m AND message:INFO

Activa core: dowype: doc

@timestamp now -120m and message: INFO

This ELK stack can be applied on single MS /RestService and also can be applied RestServices/MicroServices that are there in communication

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 like zipkin and sleuth..

yes s

To:

EI