Continuous Monitoring On Docker With ELK Stack project source code

DONE BY : B.UDAY NARASA REDDY

DemoApplication.java

```
package com.example.demo;
 import org.slf4j.Logger; import
org.slf4j.LoggerFactory; import
org.springframework.boot.SpringApplication; import
org.springframework.boot.autoconfigure.SpringBootApplica
tion; import
org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RestController;
@SpringBootApplication
@RestController public class
DemoApplication {

    private static final Logger logger =
LoggerFactory.getLogger(DemoApplication.class);

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class,
args);
    }

    @GetMapping("/")     public
String hello() {
logger.info("Hello World");
```

```java
        logger.error("Ooops, there was an error", new 
RuntimeException("I am a runtime exception"));
return "Hello World";
    }
}
```

Logback.xml

```xml
<configuration>
    <appender name="jsonConsoleAppender" 
class="ch.qos.logback.core.ConsoleAppender">
        <encoder 
class="net.logstash.logback.encoder.LogstashEncoder"/>
    </appender>
    <root level="INFO">
        <appender-ref ref="jsonConsoleAppender"/>
    </root>
</configuration>
```

DemoApplicationsTests.java

```java
package com.example.demo;
```

```
 import org.junit.Test; import
org.junit.runner.RunWith; import
org.springframework.boot.test.context.SpringBootTest;
import
org.springframework.test.context.junit4.SpringRunner;
```

```
@RunWith(SpringRunner.class)
@SpringBootTest public class
DemoApplicationTests {

    @Test     public void
contextLoads() {
    }

}
```

Logback.xml

```
<configuration>
    <appender name="jsonConsoleAppender"
class="ch.qos.logback.core.ConsoleAppender">
        <encoder
class="net.logstash.logback.encoder.LogstashEncoder"/>
    </appender>
    <root level="INFO">
        <appender-ref ref="jsonConsoleAppender"/>
    </root>
</configuration>
```

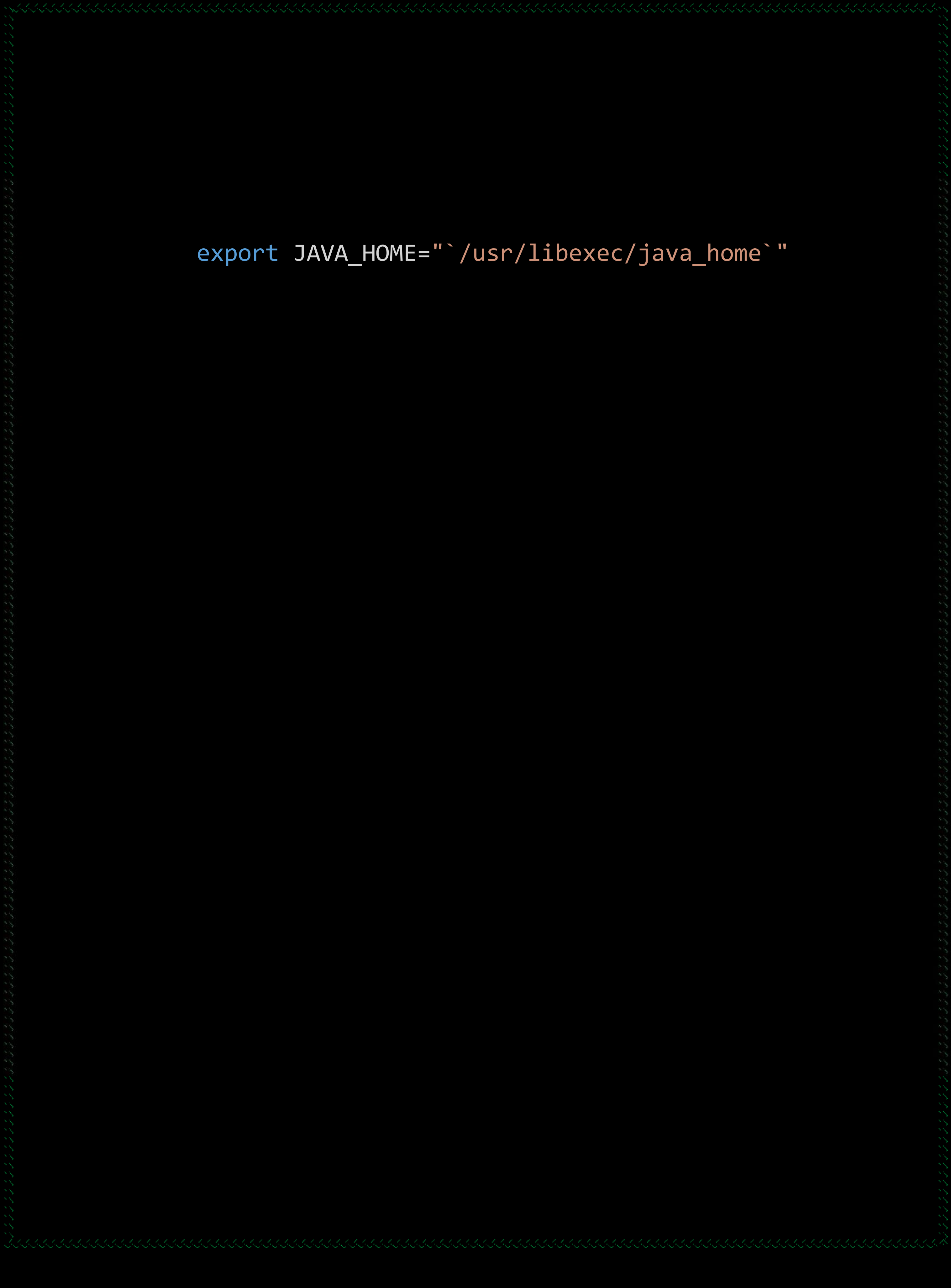
Dockerfile

```
FROM openjdk:8-jdk-alpine
VOLUME /tmp
ADD target/demo-0.0.1-SNAPSHOT.jar app.jar
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","jar","/app.jar"]
```

Mvnw

```
if [ -z "$MAVEN_SKIP_RC" ] ; then
   if [ -f /etc/mavenrc ] ;
then
    . /etc/mavenrc   fi    if [ -f
"$HOME/.mavenrc" ] ; then
    . "$HOME/.mavenrc"
fi  fi

# OS specific support.  $var _must_ be set to
either true or false. cygwin=false; darwin=false;
mingw=false case "`uname`" in
  CYGWIN*) cygwin=true ;;
  MINGW*) mingw=true;;
  Darwin*) darwin=true
    # Use /usr/libexec/java_home if available, otherwise
fall back to /Library/Java/Home
    # See
https://developer.apple.com/library/mac/qa/qa1170/_index
.html     if [ -z "$JAVA_HOME" ]; then
if [ -x "/usr/libexec/java_home" ]; then
```



```
export JAVA_HOME="`/usr/libexec/java_home`"
```

```
      else          export
JAVA_HOME="/Library/Java/Home"       fi
fi     ;; esac  if [ -z "$JAVA_HOME" ] ;
then   if [ -r /etc/gentoo-release ] ; then
JAVA_HOME=`java-config --jre-home`   fi fi
if [ -z "$M2_HOME" ] ; then
  ## resolve links - $0 may be a link to maven's home
  PRG="$0"

  # need this for relative symlinks   while [
-h "$PRG" ] ; do     ls=`ls -ld "$PRG"`
link=`expr "$ls" : '.*-> \(.*\)$'`     if
expr "$link" : '/.*' > /dev/null; then
PRG="$link"     else
      PRG="`dirname
"$PRG"`/$link"     fi   done
saveddir=`pwd`
```

```
M2_HOME=`dirname "$PRG"`/..
```

```
  # make it fully qualified
  M2_HOME=`cd "$M2_HOME" && pwd`
   cd
"$saveddir"
  # echo Using m2 at $M2_HOME fi

# For Cygwin, ensure paths are in UNIX format
before anything is touched if $cygwin ; then
  [ -n "$M2_HOME" ] &&
    M2_HOME=`cygpath --unix "$M2_HOME"`
  [ -n "$JAVA_HOME" ] &&
    JAVA_HOME=`cygpath --unix "$JAVA_HOME"`
  [ -n "$CLASSPATH" ] &&
    CLASSPATH=`cygpath --path --unix "$CLASSPATH"` fi

# For Migwn, ensure paths are in UNIX format
before anything is touched if $mingw ; then
  [ -n "$M2_HOME" ] &&
    M2_HOME="`(cd "$M2_HOME"; pwd)`"
  [ -n "$JAVA_HOME" ] &&
    JAVA_HOME="`(cd "$JAVA_HOME"; pwd)`"
# TODO classpath?
fi
```

```
if [ -z "$JAVA_HOME" ]; then
javaExecutable="`which javac`"
  if [ -n "$javaExecutable" ] && ! [
 "`expr
\"$javaExecutable\" : '\([^ ]*\)'`" = "no" ]; then
# readlink(1) is not available as standard on
Solaris 10.     readLink=`which readlink`     if [ !
`expr "$readLink" : '\([^ ]*\)'` = "no" ]; then
if $darwin ; then         javaHome="`dirname
\"$javaExecutable\"`"         javaExecutable="`cd
\"$javaHome\" && pwd P`/javac"       else
javaExecutable="`readlink -f
\"$javaExecutable\"`"       fi
javaHome="`dirname \"$javaExecutable\"`"
javaHome=`expr "$javaHome" : '\(.*\)/bin'`
JAVA_HOME="$javaHome"       export JAVA_HOME
fi   fi fi  if [ -z "$JAVACMD" ] ; then   if [ -
n "$JAVA_HOME"  ] ; then     if [ -x
"$JAVA_HOME/jre/sh/java" ] ; then
      # IBM's JDK on AIX uses strange locations for the
executables
      JAVACMD="$JAVA_HOME/jre/sh/java"
```

```
  else
     JAVACMD="$JAVA_HOME/bin/java"
fi
```

```
  else
    JAVACMD="`which java`"   fi fi  if [ ! -x
"$JAVACMD" ] ; then   echo "Error: JAVA_HOME is not
defined correctly." >&2   echo "  We cannot execute
$JAVACMD" >&2   exit 1 fi  if [ -z "$JAVA_HOME" ] ;
then   echo "Warning: JAVA_HOME environment variable is
not set." fi

CLASSWORLDS_LAUNCHER=org.codehaus.plexus.classworlds.lau
ncher.Launcher

# traverses directory structure from process work
directory to filesystem root
# first directory with .mvn subdirectory is
considered project base directory
find_maven_basedir() {
   if [ -z "$1" ]   then     echo "Path not
specified to find_maven_basedir"
```

```
    return 1
```

```
fi
```

```
  basedir="$1"   wdir="$1"
while [ "$wdir" != '/' ] ; do
if [ -d "$wdir"/.mvn ] ; then
basedir=$wdir       break     fi
    # workaround for JBEAP-8937 (on Solaris
10/Sparc)     if [ -d "${wdir}" ]; then
wdir=`cd "$wdir/.."; pwd`     fi
    # end of workaround
done   echo
"${basedir}"
}

# concatenates all lines of a file
concat_lines() {   if [ -f "$1" ];
then     echo "$(tr -s '\n' ' ' <
"$1")"   fi
}

BASE_DIR=`find_maven_basedir
"$(pwd)"` if [ -z "$BASE_DIR" ]; then
exit 1; fi
```

```
export MAVEN_PROJECTBASEDIR=${MAVEN_BASEDIR:-
"$BASE_DIR"}
echo $MAVEN_PROJECTBASEDIR MAVEN_OPTS="$(concat_lines
"$MAVEN_PROJECTBASEDIR/.mvn/jvm.config") $MAVEN_OPTS"

# For Cygwin, switch paths to Windows format before
running java if $cygwin; then
  [ -n "$M2_HOME" ] &&
    M2_HOME=`cygpath --path --windows "$M2_HOME"`
  [ -n "$JAVA_HOME" ] &&
    JAVA_HOME=`cygpath --path --windows "$JAVA_HOME"`
[ -n "$CLASSPATH" ] &&
    CLASSPATH=`cygpath --path --windows "$CLASSPATH"`
  [ -n "$MAVEN_PROJECTBASEDIR" ] &&
    MAVEN_PROJECTBASEDIR=`cygpath --path --windows
"$MAVEN_PROJECTBASEDIR"` fi

WRAPPER_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMa
in  exec "$JAVACMD" \
  $MAVEN_OPTS \
  -classpath
"$MAVEN_PROJECTBASEDIR/.mvn/wrapper/mavenwrapper.jar" \
  "-Dmaven.home=${M2_HOME}" "-
Dmaven.multiModuleProjectDirectory=${MAVEN_PROJECTBASEDI
R}" \
  ${WRAPPER_LAUNCHER} $MAVEN_CONFIG "$@"
```

Mvnw.cmd

```
@REM Begin all REM lines with '@' in case
MAVEN_BATCH_ECHO is 'on'
@echo off
@REM enable echoing my setting MAVEN_BATCH_ECHO to 'on'
@if "%MAVEN_BATCH_ECHO%" == "on"  echo
%MAVEN_BATCH_ECHO%

@REM set %HOME% to equivalent of $HOME if "%HOME%"
== "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")

@REM Execute a user defined script before this one if
not "%MAVEN_SKIP_RC%" == "" goto skipRcPre
@REM check for pre script, once with legacy .bat
ending and once with .cmd ending if exist
"%HOME%\mavenrc_pre.bat" call
"%HOME%\mavenrc_pre.bat" if exist
"%HOME%\mavenrc_pre.cmd" call
"%HOME%\mavenrc_pre.cmd"
:skipRcPre

@setlocal
 set
ERROR_CODE=0
```

```
@REM To isolate internal variables from possible post scripts, we use another setlocal
@setlocal
```

```
@REM ==== START VALIDATION ====
if not "%JAVA_HOME%" == "" goto OkJHome
 echo.
echo Error: JAVA_HOME not found in your environment.
>&2 echo Please set the JAVA_HOME variable in your
environment to match the >&2 echo location of your Java
installation. >&2 echo.
goto error
:OkJHome if exist "%JAVA_HOME%\bin\java.exe"
goto init
 echo. echo Error: JAVA_HOME is set to an invalid
directory.
>&2 echo JAVA_HOME = "%JAVA_HOME%" >&2 echo
Please set the JAVA_HOME variable in your
environment to match the >&2 echo location of
your Java installation. >&2 echo.
goto error

@REM ==== END VALIDATION ====

:init

@REM Find the project base dir, i.e. the directory that
contains the folder ".mvn".
```

```
@REM Fallback to current working directory if not found.
```

```
set MAVEN_PROJECTBASEDIR=%MAVEN_BASEDIR% IF
NOT "%MAVEN_PROJECTBASEDIR%"=="" goto
endDetectBaseDir
 set EXEC_DIR=%CD%
set WDIR=%EXEC_DIR%
:findBaseDir
IF EXIST "%WDIR%"\.mvn goto baseDirFound cd
..
IF "%WDIR%"=="%CD%" goto baseDirNotFound
set WDIR=%CD% goto findBaseDir

:baseDirFound set
MAVEN_PROJECTBASEDIR=%WDIR% cd
"%EXEC_DIR%" goto
endDetectBaseDir

:baseDirNotFound set
MAVEN_PROJECTBASEDIR=%EXEC_DIR% cd
"%EXEC_DIR%"

:endDetectBaseDir

IF NOT EXIST "%MAVEN_PROJECTBASEDIR%\.mvn\jvm.config"
goto endReadAdditionalConfig
```

```
@setlocal EnableExtensions EnableDelayedExpansion
```

```
for /F "usebackq delims=" %%a in
("%MAVEN_PROJECTBASEDIR%\.mvn\jvm.config") do set
```

```
JVM_CONFIG_MAVEN_PROPS=!JVM_CONFIG_MAVEN_PROPS! %%a
@endlocal & set
JVM_CONFIG_MAVEN_PROPS=%JVM_CONFIG_MAVEN_PROPS%

:endReadAdditionalConfig

SET MAVEN_JAVA_EXE="%JAVA_HOME%\bin\java.exe"
 set
WRAPPER_JAR="%MAVEN_PROJECTBASEDIR%\.mvn\wrapper\mavenwrapper.jar'
set
WRAPPER_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMa in

%MAVEN_JAVA_EXE% %JVM_CONFIG_MAVEN_PROPS% %MAVEN_OPTS%
%MAVEN_DEBUG_OPTS% -classpath %WRAPPER_JAR% "-
Dmaven.multiModuleProjectDirectory=%MAVEN_PROJECTBASEDIR
%" %WRAPPER_LAUNCHER% %MAVEN_CONFIG% %* if
ERRORLEVEL 1 goto error goto end  :error
set ERROR_CODE=1

:end
@endlocal & set ERROR_CODE=%ERROR_CODE%
 if not "%MAVEN_SKIP_RC%" == "" goto skipRcPost
```

```
@REM check for post script, once with legacy .bat ending
and once with .cmd ending if exist
"%HOME%\mavenrc_post.bat" call
"%HOME%\mavenrc_post.bat" if exist
"%HOME%\mavenrc_post.cmd" call
"%HOME%\mavenrc_post.cmd"
:skipRcPost

@REM pause the script if MAVEN_BATCH_PAUSE is set to
'on' if "%MAVEN_BATCH_PAUSE%" == "on"
pause
 if "%MAVEN_TERMINATE_CMD%" == "on" exit
%ERROR_CODE%
 exit /B
%ERROR_CODE%
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>
    <name>demo</name>
    <description>Demo project for Spring
Boot</description>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-
starterparent</artifactId>
        <version>1.5.9.RELEASE</version>
<relativePath/> <!-- lookup parent from repository -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-
```

```
8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-
starterweb</artifactId>
        </dependency>

        <dependency>
            <groupId>net.logstash.logback</groupId>
<artifactId>logstash-logback-
```

```
encoder</artifactId>
            <version>4.11</version>
</dependency>

        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-
startersleuth</artifactId>
            <version>1.3.0.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-
startertest</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-
mavenplugin</artifactId>             </plugin>
```

```
            <plugin>
                <groupId>com.spotify</groupId>
<artifactId>dockerfile-mavenplugin</artifactId>
```

```
                <version>1.3.6</version>
                <configuration>
                    <repository>${project.artifactId}</repository>
                    <buildArgs>
                        <JAR_FILE>target/${project.build.finalName}.jar</JAR_FILE>
                    </buildArgs>
                </configuration>
            </plugin>
        </plugins>
    </build>


</project>
```

Filebeat Dockerfile

```
FROM docker.elastic.co/beats/filebeat:7.0.0
COPY filebeat.yml /usr/share/filebeat/filebeat.yml
# must run as root to access /var/lib/docker and /var/run/docker.sock
USER root
RUN chown root /usr/share/filebeat/filebeat.yml
```

```
# dont run with -e, to disable output to stderr
CMD [""]
```

Filebeat.yml

```
# filebeat.yml
```

```
filebeat.inputs: - type:
docker   containers.ids: '*'
json.message_key: message
json.keys_under_root: true
json.add_error_key: true
json.overwrite_keys: true
 processors:
- add_docker_metadata: ~
 output.elasticsearch:
  hosts: ["elasticsearch:9200"]
 logging.to_files: true
logging.to_syslog: false
```

Docker-compose.yml

```
version: '3.2'
services:
demo:
    # run `./mvnw clean package`
before     build: ./demo     ports:
- 8080:8080   filebeat:
    build: ./filebeat
volumes:
      -
/var/lib/docker/containers:/var/lib/docker/containers:ro
```

```
-    /var/run/docker.sock:/var/run/docker.sock
networks:       - es     depends_on:       -
elasticsearch   kibana:
    image: docker.elastic.co/kibana/kibana:7.0.0
ports:       - 5601:5601     environment:
      ELASTICSEARCH_URL: http://elasticsearch:9200
networks:       - es     depends_on:       -
elasticsearch   elasticsearch:     image:
docker.elastic.co/elasticsearch/elasticsearch:7.0.0
container_name: elasticsearch     environment:
-    cluster.name=docker-cluster
-    "ES_JAVA_OPTS=-Xms512m -Xmx512m"
-    "network.host=0.0.0.0"
-    "discovery.zen.minimum_master_nodes=1"
-    "discovery.type=single-node"     ulimits:
memlock:         soft: -1         hard: -1
```

```
    volumes:
      - esdata:/usr/share/elasticsearch/data
```

```
    ports:
- 9200:9200
networks:
      - es
 volumes:
esdata:
    driver: local
 networks:
es:
```

Log4j2.properties

```
status = error
 appender.console.type = Console appender.console.name =
console appender.console.layout.type = PatternLayout
appender.console.layout.pattern = [%d{ISO8601}][%-5p][%-
25c{1.}] %marker%m%n
 rootLogger.level = info
rootLogger.appenderRef.console.ref =
Console
```

Elasticsearch.yml

```
cluster.name: "docker-cluster" network.host:
0.0.0.0
# minimum_master_nodes need to be explicitly set when
bound on a public IP
# set to 1 to allow single node clusters
# Details:
https://github.com/elastic/elasticsearch/pull/17288
discovery.zen.minimum_master_nodes: 1
```

Kibana.yml

```
# Default Kibana configuration from kibana-docker.
 server.name: kibana server.host: "0"
elasticsearch.url:
http://elasticsearch:9200
```

Pipelines.yml

```
# This file is where you define your pipelines. You can
define multiple.
# For more information on multiple pipelines, see the
documentation:
#
https://www.elastic.co/guide/en/logstash/current/multipl
e-pipelines.html

- pipeline.id: main   path.config:
"/usr/share/logstash/pipeline"
```

Logstash.conf

```
input       {
beats {
    port => 5044
  }   udp {      port =>
31338          codec  =>
msgpack          type  =>
ceilometer       tags  =>
"ceilometer"
  }    udp {
port => 5567
tags => "netflow"
type => netflow
codec => netflow
  }    udp {     port =>
5566     tags =>
"netflowprod"     type
=> netflowprod     codec
=> netflow
  }

  udp {     port => 25826
buffer_size => 1452
codec => collectd { }
tags => "collectdceph"
type => collectdceph
```

```
}
```

```
}  filter {       if [type] == "ceilometer" and
[counter_name] ==
"bandwidth" {
date {
          match => [ "timestamp", "YYY-MM-dd
HH:mm:ss.SSSSSS" ]            remove_field =>
"timestamp"            timezone => "UTC"
        }       }       if [type] == "ceilometer"
and [counter_name] ==
"volume" {
date {
          match => [ "timestamp", "YYY-MM-dd
HH:mm:ss.SSSSSS" ]            remove_field =>
"timestamp"            timezone => "UTC"
        }
date {
match
=>["[resource_metadata][created_at]","YYY-MM-dd
HH:mm:ss"]            remove_field =>
"[resource_metadata][created_at]"
target =>
"[resource_metadata][created_at_parsed]"
timezone => "UTC"      }      }
```

```
      if [type] == "ceilometer" and [counter_name] ==
"volume.size" {
date {
          match => [ "timestamp", "YYY-MM-dd
HH:mm:ss.SSSSSS" ]           remove_field =>
"timestamp"           timezone => "UTC"
        }
date {
match
=>["[resource_metadata][created_at]","YYY-MM-dd
HH:mm:ss"]           remove_field =>
"[resource_metadata][created_at]"
target =>
"[resource_metadata][created_at_parsed]"
timezone => "UTC"
        }       }       if [type] ==
"netflow" {         translate {
field => "[netflow][ipv4_src_addr]"
destination => "[netflow][tenantID]"
dictionary_path =>
"/usr/share/logstash/mapping/TenantIpMatches.yaml"
add_field => {
            "[netflow][direction]" => "incoming"
          }
```

```
       }      translate {           field =>
"[netflow][ipv4_dst_addr]"
```

```
          destination => "[netflow][tenantID]"
dictionary_path =>
"/usr/share/logstash/mapping/TenantIpMatches.yaml"
add_field => {
            "[netflow][direction]" => "outgoing"
          }
        }       }       if [type] ==
"netflowprod" {         translate {
field => "[netflow][ipv4_src_addr]"
destination => "[netflow][tenantID]"
dictionary_path =>
"/usr/share/logstash/mappingprod/TenantIpMatches.yaml"
add_field => {
            "[netflow][direction]" => "incoming"
          }         }         translate {
field => "[netflow][ipv4_dst_addr]"
destination => "[netflow][tenantID]"
dictionary_path =>
"/usr/share/logstash/mappingprod/TenantIpMatches.yaml"
add_field => {
            "[netflow][direction]" => "outgoing"
          }
        }
      }
```

```
}
```

```
output {
   if "ceilometer" in [tags] {     elasticsearch
{                    index => "logstash-
ceilometer-
%{+YYYY.MM.dd}"                  hosts =>
["172.26.36.2:9200"]
    }
  }     if "netflow" in [tags] {      elasticsearch {
index          =>          "netflow-new-%{+YYYY.MM.dd}"
hosts =>  ["172.26.36.2:9200"]
    }
  }    if "netflowprod" in [tags] {
elasticsearch {                  index =>
"netflow-prod-new-
%{+YYYY.MM.dd}"                  hosts =>
["172.26.36.2:9200"]
    }
  }    if "collectdceph" in [tags] {     elasticsearch {
index => "collectdceph-%{+YYYY.MM.dd}"
                  hosts =>  ["172.26.36.2:9200"]
```

```
        }
    }
```

```
}
```

===================================X=====================================