Course Exercises

IBM Operations Analytics Log Analysis: Scalable Collection Architecture

Course code TN631 ERC 1.0

IBM Training

August 2016 edition

Contents

About these exercises

Server access

The lab uses six Linux servers that run in VMware. You access these servers through the VMware console.

collection.csite.ibm.edu

This host runs Filebeat and the IBM Log File Agent (LFA).

- **IP address**: eth0 192.168.100.175
- **Host name**: collection.csite.ibm.edu associated with eth0

ha-proxy.csite.ibm.edu

This host runs the HAProxy load balancer.

- **IP address**: eth0 192.168.100.176
- **Host name**: ha-proxy.csite.ibm.edu associated with eth0

r-logstash.csite.ibm.edu

This host runs four Logstash receiver servers.

- **IP address**: eth0 192.168.100.177
- **Host name**: r-logstash.csite.ibm.edu associated with eth0

kafka.csite.ibm.edu

This host runs Kafka, Zookeeper, and Kafka Manager.

- **IP address**: eth0 192.168.100.178
- **Host name**: kafka.csite.ibm.edu associated with eth0

s-logstash.csite.ibm.edu

This host runs two Logstash sender servers.

- **IP address**: eth0 192.168.100.179
- **Host name**: s-logstash.csite.ibm.edu associated with eth0

log-analysis.csite.ibm.edu

This host runs the IBM Operations Analytics Log Analysis software.

- **IP address**: eth0 192.168.100.180
- **Host name**: log-analysis.csite.ibm.edu associated with eth0

Software

The lab image is configured with the following software:

- Filebeat version 1.1.1
- IBM Tivoli Log File Agent version 6.3.0
- HAProxy version 1.5.4
- Logstash version 2.2.1
- Kafka version 0.8.2.2
- Kafka Manager version 1.3.0.8
- IBM Operations Analytics Log Analysis version 1.3.3.1 (fix pack 1)

User IDs and passwords

The user IDs and passwords for this lab are listed in the following table.

Type	User ID	Password	Usage
Linux	netcool	object00	Linux user with sudo access who owns all software
IBM Operations Analytics Log Analysis application user	unityadmin	object00	Log Analysis super user

Starting and stopping the software

Use the following commands to start and stop Filebeat. Start and stop this software as the **netcool** user.

```
/opt/filebeat-1.1.1-x86_64/filebeat &

pkill -f filebeat
```

Use the following commands to start and stop the Tivoli Log File Agent. Start and stop this software as the **netcool** user.

```
/opt/LFA/IBM-LFA-6.30/bin/itmcmd agent -o default_workload_instance -f start lo

/opt/LFA/IBM-LFA-6.30/bin/itmcmd agent -o default_workload_instance -f stop lo
```

Use the following commands to start and stop HAProxy. Start and stop this software as the **netcool** user.

```
sudo service haproxy start

sudo service haproxy stop
```

Use the following commands to start and stop Zookeeper. Start and stop this software as the **netcool** user.

```
/opt/kafka_2.9.1-0.8.2.2/bin/zookeeper-server-start.sh
/opt/kafka_2.9.1-0.8.2.2/config/zookeeper.properties &

pkill -f zookeeper
```

Use the following commands to start and stop Kafka. Start and stop this software as the **netcool** user.

```
/opt/kafka_2.9.1-0.8.2.2/bin/kafka-server-start.sh -daemon
/opt/kafka_2.9.1-0.8.2.2/config/server.properties

/opt/kafka_2.9.1-0.8.2.2/bin/kafka-server-stop.sh
```

Use the following commands to start and stop Kafka Manager. Start and stop this software as the **netcool** user.

```
/opt/kafka-manager-1.3.0.8/bin/kafka-manager
-Dkafka-manager.zkhosts="kafka.csite.ibm.edu:17981" &

pkill -f kafka-manager
```

Use the following commands to start and stop IBM Operations Analytics Log Analysis. Start and stop this software as the **netcool** user.

```
/opt/IBM/LogAnalysis/utilities/unity.sh -start

/opt/IBM/LogAnalysis/utilities/unity.sh -stop
```

The commands to start and stop Logstash vary depending on the specific Logstash server you want to restart. The commands to stop each Logstash server are provided in the exercise guide instructions.

Commonly used URLs

Use the following URL to access the HAProxy statistics page:

http://ha-proxy.csite.ibm.edu:9000/haproxy_stats

Use the following URL to access the Kafka Manager page:

http://kafka.csite.ibm.edu:9000/

Use the following URL to access the IBM Operations Analytics Log Analysis user interface:

https://log-analysis.csite.ibm.edu:9987/Unity

Configuration files

You create several configuration files during these exercises. You can find a finished copy of each configuration file in the `/software/LabFiles/configs` directory of each host.

Unit 1 Overview exercises

This unit has no student exercises.

Unit 2 Filebeat, HAProxy, and Logstash receivers exercises

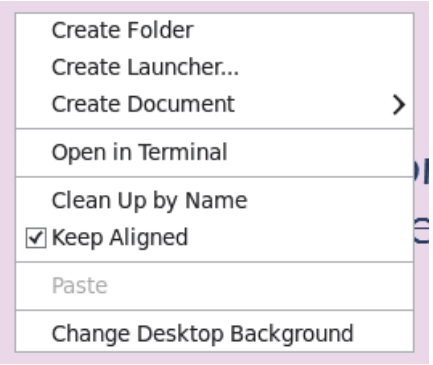
In the exercises for this unit, you install and configure a log collection agent, the HAProxy load balancer, and a Logstash receiver cluster. You configure these components to monitor a web access log from an IBM HTTP Server (IHS).

Exercise 1 Installing and configuring Filebeat

You use Filebeat to capture messages from a log file and forward them for processing. In this exercise, you install and configure an instance of Filebeat.

Important: Run all of the steps in this exercise on the host named **collection.csite.ibm.edu** as the **netcool** user.

1. The Filebeat software is included with Log Analysis. Copy the Filebeat software from the Log Analysis server to the collection host.
   a. Open a terminal window on the host that is named **collection.csite.ibm.edu**. Right-click the desktop and click **Open In Terminal**.



b. Run the following command to verify that you are working on the correct host. You should be working on the host named **collection.csite.ibm.edu**.

```
hostname

collection.csite.ibm.edu
```

c. Run the following commands to create a directory for the Filebeat installation file and to change into the new directory.

```
mkdir /software/filebeat

cd /software/filebeat
```

d. Run the following command to copy the Filebeat installation file from the Log Analysis server. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**.

```
scp
netcool@192.168.100.180:/opt/IBM/LogAnalysis/filebeat/filebeat-1.1.1.tar.gz
.


The authenticity of host '192.168.100.180 (192.168.100.180)' can't be
established.
RSA key fingerprint is a7:09:f9:fc:ec:62:ad:6e:69:2a:d3:7a:2d:e5:d8:a0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.180' (RSA) to the list of known
hosts.
netcool@192.168.100.180's password: object00
```

2. Run the following command to decompress the installation file into the **/opt/** directory.

```
tar -zxvf filebeat-1.1.1.tar.gz -C /opt/
```

3. Configure Filebeat to monitor an IBM HTTP Server access log file. The full path to the log file you monitor is `/software/log_samples/IHS_logs/Dallas-IHS-access.log`.

Note: You configure Filebeat by editing the `filebeat.yml` file.

a. Run the following command to change to the Filebeat directory.

```
cd /opt/filebeat-1.1.1-x86_64/
```

b. Run the following command to change the name of the default configuration file to `filebeat.yml.OLD`.

```
mv filebeat.yml filebeat.yml.OLD
```

c. Create and edit a Filebeat configuration file named `filebeat.yml` in a text editor. This example uses vi.

```
vi /opt/filebeat-1.1.1-x86_64/filebeat.yml
```

d. Add the following lines to your `filebeat.yml` file.

```
filebeat:
 prospectors:
 -
  paths:
   - /software/log_samples/IHS_logs/Dallas-IHS-access.log
  input_type: log

output:
 logstash:
  hosts: ["ha-proxy.csite.ibm.edu:20737"]

shipper:

logging:
 to_files: true
 files:
  path: /opt/filebeat-1.1.1-x86_64/
  name: mybeat
  rotateeverybytes: 10485760
 level: debug
```

e. Save and close the file when you are finished.

Important: The indentations in the `filebeat.yml` file are very important. This file is indented with spaces, not tabs.

```
filebeat:
 prospectors:
 -
  paths:
   - /software/log_samples/IHS_logs/Dallas-IHS-access.log
  input_type: log

output:
 logstash:
  hosts: ["ha-proxy.csite.ibm.edu:20737"]

shipper:

logging:
 to_files: true
 files:
  path: /opt/filebeat-1.1.1-x86_64/
  name: mybeat
  rotateeverybytes: 10485760
 level: debug
```

These lines configure the locaton of the target log file

These lines send messages from the target log file to the HAProxy server

These are instructions to log Filebeat operations to a log file named `mybeat` and to set the log level

Figure 1 Key fields in filebeat.yml

Note: Later in these exercises, you configure HAProxy to forward all messages to a Logstash receiver cluster. This is the reason that you are using the Logstash output option.

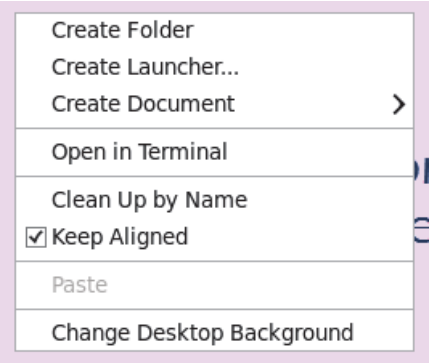
4. Run the following command to start Filebeat.

```
/opt/filebeat-1.1.1-x86_64/filebeat &
```

5. Run the following command to verify that Filebeat is running.

```
ps -ef | grep -i filebeat

netcool   2796  2415  0 19:54 pts/0    00:00:00
/opt/filebeat-1.1.1-x86_64/filebeat
```

6. Verify that Filebeat is reading messages from the target log file.
   a. Open a new terminal window in the **collection.csite.ibm.edu** host. Right-click the desktop and click **Open In Terminal**.



   b. Run the following command to watch for activity in the Filebeat log file.

```
tail -f /opt/filebeat-1.1.1-x86_64/mybeat
```

   c. Return to the first terminal window. Run the following command to add more messages to the `/software/log_samples/IHS_logs/Dallas-IHS-access.log` file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

   d. Return to the terminal window that shows the activity in the Filebeat log file. Look for a message like the following example. Messages like this one verify that Filebeat is reading messages from the target log file.

```
2016-06-08T12:03:14Z DBG  full line read
```

   e. Press Ctrl + C to stop the `tail` command.

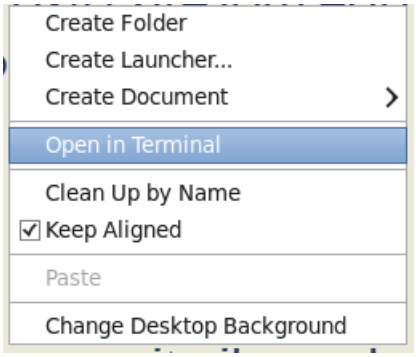
Note: You can ignore any connection refused messages: `Connecting error publishing events (retrying): dial tcp 192.168.100.176:20737: getsockopt: connection refused`. This error is present because you have not configured the HAProxy and Logstash receiver components yet.

Exercise 2 Installing and configuring HAProxy

HAProxy is a load balancer and proxy for TCP and HTTP-based applications. In this exercise, you install and configure HAProxy. Your instance of HAProxy listens for traffic from Filebeat and forwards it to a cluster of Logstash receivers. You configure the Logstash receivers later.

Important: Run all of the steps in this exercise on the host named **ha-proxy.csite.ibm.edu** as the **netcool** user.

1. Decompress the HAProxy installation files.
   a. Open a terminal window on the host that is named **ha-proxy.csite.ibm.edu**. Right-click the desktop and click **Open In Terminal**.



   b. Run the following command to verify that you are working on the correct host. You should be working on the host named **ha-proxy.csite.ibm.edu**.

```
hostname

ha-proxy.csite.ibm.edu
```

   c. Run the following command to change to the correct directory.

```
cd /software/haproxy_installer/
```

   d. Run the following command to decompress the HAProxy package.

```
tar -zxvf haproxy-1.5.4.tar.gz
```

Note: In many environments, you can use software installation managers such as Yum or YaST to install HAProxy. However, in your lab environment you do not have Internet access.

2. Follow the next steps to compile and install HAProxy from the source code you previously decompressed.
   a. Run the following commands to compile HAProxy for your environment.

      ```
      cd /software/haproxy_installer/haproxy-1.5.4

      make TARGET=linux2628
      ```

   b. Wait for the previous command to finish running. Run the following command to install HAProxy.

      ```
      sudo make install
      ```

3. Run the following command to copy the HAProxy binary files to the `/usr/sbin` directory.

   ```
   sudo cp /usr/local/sbin/haproxy /usr/sbin/
   ```

4. Configure HAProxy to start automatically when the server starts up.
   a. Run the following command to copy the sample init file for HAProxy to the system init directory:

      ```
      sudo cp /software/haproxy_installer/haproxy-1.5.4/examples/haproxy.init
      /etc/init.d/haproxy
      ```

   b. Run the following command to modify the file permissions of the init file.

      ```
      sudo chmod 755 /etc/init.d/haproxy
      ```

   c. Run the following commands to enable the HAProxy service.

      ```
      cd /etc/init.d

      sudo chkconfig haproxy on
      ```

5. Run the following commands to create the directories for the HAProxy configuration file and statistics page.

   ```
   sudo mkdir -p /etc/haproxy

   sudo mkdir -p /run/haproxy

   sudo mkdir -p /var/lib/haproxy

   sudo touch /var/lib/haproxy/stats
   ```

6. Run the following command to add the **haproxy** user account. This user account is required for HAProxy to run.

   ```
   sudo useradd -r haproxy
   ```

7. Run the following command to verify that HAProxy was installed correctly:

   ```
   sudo haproxy -v

   HA-Proxy version 1.5.4 2014/09/02
   ```

8. An example HAProxy configuration file is included with Log Analysis. Run the following commands to copy the example HAProxy configuration file from the Log Analysis server to the ha-proxy host. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**.

```
cd /etc/haproxy/

sudo scp
netcool@192.168.100.180:/opt/IBM/LogAnalysis/kafka/test-configs/haproxy-configs
/haproxy.cfg .

The authenticity of host '192.168.100.180 (192.168.100.180)' can't be
established.
RSA key fingerprint is a7:09:f9:fc:ec:62:ad:6e:69:2a:d3:7a:2d:e5:d8:a0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.180' (RSA) to the list of known hosts.
netcool@192.168.100.180's password: object00
```

9. Configure HAProxy logging.
   a. Open the `/etc/haproxy/haproxy.cfg` file in a text editor. This example uses `vi`.

      ```
      sudo vi /etc/haproxy/haproxy.cfg
      ```

   b. Find the following lines:

      ```
      log         127.0.0.1 local0
      log         127.0.0.1 local1
      log         127.0.0.1 local2
      log         127.0.0.1 local3
      ```

   c. Add a comment character in front of the **local1**, **local2**, and **local3** lines.

      ```
      log         127.0.0.1 local0
      #log         127.0.0.1 local1
      #log         127.0.0.1 local2
      #log         127.0.0.1 local3
      ```

10. Exclude the entire section for the Log File Agent listener.
    a. Find the following lines:

      ```
      listen  LFA_Cluster co9122123210.in.ibm.com:20738
        mode tcp
        balance roundrobin
        hash-type consistent
        server receiver1-linux-x64 co9122123210.in.ibm.com:18738 check inter 1s fall 2 rise 3
        server receiver1-linux-s390 9.42.6.72:18738 check inter 1s fall 2 rise 3
      ```

b. Add a comment character in front of all the lines in the Log File Agent listener section.

```
#listen  LFA_Cluster co9122123210.in.ibm.com:20738
  #mode tcp
  #balance roundrobin
  #hash-type consistent
  #server receiver1-linux-x64 co9122123210.in.ibm.com:18738 check inter 1s fall 2 rise
3
  #server receiver1-linux-s390 9.42.6.72:18738 check inter 1s fall 2 rise 3
```

11. Exclude the entire section for the Syslog Logstash cluster listener.

a. Find the following lines:

```
listen  Syslog_Cluster co9122123210.in.ibm.com:20739
 mode tcp
 balance roundrobin
 server receiver1-linux-x64 co9122123210.in.ibm.com:18739 check fall 2 rise 3 inter 1000
 server receiver1-linux-s390 9.42.6.72:18739 check fall 2 rise 3 inter 1000
```

b. Add a comment character in front of all the lines in the Syslog Logstash cluster section.

```
#listen  Syslog_Cluster co9122123210.in.ibm.com:20739
 #mode tcp
 #balance roundrobin
 #server receiver1-linux-x64 co9122123210.in.ibm.com:18739 check fall 2 rise 3 inter 100
 #server receiver1-linux-s390 9.42.6.72:18739 check fall 2 rise 3 inter 1000
```

12. Edit the Filebeat Logstash cluster section to listen for traffic from the Filebeat server and forward the traffic to two instances of Logstash.

Note: You installed and configured Filebeat in the preceding exercise. You install and configure the two instances of Logstash in the next exercise.

a. Find the following line:

```
listen  Beats_Cluster co9122123210.in.ibm.com:20737
```

b. Change the host name to **ha-proxy.csite.ibm.edu**.

```
listen  Beats_Cluster ha-proxy.csite.ibm.edu:20737
```

Hint: Remember, you configured your Filebeat `filebeat.yml` file to send messages to ha-proxy.csite.ibm.edu:20737.

c. Find the following line:

```
balance source
```

d. Change the balance mode to **roundrobin**. This mode configures HAProxy to send traffic to each Logstash server in turns.

```
balance roundrobin
```

e. Find the following line:

```
hash-type consistent
```

f. Add a comment character in front of the hash-type line.

```
#hash-type consistent
```

g. Find the following line:

```
server receiver1-linux-x64 co9122123210.in.ibm.com:18737 check fall 2 rise 3
inter 1000
```

h. Change the server name, host name, and port number to **receiver-logstashA**, **r-logstash.csite.ibm.edu**, and **18737**. These settings send traffic to a Logstash server on port 18737. You install and configure this Logstash server in the next exercises.

```
server receiver-logstashA r-logstash.csite.ibm.edu:18737 check fall 2 rise 3
inter 1000
```

i. Find the following line:

```
server receiver1-linux-s390 9.42.6.72:18737 check fall 2 rise 3 inter 1000
```

j. Change the server name, host name, and port number to **receiver-logstashB**, **r-logstash.csite.ibm.edu**, and **18738**. These settings send traffic to a Logstash server on port 18738. You install and configure this Logstash server in the next exercises.

```
server receiver-logstashB r-logstash.csite.ibm.edu:18738 check fall 2 rise 3
inter 1000
```

13. Configure the HAProxy statistics page. You can look at the statistics page to view the status of the servers in your configuration and details about data that has been forwarded to them.

a. Add the following lines to the bottom of the file.

```
listen stats :9000
    mode http
    stats enable
    stats hide-version
    stats uri /haproxy_stats
```

b. Save and close the file when you are finished.

14. Configure rsyslog on the ha-proxy host to manage log messages from HAProxy.

a. Create and edit a new file in the `/etc/rsyslog.d/` directory named `haproxy.conf`. This example uses `vi`.

```
sudo vi /etc/rsyslog.d/haproxy.conf
```

b. Add the following line in the file.

```
if ($programname == 'haproxy') then -/var/log/haproxy.log
```

c. Save and close the file when you are finished.

d. Open the `/etc/rsyslog.conf` file in a text editor.

```
sudo vi /etc/rsyslog.conf
```

e. Find the following lines:

```
#$ModLoad imudp
#$UDPServerRun 514
```

f. Remove the comment character from these two lines.

```
$ModLoad imudp
$UDPServerRun 514
```

g. Add the following line below the lines you just edited.

```
$UDPServerAddress 127.0.0.1
```

h. Save and close the file when you are finished.

15. Run the following command to restart the rsyslog service.

```
sudo service rsyslog restart
```

16. Run the following command to start HA Proxy.

```
sudo service haproxy start
```

Hint: You can ignore the message about `Beats_Cluster has no server available!` This message is present because you have not installed any Logstash servers yet.

17. Run the following command to view the HAProxy log file.

```
sudo tail -50 /var/log/haproxy.log
```

```
May 17 17:25:11 localhost haproxy[6845]: Server
Beats_Cluster/receiver-logstashA is DOWN, reason: Layer4 connection problem,
info: "Connection refused", check duration: 0ms. 1 active and 0 backup servers
left. 0 sessions active, 0 requeued, 0 remaining in queue.

Message from syslogd@localhost at May 17 17:25:12 ...
 haproxy[6846]: proxy Beats_Cluster has no server available!
May 17 17:25:12 localhost haproxy[6846]: Server
Beats_Cluster/receiver-logstashB is DOWN, reason: Layer4 connection problem,
info: "Connection refused", check duration: 0ms. 0 active and 0 backup servers
left. 0 sessions active, 0 requeued, 0 remaining in queue.
May 17 17:25:12 localhost haproxy[6846]: proxy Beats_Cluster has no server
available!
```

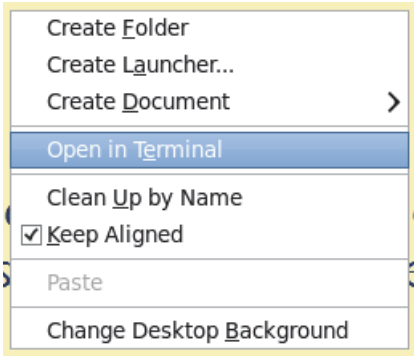
The messages `receiver-logstashA is DOWN` and `receiver-logstashB is DOWN` indicate that HAProxy cannot connect to any Logstash servers. In the next exercises, you install and configure these two Logstash servers.

Exercise 3 Installing and configuring the first Logstash receiver

Logstash receivers accept log messages from the data collection agent, add metadata to the log messages, and forward them to the Kafka message broker server. In this exercise, you install and configure the first of two Logstash receiver servers.

Important: Run all of the steps in this exercise on the host named **r-logstash.csite.ibm.edu** as the **netcool** user.

1. The Logstash software is included with Log Analysis. Copy the Logstash software from the Log Analysis server to the receiver Logstash host.
   a. Open a terminal window on the host that is named **r-logstash.csite.ibm.edu**. Right-click the desktop and click **Open In Terminal**.



   b. Run the following command to verify that you are working on the correct host. You should be working on the host named **r-logstash.csite.ibm.edu**.

```
hostname
```

```
r-logstash.csite.ibm.edu
```

   c. Run the following commands to create a directory for the Logstash installation file and to change into the new directory.

```
mkdir /software/logstash
```

```
cd /software/logstash/
```

d. Run the following command to copy the Logstash installation file from the Log Analysis server. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**.

```
scp
netcool@192.168.100.180:/opt/IBM/LogAnalysis/logstash-2.2.1/logstash-2.2.1.t
ar.gz .


The authenticity of host '192.168.100.180 (192.168.100.180)' can't be
established.
RSA key fingerprint is a7:09:f9:fc:ec:62:ad:6e:69:2a:d3:7a:2d:e5:d8:a0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.180' (RSA) to the list of known
hosts.
netcool@192.168.100.180's password: object00
```

2. Run the following command to create a directory for the first Logstash receiver. In the topology you are building for this lab, the first Logstash receiver instance is called **logstashA**.

```
mkdir /opt/logstashA
```

3. Run the following command to decompress the installation file and install Logstash into the `/opt/logstashA` directory.

```
tar -zxvf logstash-2.2.1.tar.gz -C /opt/logstashA/
```

4. Create a directory for the logstashA configuration file.

```
mkdir /opt/logstashA/logstash-2.2.1/conf
```

5. Create a directory for the logstashA log file.

```
mkdir /opt/logstashA/logstash-2.2.1/log
```

6. Configure the logstashA instance to accept messages from the HAProxy server and output them to a log file. Remember you configured HAProxy to forward messages to `r-logstash.csite.ibm.edu:18737`.

   a. Create and edit a Logstash configuration file named `logstashA.conf` in a text editor. Create the file in the `/opt/logstashA/logstash-2.2.1/conf` directory. This example uses `vi`.

```
vi /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
```

b. Add the following lines to your `logstashA.conf` file.

```
input {

    beats {
       port => 18737
    } #end beats input

} #end input section

filter {
} #end filter section

output {

    file {
        path => "/opt/logstashA/logstash-2.2.1/log/logstashA-debug.log"
        codec => rubydebug
    } #end file

} # end output section
```

c. Save and close the file when you are done.

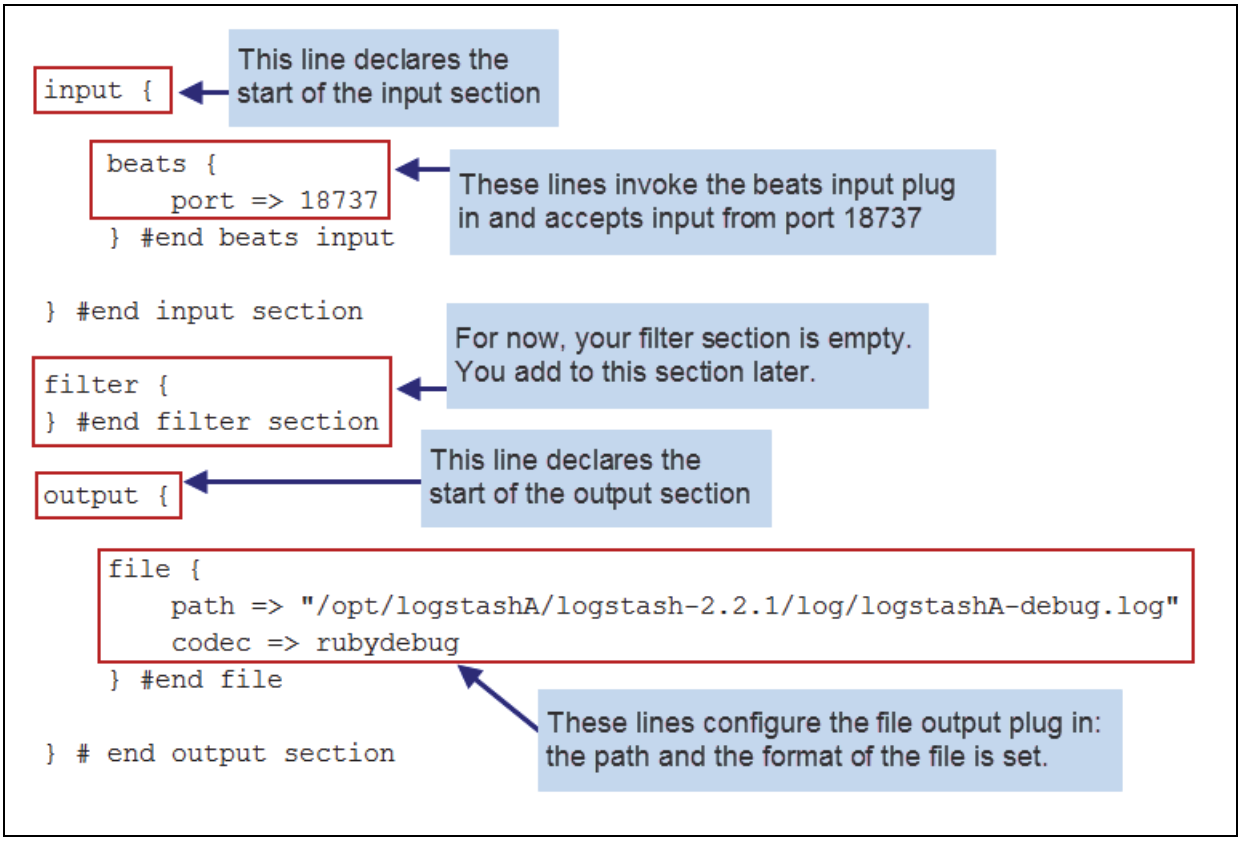


Figure 2 Key fields in logstashA.conf

7. Run the following command to start the Logstash server named logstashA. Run the entire command on one line.

```
/opt/logstashA/logstash-2.2.1/bin/logstash -f
/opt/logstashA/logstash-2.2.1/conf/logstashA.conf -l
/opt/logstashA/logstash-2.2.1/log/logstashA-debug.log &
```

8. Run the following command to verify that the logstashA server is running.

```
ps -ef | grep logstashA

netcool   5090  2496 34 19:54 pts/0    00:00:29 /usr/bin/java -XX:+UseParNewGC
-XX:+UseConcMarkSweepGC -Djava.awt.headless=true
-XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly
-XX:+HeapDumpOnOutOfMemoryError -Xmx1g -Xss2048k
...
```

Exercise 4 Verifying data flow

In this exercise, you verify that log messages are flowing from the Filebeat collector, through the HAProxy server, and then through the first Logstash receiver instance.

Important: You use three different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **r-logstash.csite.ibm.edu**. You installed the first instance of the Logstash receiver (logstashA) on this host.

2. Run the following command to watch for activity in the logstashA log file. Leave the `tail` command running.

```
tail -f /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
```

3. Go to the host named **ha-proxy.csite.ibm.edu**. You installed the HAProxy server on this host.

4. Run the following command to watch for activity in the HAProxy log file. Leave the `tail` command running.

```
sudo tail -f /var/log/haproxy.log
```

5. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

6. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

7. Go to the host named **r-logstash.csite.ibm.edu**. Look at the `logstashA-debug.log` file. Look for messages like the following example. Messages like these verify that logstashA is receiving log messages from the Filebeat server.

```
       "message" => "Apache/IHS,192.168.2.94,-,-,[10/May/2016:15:41:37
-0400],GET,\"GET /daytrader/app?action=quotes&symbols=s:0,s:1,s:2,s:3,s:4
HTTP/1.1\",200,3409,1281,\"-\",\"curl/7.21.3 (x86_64-unknown-linux-gnu)
libcurl/7.21.3 OpenSSL/1.0.0 zlib/1.2.3\"",
      "@version" => "1",
    "@timestamp" => "2016-05-17T20:23:43.641Z",
          "beat" => {
        "hostname" => "collection.csite.ibm.edu",
            "name" => "collection.csite.ibm.edu"
    },
         "count" => 1,
        "fields" => nil,
    "input_type" => "log",
        "offset" => 466341,
        "source" => "/software/log_samples/IHS_logs/Dallas-IHS-access.log",
          "type" => "log",
          "host" => "collection.csite.ibm.edu",
          "tags" => [
        [0] "beats_input_codec_plain_applied"
```

This is the original message from the target log file

```
    "message" =>
    "Apache/IHS,192.168.2.94,-,-,[10/May/2016:15:41:37
    -0400],GET,\"GET
    /daytrader/app?action=quotes&symbols=s:0,s:1,s:2,s:3,s:4
    HTTP/1.1\",200,3409,1281,\"-\",\"curl/7.21.3
    (x86_64-unknown-linux-gnu) libcurl/7.21.3 OpenSSL/1.0.0
    zlib/1.2.3\"",
   "@version" => "1",
 "@timestamp" => "2016-05-17T20:23:43.641Z",
       "beat" => {
     "hostname" => "collection.csite.ibm.edu",
         "name" => "collection.csite.ibm.edu"
 },
      "count" => 1,
     "fields" => nil,
 "input_type" => "log",
     "offset" => 466341,
     "source" =>
     "/software/log_samples/IHS_
       "type" => "log",
       "host" => "collection.csite.ibm.edu",
       "tags" => [
     [0] "beats_input_codec_plain_applied"
```

This is a list of meta data tags that are attached to the log message

Figure 3 Key fields in the Logstash log file

8. After a few moments, Logstash receives a signal that Filebeat has reached the end of the target file, and the connection is closed. Look for a message like the following example.

   `{:timestamp=>"2016-05-17T20:24:46.734000+0000", :message=>"Beats Input: Remote connection closed", :peer=>"192.168.100.176:51010", :exception=>#<Lumberjack::Beats::Connection::ConnectionClosed: Lumberjack::Beats::Connection::ConnectionClosed wrapping: EOFError, `**`End of file reached`**`>, :level=>:warn}`

9. Press Ctrl + C to stop the `tail` of the `logstashA-debug.log` file.

10. Go to the host named **ha-proxy.csite.ibm.edu**. Look at the `haproxy.log` file. Look for a message like the following example. Messages like these verify that HAProxy has completed a TCP transaction. These messages arrive only after the transaction is complete and the session is closed.

    ```
    May 17 20:24:46 localhost haproxy[6846]: 192.168.100.175:38304
    [17/May/2016:20:23:45.773] Beats_Cluster Beats_Cluster/receiver-logstashA
    1/0/60809 60 cD 0/0/0/0/0 0/0
    ```

11. Press Ctrl + C to stop the `tail` of the `haproxy.log` file.

12. Use the HAProxy statistics page to verify that data has been forwarded by the server.
    a. Go to the host named **ha-proxy.csite.ibm.edu**. Open a Firefox browser.

    b. Browse to the following address.

    `http://ha-proxy.csite.ibm.edu:9000/haproxy_stats`

    Notice the following information that you can view on the HAProxy statistics page:

    - You can see the status of each end point. The logstashA server is up and the logstashB server is down.
    - You can see the volume of traffic that has been forwarded to each end point.
    - You can see the number of current connections to each end point.

Beats_Cluster

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status
Frontend				0	1	-	0	1	3 000	151			27 833	516	0	0	0					OPEN
receiver-logstashA	0	0	-	0	1		0	1	-	1	1	13m25s	27 833	516		0		0	0	0	0	14m17s UP
receiver-logstashB	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	2h43m DOWN
Backend	0	0		0	1		0	1	300	151	1	13m25s	27 833	516	0	0		150	0	0	0	14m17s UP

Exercise 5 Adding metadata to the log message

With Filebeat and Logstash, you can attach metadata to each message in the target log file. This metadata is useful when you add conditional logic to your Logstash configuration file, when you configure Kafka, and when you create the data source in the Log Analysis core software. In this exercise, you change the configuration of Filebeat and Logstash to add metadata.

Important: You use two different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **collection.csite.ibm.edu**. This is the host where you installed Filebeat.
2. Configure Filebeat to add the following metadata fields to the target log file:
   - **collector**: filebeats-collection.csite.ibm.edu
   - **env**: DEV
   - **module**: IBM-HTTP-Server
   - **type**: access-log
   - **site**: DALLAS
   - **platform**: RHEL

   a. Open the `/opt/filebeat-1.1.1-x86_64/filebeat.yml` file in a text editor. This example uses `vi`.

   ```
   vi /opt/filebeat-1.1.1-x86_64/filebeat.yml
   ```

   b. Find the following line:

   ```
   input_type: log
   ```

   c. Add the following lines in bold typeface below the **input_type** line.

   ```
   input_type: log
   fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: DALLAS
    platform: RHEL
   ```

   d. Save and close the file when you are finished.

Important: Remember, indentation in the `filebeat.yml` file is important. This file is indented with spaces, not tabs.

- Ensure that the `fields:` line is indented to the same level as the `input_type:` line.
- Ensure that each of the actual field names, such as `collector` and `env`, are indented past the `fields:` line.

The following list explains the fields that you added to your configuration:

- **collector**: This field identifies the collection agent that is monitoring the target log file. You use this field for conditional logic in the Logstash receiver configuration.
- **env**: This field identifies the environment where the application that generates the target log is operating, such as development or production. You use this field to configure the output to Kafka and to create a data source in the Log Analysis core.
- **module**: This field identifies the application generates the target log. You use this field to configure the output to Kafka and to create a data source in the Log Analysis core.
- **type**: This field identifies the type of log file that you are monitoring, such as access log or error log. You use this field to create a data source in the Log Analysis core.
- **site**: This field identifies the geographic location of the application that generates the target log. You use this field later in the course.
- **platform**: This field identifies the operating system where the logging application is running. You use this field later in the course.

Note: The metadata fields that you add and use in this lab are like fields you might add in a production environment, but they are only examples. There are no strict requirements or guidelines concerning the metadata that you need in a production environment.

3. Restart Filebeat so that it uses the new configuration.
   a. Run the following command to stop Filebeat.

      ```
      pkill -f filebeat
      ```

   b. Run the following command to start Filebeat.

      ```
      /opt/filebeat-1.1.1-x86_64/filebeat &
      ```

4. Run the following command to add more messages to the target log file.

   ```
   /software/log_samples/scripts/Dallas_Web_Logs.sh
   ```

5. Go to the host named **r-logstash.csite.ibm.edu**. You installed the first instance of the Logstash receiver (logstashA) on this host.

6. Confirm that the logstashA server received the additional metadata fields.
   a. Run the following command to see the most recent messages.

      ```
      tail -100 /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
      ```

   b. Look for messages like the following example. Notice that the six new fields are present: **collector**, **env**, **module**, **type**, **site**, and **platform**.

      ```
      "message" => "Apache/IHS,192.168.2.94,-,-,[10/May/2016:15:41:37
      -0400],GET,\"GET /daytrader/app?action=buy&symbol=s%3A0&quantity=100
      HTTP/1.1\",200,3409,1184,\"-\",\"curl/7.21.3 (x86_64-unknown-linux-gnu)
      libcurl/7.21.3 OpenSSL/1.0.0 zlib/1.2.3\"",
            "@version" => "1",
          "@timestamp" => "2016-05-19T15:25:00.705Z",
                "beat" => {
              "hostname" => "collection.csite.ibm.edu",
                  "name" => "collection.csite.ibm.edu"
          },
               "count" => 1,
              "fields" => {
              "collector" => "filebeats-collection.csite.ibm.edu",
                    "env" => "DEV",
                 "module" => "IBM-HTTP-Server",
               "platform" => "RHEL",
                   "site" => "DALLAS",
                   "type" => "access-log"
          },
          "input_type" => "log",
              "offset" => 559931,
              "source" => "/software/log_samples/IHS_logs/Dallas-IHS-access.log",
                "type" => "log",
                "host" => "collection.csite.ibm.edu",
                "tags" => [
              [0] "beats_input_codec_plain_applied"
      ```

7. Edit the logstashA server configuration to use the additional fields.
   a. Edit the `logstashA.conf` file. This example uses `vi`.

      ```
      vi /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
      ```

b. Add the following lines in bold typeface to the filter section of the `logstashA.conf` file.

```
filter {

    if [fields][collector] == "filebeats-collection.csite.ibm.edu" {
        mutate {
          add_field => [ "datasource",
"%{[fields][env]}_%{[fields][module]}_%{[fields][type]}" ]
          add_field => [ "resourceID", "%{[beat][hostname]}_%{source}_1" ]
          add_tag => "mutate_filebeat"
        }# end mutate
    } #end filebeat condition

} #end filter section
```

c. Save and close the file when you are done.

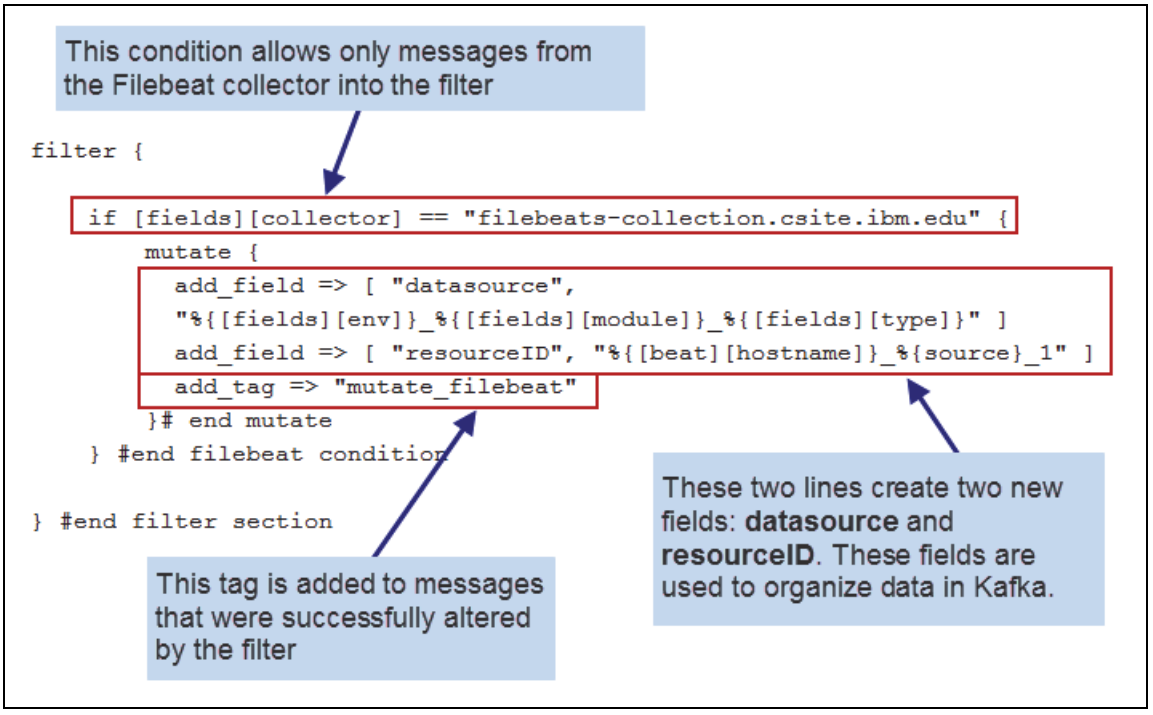


Figure 4 Key fields in logstashA.conf

8. Run the following command to stop the logstashA server.

```
pkill -f logstashA
```

9. In these exercises, you start and stop the logstashA server many times as you build your configuration. Add a command alias to make it easier to start Logstash.

   a. Open the `.bashrc` file of the **netcool** user in a text editor.

```
vi /home/netcool/.bashrc
```

b. Add the following line to the bottom of the file.

```
alias startlogstashA='/opt/logstashA/logstash-2.2.1/bin/logstash -f
/opt/logstashA/logstash-2.2.1/conf/logstashA.conf -l
/opt/logstashA/logstash-2.2.1/log/logstashA-debug.log &'
```

c. Save and close the file when you are finished.

d. Run the following command to source the modified environment file.

```
source /home/netcool/.bashrc
```

e. Run the following command to start the logstashA server.

```
startlogstashA
```

10. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

11. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

12. Verify that your logstashA server is using the new filter and adding the additional fields.

a. Go to the host named **r-logstash.csite.ibm.edu**. You installed the first instance of the Logstash receiver (logstashA) on this host.

b. Run the following command to see the most recent messages in the `logstashA-debug.log` file.

```
tail -100 /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
```

c. Look for messages like the following example. Confirm that you see the following fields and the filebeat tag: **datasource**, **resourceID**, and **mutate_filebeat**.

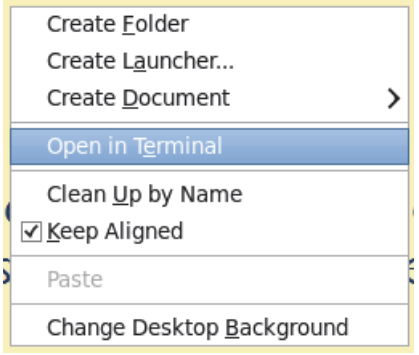
```
"input_type" => "log",
        "offset" => 1586436,
        "source" => "/software/log_samples/IHS_logs/Dallas-IHS-access.log",
          "type" => "log",
          "host" => "collection.csite.ibm.edu",
          "tags" => [
        [0] "beats_input_codec_plain_applied",
        [1] "mutate_filebeat"
    ],
    "datasource" => "DEV_IBM-HTTP-Server_access-log",
    "resourceID" =>
"collection.csite.ibm.edu_/software/log_samples/IHS_logs/Dallas-IHS-access.l
og_1"
```

Exercise 6 Installing and configuring the second Logstash receiver

In a preceding exercise, you installed and configured a Logstash receiver instance named logstashA. In this exercise, you add redundancy to your environment by installing and configuring a second Logstash receiver named logstashB.

Important: You use three different hosts in this exercise. Pay careful attention to the host you are working on when you complete each step.

1. Verify that you are working on the correct host.
   a. Open a terminal window on the host named **r-logstash.csite.ibm.edu** if you do not already have one open. Right-click the desktop and click **Open In Terminal**.



   b. Run the following command to verify that you are working on the correct host. You should be working on the host named **r-logstash.csite.ibm.edu**.

```
hostname

r-logstash.csite.ibm.edu
```

2. Run the following command to create a directory for the second Logstash receiver. In the topology you are building for this lab, the second Logstash receiver instance is called **logstashB**.

```
mkdir /opt/logstashB
```

3. Run the following commands to extract the Logstash installation file and install Logstash into the */opt/logstashB* directory. Remember, you copied the Logstash installation file from the Log Analysis server in a preceding exercise.

```
cd /software/logstash/

tar -zxvf logstash-2.2.1.tar.gz -C /opt/logstashB/
```

4. Create a directory for the logstashB configuration file.

   ```
   mkdir /opt/logstashB/logstash-2.2.1/conf
   ```

5. Create a directory for the logstashB log file.

   ```
   mkdir /opt/logstashB/logstash-2.2.1/log
   ```

6. Configure the logstashB server.
   a. The configuration of the logstashB server is almost identical to the logstashA configuration. Run the following command to copy the logstashA configuration to logstashB. Run the entire command on one line.

      ```
      cp /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
      /opt/logstashB/logstash-2.2.1/conf/logstashB.conf
      ```

   b. Open the logstashB configuration file in a text editor.

      ```
      vi /opt/logstashB/logstash-2.2.1/conf/logstashB.conf
      ```

   c. Find the following line:

      ```
      port => 18737
      ```

   d. Change the line to use port number **18738**.

      ```
      port => 18738
      ```

   e. Find the following line:

      ```
      path => "/opt/logstashA/logstash-2.2.1/log/logstashA-debug.log"
      ```

   f. Change the directory and name of the debug log file.

      ```
      path => "/opt/logstashB/logstash-2.2.1/log/logstashB-debug.log"
      ```

   g. Save and close the file when you are finished.

7. In these exercises, you start and stop the logstashB server many times as you build your configuration. Add a command alias to make it easier to start Logstash.
   a. Open the `.bashrc` file of the netcool user in a text editor.

      ```
      vi /home/netcool/.bashrc
      ```

   b. Add the following line to the bottom of the file.

      ```
      alias startlogstashB='/opt/logstashB/logstash-2.2.1/bin/logstash -f
      /opt/logstashB/logstash-2.2.1/conf/logstashB.conf -l
      /opt/logstashB/logstash-2.2.1/log/logstashB-debug.log &'
      ```

   c. Save and close the file when you are finished.

   d. Run the following command to source the modified environment file.

      ```
      source /home/netcool/.bashrc
      ```

   e. Run the following command to start the logstashB server.

      ```
      startlogstashB
      ```

f. Run the following command to verify that the logstashB server is running.

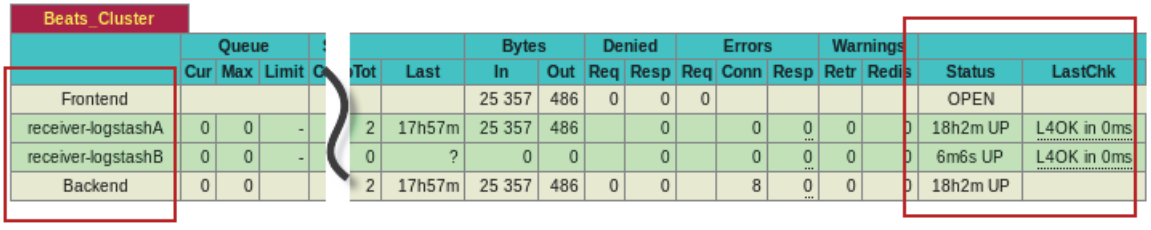
```
ps -ef | grep -i logstashB

netcool   6586  6426 34 13:39 pts/1    00:00:22 /usr/bin/java
-XX:+UseParNewGC -XX:+UseConcMarkSweepGC -Djava.awt.headless=true
-XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly
-XX:+HeapDumpOnOutOfMemoryError -Xmx1g -Xss2048k
-Djffi.boot.library.path=/opt/logstashB/logstash-2.2.1/vendor/jruby/lib/jn
...
```

g. Go to the host named **ha-proxy.csite.ibm.edu**. You installed the HAProxy server on this host. Open a Firefox browser.

h. Browse to the following address.

```
http://ha-proxy.csite.ibm.edu:9000/haproxy_stats
```

Notice that the HAProxy server can connect to both Logstash receiver instances.



Beats_Cluster

	Queue						Bytes		Denied		Errors			Warnings			
	Cur	Max	Limit	Cur	Tot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk
Frontend							25 357	486	0	0	0					OPEN	
receiver-logstashA	0	0	-		2	17h57m	25 357	486		0		0	0	0		18h2m UP	L4OK in 0ms
receiver-logstashB	0	0	-		0	?	0	0		0		0	0	0		6m6s UP	L4OK in 0ms
Backend	0	0			2	17h57m	25 357	486	0	0		8	0	0		18h2m UP	

8. Verify that HAProxy is balancing traffic between the logstashA and logstashB servers.

a. Go to the host named **r-logstash.csite.ibm.edu**.

b. In a terminal window, run the following command to watch for activity in the logstashA log file.

```
tail -f /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
```

c. In a different terminal window, run the following command to watch for activity in the logstashB log file.

```
tail -f /opt/logstashB/logstash-2.2.1/log/logstashB-debug.log
```

d. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

e. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

f. Go to the host named **r-logstash.csite.ibm.edu**. Look at the `tail` for both of the Logstash log files. One of the Logstash servers has received new messages from Filebeat.

g. Wait a moment until you see a message like the following example. Messages like this one indicate that the connection from Filebeat to the Logstash server is closed.

```
Beats::Connection::ConnectionClosed wrapping: EOFError, End of file reached
```

h. Return to the host named **collection.csite.ibm.edu**. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

i. Return to the host named **r-logstash.csite.ibm.edu**. Look at the `tail` for both of the Logstash log files. The other Logstash server has received the next set of messages from Filebeat.

j. Leave the `tail` of both Logstash server log files running.

9. Verify redundancy for the Logstash receivers.

a. Verify that you are working on the host named **r-logstash.csite.ibm.edu**.

b. Run the following command to stop the logstashA server. This action also stops the `tail` of the logstashA log file.

```
pkill -f logstashA
```

c. Go to the host named **collection.csite.ibm.edu**. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

d. Return to the host named **r-logstash.csite.ibm.edu**. Look at the `tail` of the logstashB log file. Verify that messages arrived from Filebeat.

e. Run the following command to start the logstashA server.

```
startlogstashA
```

f. Run the following command to watch for activity in the logstashA log file.

```
tail -f /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
```

g. Run the following command to stop the logstashB server. This action also stops the `tail` of the logstashB log file.

```
pkill -f logstashB
```

h. Go to the host named **collection.csite.ibm.edu**. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

i. Return to the host named **r-logstash.csite.ibm.edu**. Look at the `tail` of the logstashA log file. Verify that messages arrived from Filebeat.

Notice that when you stopped logstashA, HAProxy forwarded messages from Filebeat to logstashB. Also, when you stopped logstashB, HAProxy forwarded messages to logstashA. This verifies redundancy and failover between the two Logstash receivers.

j. Press Ctrl + C to stop the `tail` of any log files that you are currently watching.

k. Run the following command to start the logstashB server.

```
startlogstashB
```

Unit 3 Kafka exercises

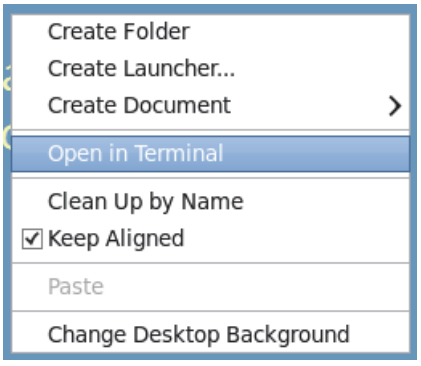
Apache Kafka is a high-throughput distributed messaging system. In this scalable collection architecture, Kafka acts as a durable message store with fault tolerance. Kafka requires Apache Zookeeper for coordination between brokers. In these exercises, you install Kafka and Zookeeper. You then configure the Logstash receiver cluster to send messages to Kafka and verify the data flow. Finally, you install Kafka Manager, which allows you to view your configurations and data that Kafka is processing.

Exercise 1 Installing and configuring Kafka

In this exercise, you install and configure Kafka and Zookeeper.

Important: Run all of the steps in this exercise on the host named **kafka.csite.ibm.edu** as the **netcool** user.

1. The Kafka software is included with Log Analysis. Copy the Kafka software from the Log Analysis server to the kafka host.
   a. Open a terminal window on the host that is named **kafka.csite.ibm.edu**. Right-click the desktop and click **Open In Terminal**.



   b. Run the following command to verify that you are working on the correct host. You should be working on the host named **kafka.csite.ibm.edu**.

```
hostname
```

```
kafka.csite.ibm.edu
```

c. Run the following commands to create a directory for the Kafka installation file and to change into the new directory.

```
mkdir /software/kafka

cd /software/kafka
```

d. Run the following command to copy the Kafka installation file from the Log Analysis server. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**. Enter the entire command on one line.

```
scp
netcool@log-analysis.csite.ibm.edu:/opt/IBM/LogAnalysis/kafka/kafka_2.9.1-0.
8.2.2.tgz .


The authenticity of host '192.168.100.180 (192.168.100.180)' can't be
established.
RSA key fingerprint is a7:09:f9:fc:ec:62:ad:6e:69:2a:d3:7a:2d:e5:d8:a0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.180' (RSA) to the list of known
hosts.
netcool@192.168.100.180's password: object00
```

2. Run the following command to decompress the installation file into the `/opt/` directory. This action installs Kafka and Zookeeper.

```
tar -zxvf kafka_2.9.1-0.8.2.2.tgz -C /opt/
```

3. Configure Zookeeper.

a. Open the `zookeeper.properties` file in a text editor. This example uses `vi`.

```
vi /opt/kafka_2.9.1-0.8.2.2/config/zookeeper.properties
```

b. Find the following line. This is the port that Zookeeper uses to communicate.

```
clientPort=2181
```

c. Change the port number to **17981**.

```
clientPort=17981
```

d. Save and close the file when you are finished.

4. Run the following command to start Zookeeper. Run the entire command on one line.

```
/opt/kafka_2.9.1-0.8.2.2/bin/zookeeper-server-start.sh -daemon
/opt/kafka_2.9.1-0.8.2.2/config/zookeeper.properties &
```

5. Configure Kafka.

a. Open a new terminal window.

b. Open the **server.properties** file in a text editor. This example uses `vi`.

```
vi /opt/kafka_2.9.1-0.8.2.2/config/server.properties
```

c. Find the following line. This is the port that Kafka uses to communicate.

```
port=9092
```

d. Change the port number to **17991**.

```
port=17991
```

e. Find the following line. This is the directory where Kafka writes the commit logs.

```
log.dirs=/tmp/kafka-logs
```

f. Change the directory to `/tmp/kafka-logs-server-0`.

```
log.dirs=/tmp/kafka-logs-server-0
```

g. Find the following line. This is the host and port that Kafka uses to connect to Zookeeper.

```
zookeeper.connect=localhost:2181
```

h. Change the host and port to kafka.csite.ibm.edu:17981. Remember, you configured Zookeeper to use port 17981 in a preceding step.

```
zookeeper.connect=kafka.csite.ibm.edu:17981
```

i. Save and close the file when you are finished.

6. Run the following command to start Kafka. Run the entire command on one line.

```
/opt/kafka_2.9.1-0.8.2.2/bin/kafka-server-start.sh -daemon
/opt/kafka_2.9.1-0.8.2.2/config/server.properties
```

Exercise 2 Configuring Logstash receiver output

In this exercise, you change the configuration of the Logstash receivers to send messages to Kafka.

Important: Run all of the steps in this exercise on the host named **r-logstash.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **r-logstash.csite.ibm.edu**. This is the host where you installed the Logstash receivers.

2. Configure the logstashA server to send messages from the target log file to Kafka.

   a. Open the logstashA configuration file in a text editor.

```
vi /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
```

b. Add the following lines in bold typeface to the output section of your Logstash configuration. Add them under the file plug-in configuration.

```
file {
        path => "/opt/logstashA/logstash-2.2.1/log/logstashA-debug.log"
        codec => rubydebug
    } #end file

   if ("mutate_filebeat" in [tags]) and ! ("_grokparsefailure" in [tags])
{
        kafka {
            bootstrap_servers =>"kafka.csite.ibm.edu:17991"
            topic_id => "%{datasource}"
            message_key => "%{resourceID}"
        } #end Kafka output
    } #end Kafka condition
```

c. Save and close the file when you are finished.

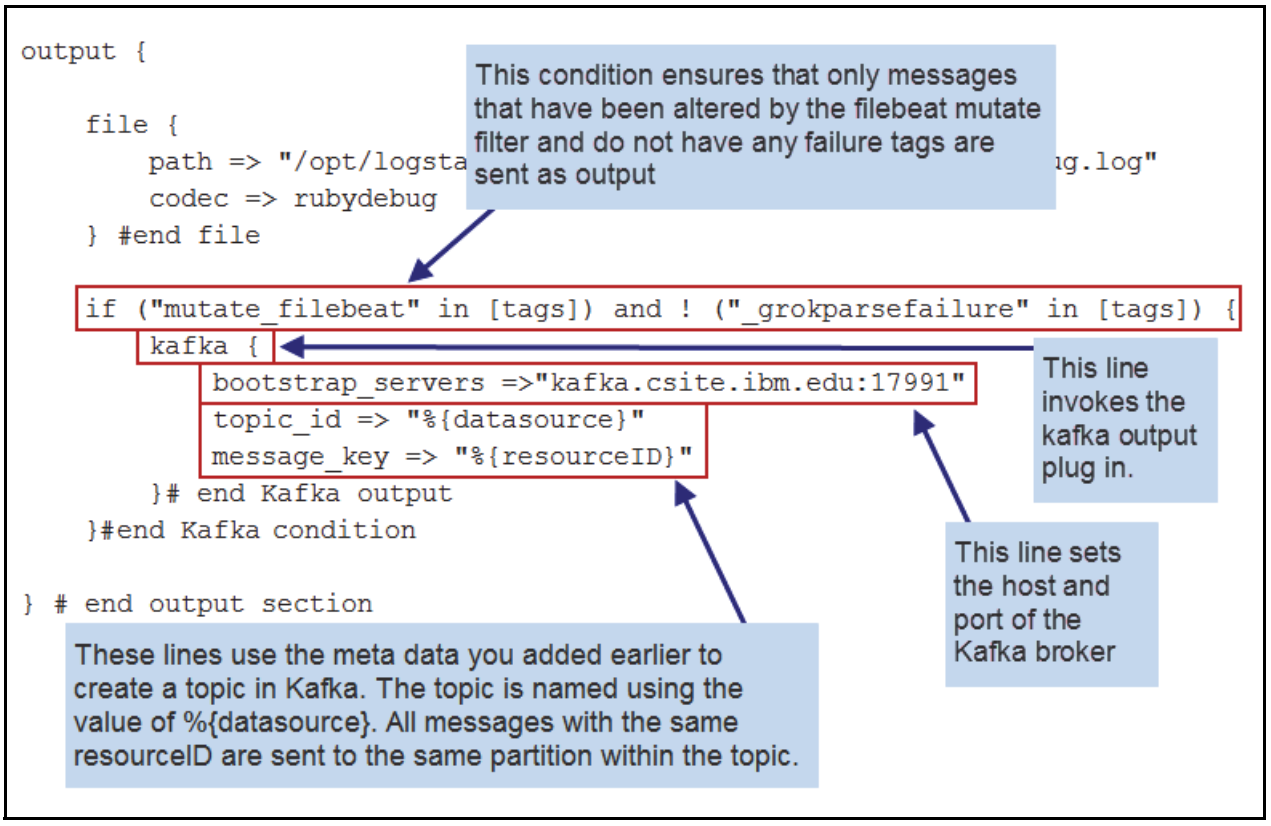


Figure 1 Key fields in the Logstash receiver configuration

3. Configure the logstashB server to send messages from the target log file to Kafka.

   a. Open the logstashB configuration file in a text editor.

```
vi /opt/logstashB/logstash-2.2.1/conf/logstashB.conf
```

b. Add the following lines in bold typeface to the output section of your Logstash configuration. Add them under the file plug-in configuration.

```
file {
        path => "/opt/logstashB/logstash-2.2.1/log/logstashB-debug.log"
        codec => rubydebug
    } #end file

   if ("mutate_filebeat" in [tags]) and ! ("_grokparsefailure" in [tags])
{
        kafka {
            bootstrap_servers =>"kafka.csite.ibm.edu:17991"
            topic_id => "%{datasource}"
            message_key => "%{resourceID}"
        } #end Kafka output
    } #end Kafka condition
```

c. Save and close the file when you are finished.

4. Run the following commands to restart the logstashA server.

```
pkill -f logstashA

startlogstashA
```

5. Run the following commands to restart the logstashB server.

```
pkill -f logstashB

startlogstashB
```

Exercise 3 Verifying data flow

In this exercise, you verify that log messages are flowing from Filebeat through HAProxy, through the Logstash receivers, and into Kafka.

Important: You use two different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Add more messages to the target log file.

a. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

b. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

2. Go to the host named **kafka.csite.ibm.edu**. You installed Kafka and Zookeeper on this host.
3. Verify that the Logstash receiver created a topic for messages from the target file.
   a. Open a new terminal window.
   b. Run the following commands to list all topics.

```
cd /opt/kafka_2.9.1-0.8.2.2/bin/

./kafka-topics.sh --list --zookeeper kafka.csite.ibm.edu:17981

DEV_IBM-HTTP-Server_access-log
```

Note: Notice that the name of the topic is **DEV_IBM-HTTP-Server_access-log**. This name came from the filter and from the kafka output plug-in in your Logstash configuration:

- From your Logstash receiver filter section:

```
add_field => [ "datasource",
"%{[fields][env]}_%{[fields][module]}_%{[fields][type]}" ]
```

- From your Logstash receiver output section:

```
if ("mutate_filebeat" in [tags]) and ! ("_grokparsefailure" in [tags]) {
        kafka {
            bootstrap_servers =>"kafka.csite.ibm.edu:17991"
            topic_id => "%{datasource}"
            message_key => "%{resourceID}"
        } #end Kafka output
    } #end Kafka condition
```

The value for %{datasource} is set by the metadata you added in your Filebeat configuration:

```
fields:
        collector: filebeats-collection.csite.ibm.edu
        env: DEV
        module: IBM-HTTP-Server
        type: access-log
        site: DALLAS
        platform: RHEL
```

4. Verify that the topic contains messages from the target log file.
   a. Run the following command on one line. Notice the log messages from the HTTP server.

```
./kafka-console-consumer.sh --zookeeper kafka.csite.ibm.edu:17981 --topic
DEV_IBM-HTTP-Server_access-log --from-beginning

{"message":"Apache/IHS,192.168.2.94,-,-,[10/May/2016:14:05:31
-0400],GET,\"GET /daytrader/app?action=sell&holdingID=5015
...
```

b. Press Ctrl + C to stop the output of the messages.

Note: You can use this command to watch messages arrive into a Kafka topic, similar to tailing a log file.

Exercise 4 Kafka Manager

Kafka Manager provides a graphical interface that you can use to view Kafka brokers, configurations, and data that Kafka is processing. You use a web browser to access Kafka Manager. In this exercise, you install and use Kafka Manager.

Important: You use two different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **kafka.csite.ibm.edu**. You installed Kafka and Zookeeper on this host.
2. Run the following commands to install Kafka Manager into the **/opt** directory.

   ```
   cd /software/Kafka_Manager/

   unzip kafka-manager-1.3.0.8.zip -d /opt/
   ```

3. To connect to Kafka Manager, Kafka uses Java Management Extensions (JMX). Stop Kafka, and restart it so that it uses JMX port 8092.
   a. Run the following command to stop Kafka.

      ```
      /opt/kafka_2.9.1-0.8.2.2/bin/kafka-server-stop.sh
      ```

   b. Set the JMX port in your environment.

      ```
      export JMX_PORT=8092
      ```

   c. Run the following command to start Kafka. Run the entire command on one line.

      ```
      /opt/kafka_2.9.1-0.8.2.2/bin/kafka-server-start.sh -daemon
      /opt/kafka_2.9.1-0.8.2.2/config/server.properties
      ```

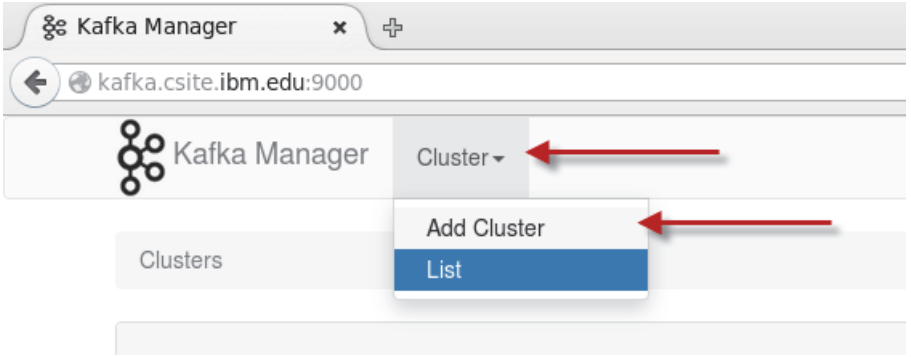
4. Run the following command to start Kafka Manager. Run the entire command on one line. Notice that you specify the Zookeeper host and port number when you start Kafka Manager.

   ```
   /opt/kafka-manager-1.3.0.8/bin/kafka-manager
   -Dkafka-manager.zkhosts="kafka.csite.ibm.edu:17981" &
   ```

5. Open the Kafka Manager page.
   a. Open a Firefox browser.

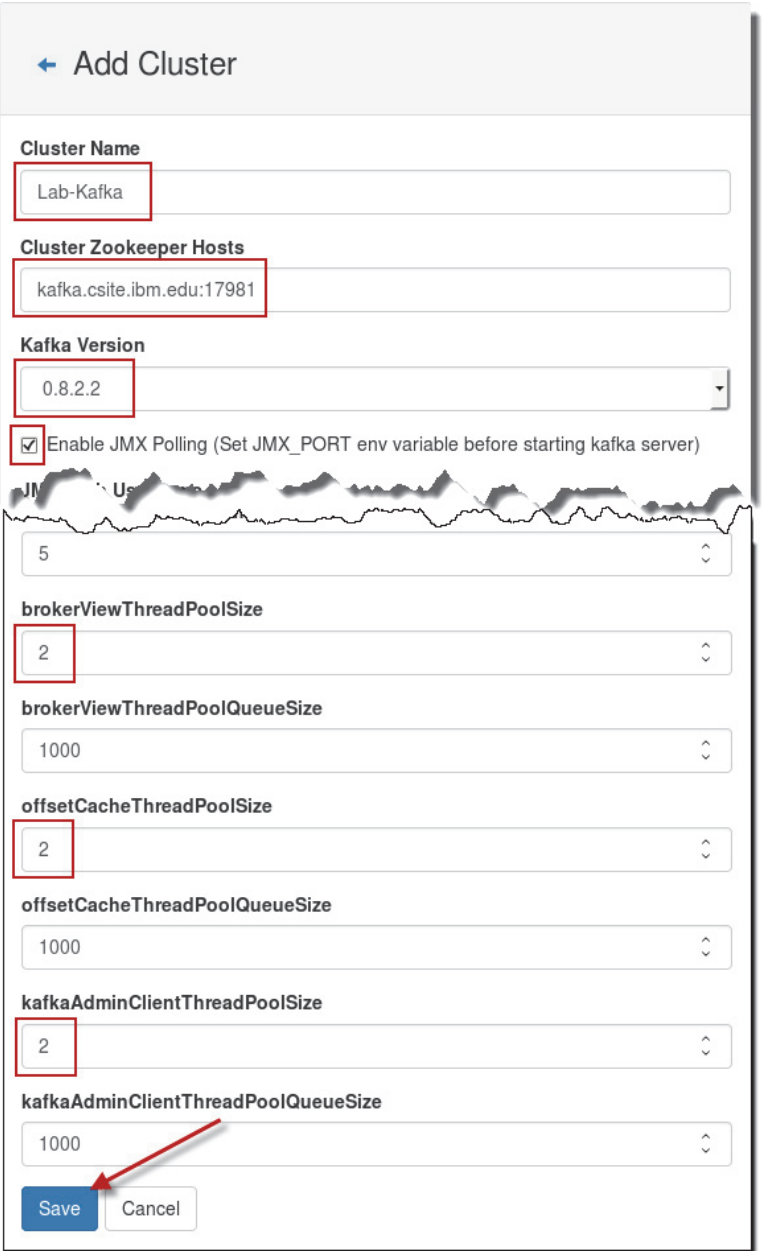
   b. Enter the following address:
      http://kafka.csite.ibm.edu:9000/

6. Add a cluster object and add your Kafka broker to the new cluster.
   a. Click **Cluster > Add Cluster**.



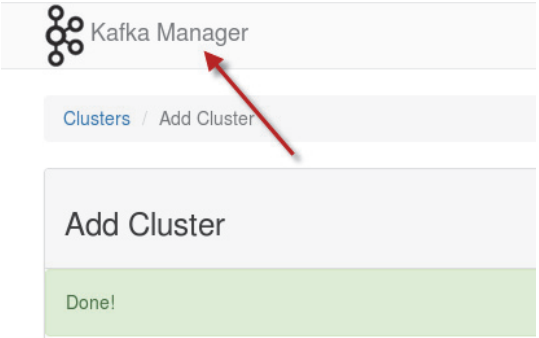
   b. Enter **Lab-Kafka** as the cluster name.
   c. Enter **kafka.csite.ibm.edu:17981** as the Zookeeper host.
   d. Select **0.8.2.2** as the Kafka version.
   e. Select **Enable JMX Polling**.
   f. Enter **2** as the brokerViewThreadPoolSize.
   g. Enter **2** as the offsetCacheThreadPoolSize.
   h. Enter **2** as the kafkaAdminClientThreadPoolSize.

i. Click **Save**.

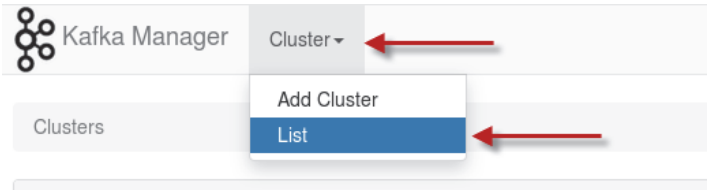


7. Use Kafka Manager to view details about your environment.

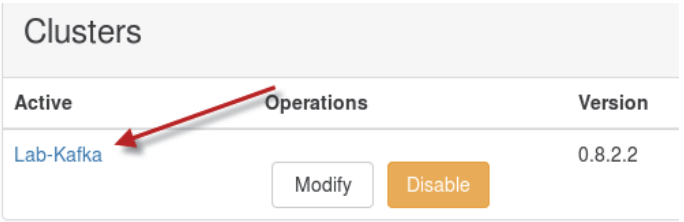
   a. Click **Kafka Manager** at the top of the page.



b. Click **Cluster > List** at the top of the page.

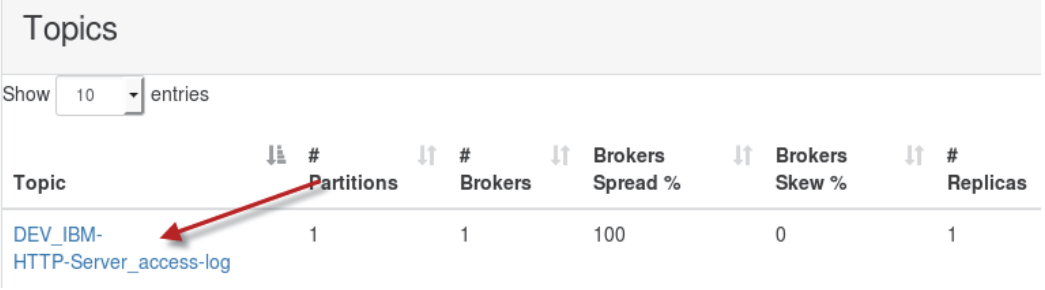


c. Click the **Lab-Kafka** link.

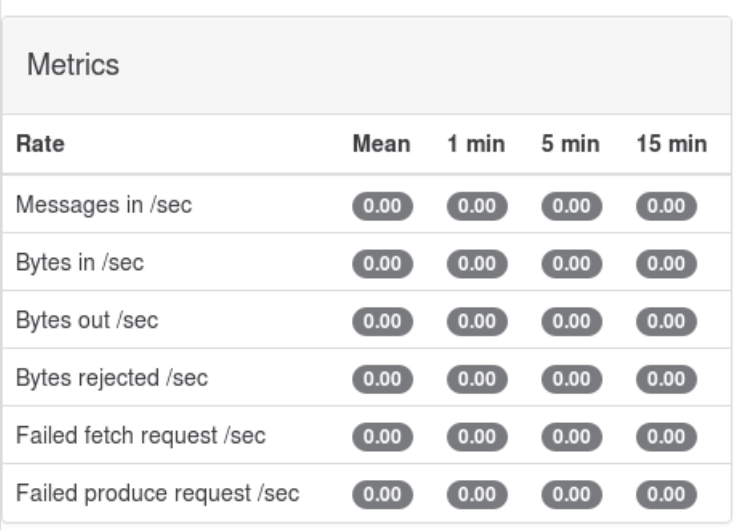


d. Click the number **1** link.

e. This page lists all topics in your environment. You have one topic named **DEV_IBM-HTTP-Server_access-log**. Click the topic name.

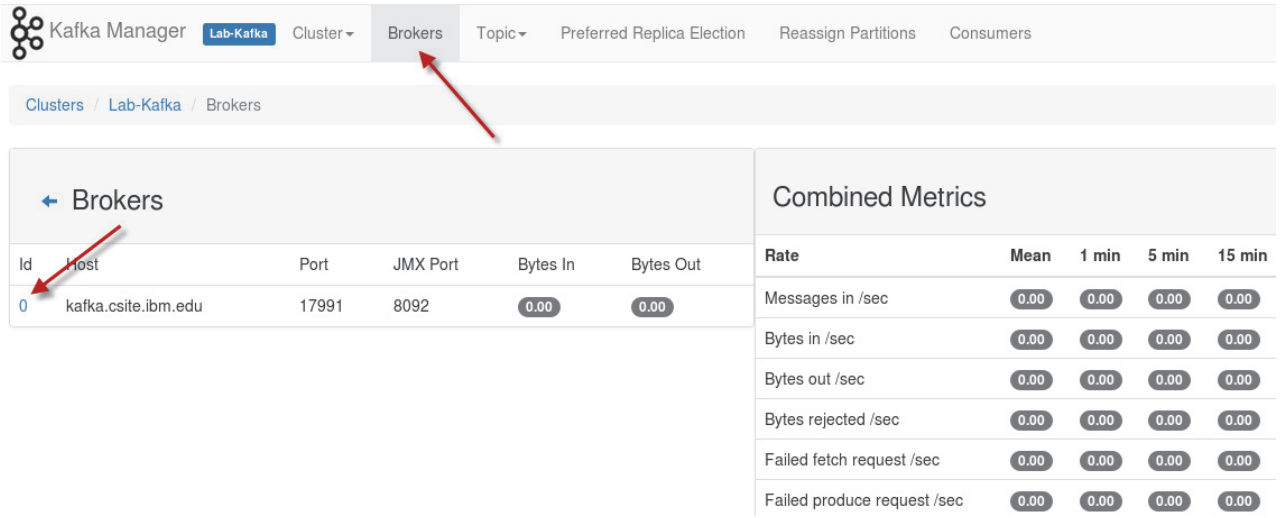


Notice the Metric pane. This pane shows details about the traffic in and out of your topic.

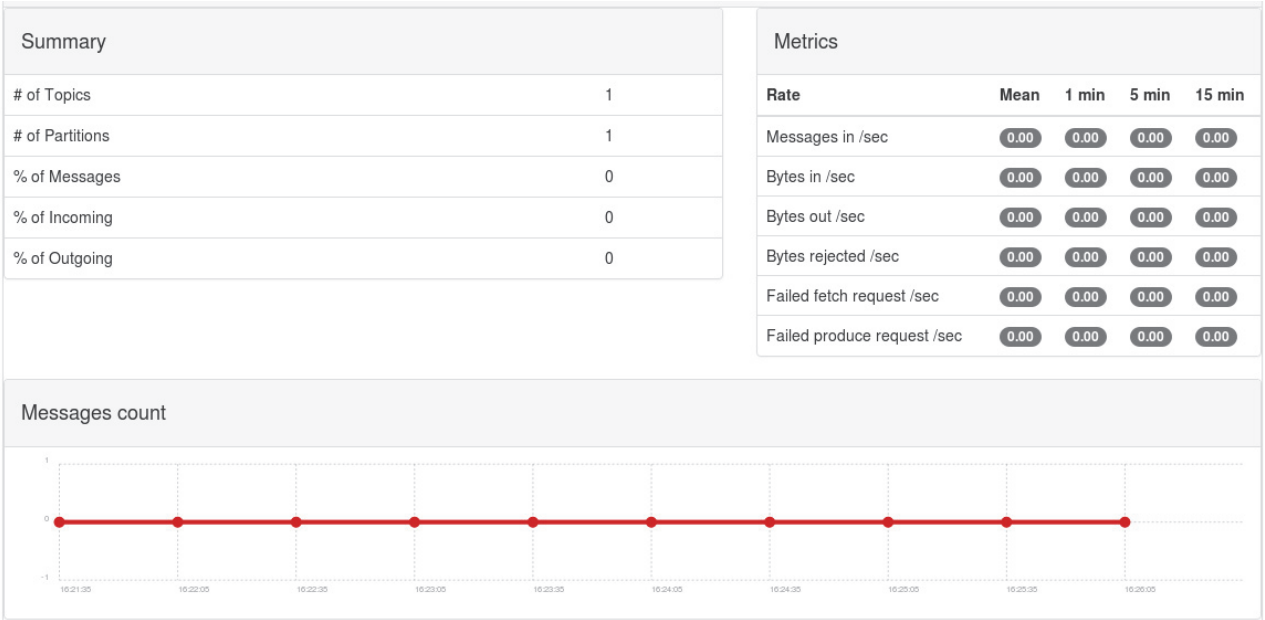


f. Look at some details about your Kafka server. Click **Brokers** at the top of the page.

g. Click your broker ID, which is **0**.



Notice the details about the traffic in and out of your broker.



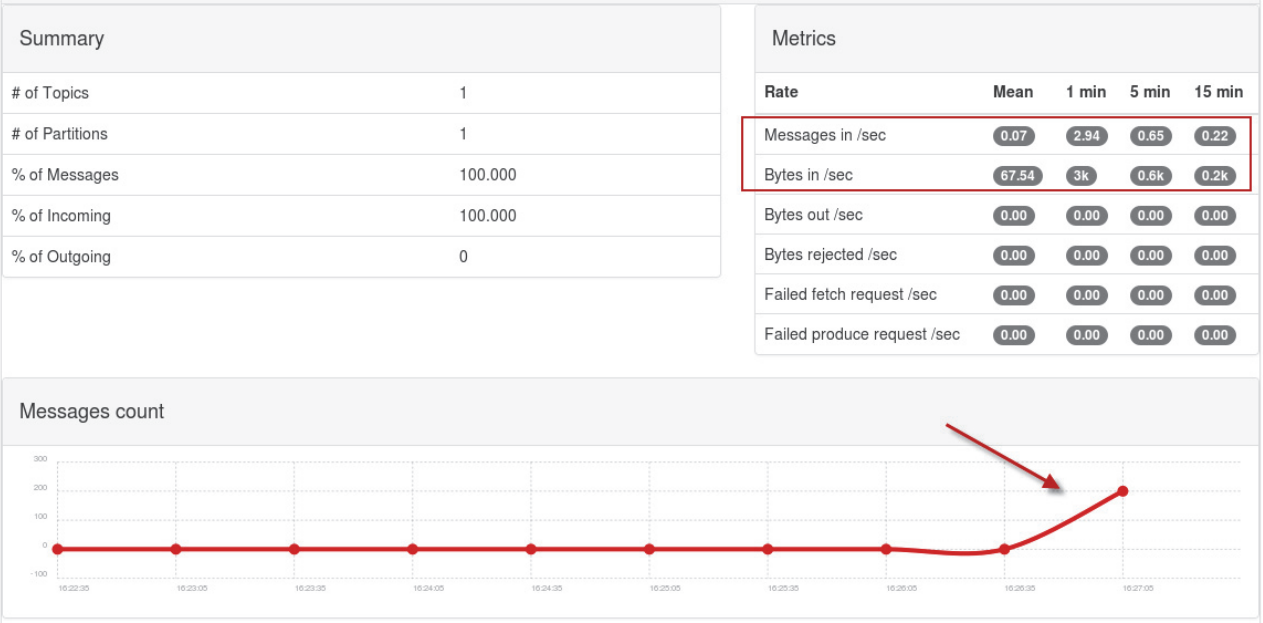
h. Leave this page open. You use it again in a moment.

8. Add more messages to the target log file.
   a. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.
   b. Run the following command to add more messages to the target log file.

      `/software/log_samples/scripts/Dallas_Web_Logs.sh`

9. Return to the host named **kafka.csite.ibm.edu**.

10. Refresh the details page for broker ID 0. Notice that the volume of traffic and messages into the server have increased.



11. Take some time to explore the other Kafka Manager pages.

Unit 4 Logstash senders and Log Analysis core exercises

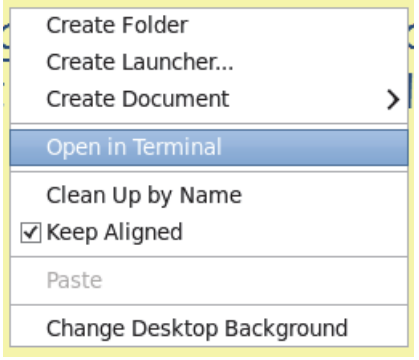
Logstash senders pull messages from Kafka, then send them to the Log Analysis core software. They can also alter the messages before they send them. In these exercises, you configure two Logstash sender instances and verify that they are moving data from Kafka to Log Analysis.

Exercise 1 Installing and configuring the Logstash sender

Logstash senders pull messages from the Kafka brokers, process the log messages, and send them to the Log Analysis core software. In this exercise, you install and configure a Logstash sender server.

Important: You use two different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **s-logstash.csite.ibm.edu**.
2. The Logstash software is included with Log Analysis. Copy the Logstash software from the Log Analysis server to the sender Logstash host.
   a. Open a terminal window on the host that is named **s-logstash.csite.ibm.edu**. Right-click the desktop and click **Open In Terminal**.



b. Run the following command to verify that you are working on the correct host. You should be working on the host named **s-logstash.csite.ibm.edu**.

```
hostname

s-logstash.csite.ibm.edu
```

c. Run the following commands to create a directory for the Logstash installation file and to change into the new directory.

```
mkdir /software/logstash

cd /software/logstash/
```

d. Run the following command to copy the Logstash installation file from the Log Analysis server. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**.

```
scp
netcool@192.168.100.180:/opt/IBM/LogAnalysis/logstash-2.2.1/logstash-2.2.1.t
ar.gz .


The authenticity of host '192.168.100.180 (192.168.100.180)' can't be
established.
RSA key fingerprint is a7:09:f9:fc:ec:62:ad:6e:69:2a:d3:7a:2d:e5:d8:a0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.180' (RSA) to the list of known
hosts.
netcool@192.168.100.180's password: object00
```

3. Run the following command to create a directory for the Logstash sender. In the topology you are building for this lab, the Logstash sender instance is called **logstashW**.

```
mkdir /opt/logstashW
```

4. Run the following command to decompress the installation file and install Logstash into the `/opt/logstashW` directory.

```
tar -zxvf logstash-2.2.1.tar.gz -C /opt/logstashW/
```

5. Create a directory for the logstashW configuration file.

```
mkdir /opt/logstashW/logstash-2.2.1/conf
```

6. Create a directory for the logstashW log file.

```
mkdir /opt/logstashW/logstash-2.2.1/log
```

7. Configure the logstashW instance to pull messages from the Kafka server and send them to a log file. Remember, the Kafka topic that contains the log messages is named **DEV_IBM-HTTP-Server_access-log**.

Note: You configure Logstash by editing a text file.

a. Create and edit a Logstash configuration file named `logstashW.conf` in a text editor. Create the file in the `/opt/logstashW/logstash-2.2.1/conf` directory. This example uses vi.

```
vi /opt/logstashW/logstash-2.2.1/conf/logstashW.conf
```

b. Add the following lines to your `logstashW.conf` file.

```
input {

    kafka {
        zk_connect => "kafka.csite.ibm.edu:17981"
        group_id => "G-DEV_IBM-HTTP-Server_access-log"
        topic_id => "DEV_IBM-HTTP-Server_access-log"
        consumer_threads => 1
        consumer_restart_on_error => true
        consumer_restart_sleep_ms => 100
        decorate_events => true
    }# end HTTP server log Kafka input

}# end input section

filter {

    if "mutate_filebeat" in [tags] {
        mutate {
          add_field => [ "path", "%{[fields][type]}" ]
          replace => { "host" => "%{[fields][env]}_%{[fields][module]}" }
        } # end mutate
    } # end filebeat mutate condition

}# end filter section

output {

    file {
        path => "/opt/logstashW/logstash-2.2.1/log/logstashW-debug.log"
        codec => rubydebug
    }#end file

}# end output section
```

c. Save and close the file when you are done.

```
input {

    kafka {
        zk_connect => "kafka.csite.ibm.edu:17981"
        group_id => "G-DEV_IBM-HTTP-Server_access-log"
        topic_id => "DEV_IBM-HTTP-Server_access-log"
        consumer_threads => 1
        consumer_restart_on_error => true
        consumer_restart_sleep_ms => 100
        decorate_events => true
    }# end HTTP server log Kafka input

}# end input section

filter {

    if "mutate_filebeat" in [tags] {
        mutate {
          add_field => [ "path", "%{[fields][type]}" ]
          replace => { "host" => "%{[fields][env]}_%{[fields][module]}" }
        } # end mutate
    } # end filebeat mutate condition

}# end filter section

output {

    file {
        path => "/opt/logstashW/logstash-2.2.1/log/logstashW-debug.log"
        codec => rubydebug
    }#end file

}# end output section
```

This line invokes the kafka input plug-in

This is the host and port number of Zookeeper

This line sets the consumer group in Kafka

This is the Kafka topic you want to read from

The number of threads to use to read from Kafka

These options restart the Kafka consumer after 100 milliseconds if an error is encountered

This option adds meta data about Kafka to the mesages, such as topic ID

This filter sets two fields: **path** and **host**. These fields are used when you configure the data source in Log Analysis.

These line output the result of logstash procesing to a log file

Figure 1 Key fields in logstashW.conf

8. In these exercises, you start and stop the logstashW server many times as you build your configuration. Add a command alias to make it easier to start Logstash.
   a. Open the `.bashrc` file of the **netcool** user in a text editor.

      `vi /home/netcool/.bashrc`

b. Add the following line to the bottom of the file.

```
alias startlogstashW='/opt/logstashW/logstash-2.2.1/bin/logstash -f
/opt/logstashW/logstash-2.2.1/conf/logstashW.conf -l
/opt/logstashW/logstash-2.2.1/log/logstashW-debug.log &'
```

c. Save and close the file when you are finished.

d. Run the following command to source the modified environment file.

```
source /home/netcool/.bashrc
```

e. Run the following command to start the logstashW server.

```
startlogstashW
```

9. Run the following command to verify that the logstashW server is running.

```
ps -ef | grep logstashW

netcool   5090  2496 34 19:54 pts/0    00:00:29 /usr/bin/java -XX:+UseParNewGC
-XX:+UseConcMarkSweepGC -Djava.awt.headless=true
-XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly
-XX:+HeapDumpOnOutOfMemoryError -Xmx1g -Xss2048k
...
```

10. Run the following command to watch for activity in the logstashW log file. Leave the `tail` command running.

```
tail -f /opt/logstashW/logstash-2.2.1/log/logstashW-debug.log
```

11. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

12. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Dallas_Web_Logs.sh
```

13. Return to the host named **s-logstash.csite.ibm.edu**. Look at the `logstashW-debug.log` file. Look for messages like the following example. Messages like these verify that Logstash is pulling log messages from the Kafka server.

    a. Notice the metadata about Kafka at the bottom of each message.

b. Notice the host and path that were set by the mutate filter.

```
"message" => "Apache/IHS,192.168.2.94,-,-,[10/May/2016:15:41:37
-0400],GET,\"GET /daytrader/app?action=buy&symbol=s%3A0&quantity=100
HTTP/1.1\",200,3409,1184,\"-\",\"curl/7.21.3 (x86_64-unknown-linux-gnu)
libcurl/7.21.3 OpenSSL/1.0.0 zlib/1.2.3\"",
      "@version" => "1",
    "@timestamp" => "2016-05-31T20:22:01.383Z",
          "beat" => {
        "hostname" => "collection.csite.ibm.edu",
            "name" => "collection.csite.ibm.edu"
    },
         "count" => 1,
        "fields" => {
        "collector" => "filebeats-collection.csite.ibm.edu",
              "env" => "DEV",
           "module" => "IBM-HTTP-Server",
         "platform" => "RHEL",
             "site" => "DALLAS",
             "type" => "access-log"
    },
    "input_type" => "log",
        "offset" => 1726906,
        "source" => "/software/log_samples/IHS_logs/Dallas-IHS-access.log",
          "type" => "log",
          "host" => "DEV_IBM-HTTP-Server",
          "tags" => [
        [0] "beats_input_codec_plain_applied",
        [1] "mutate_filebeat"
    ],
    "datasource" => "DEV_IBM-HTTP-Server_access-log",
    "resourceID" =>
"collection.csite.ibm.edu_/software/log_samples/IHS_logs/Dallas-IHS-access.l
og_1",
         "kafka" => {
              "msg_size" => 880,
                 "topic" => "DEV_IBM-HTTP-Server_access-log",
        "consumer_group" => "G-DEV_IBM-HTTP-Server_access-log",
             "partition" => 0,
                   "key" => byte[99, 111, 108, 108, 101, 99, 116, 105, 111,
110, 46, 99, 115, 105, 116, 101, 46, 105, 98, 109, 46, 101, 100, 117, 95, 47,
115, 111, 102, 116, 119, 97, 114, 101, 47, 108, 111, 103, 95, 115, 97, 109,
112, 108, 101, 115, 47, 73, 72, 83, 95, 108, 111, 103, 115, 47, 73, 72, 83,
45, 97, 99, 99, 101, 115, 115, 46, 108, 111, 103, 95, 49]@1d4b7b63
    },
          "path" => "access-log"
```

14. Press Ctrl + C to stop the tail command.

Exercise 2 Sending data to Log Analysis

Logstash senders use a custom output plug-in named scala to send messages to Log Analysis. In this exercise, you configure Logstash to use this custom output plug-in.

Important: You use three different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Copy the custom output plug-ins for Log Analysis to the Logstash working directory.
   a. Go to the host named **s-logstash.csite.ibm.edu**.
   b. Run the following commands to copy the custom output plug-in to the Logstash working directory.

```
cd /opt/logstashW/logstash-2.2.1/vendor/bundle/jruby/1.9/gems/
logstash-core-2.2.1-java/lib/logstash/outputs/

cp /software/scala_plugin/* .
```

2. Configure the logstashW server to use the scala output plug-in.
   a. Open the `logstashW.conf` file in a text editor.

```
vi /opt/logstashW/logstash-2.2.1/conf/logstashW.conf
```

   b. Add the following lines to the output section of your `logstashW.conf` file.

```
    scala {
        scala_url =>
"https://log-analysis.csite.ibm.edu:9987/Unity/DataCollector"
        scala_user => "unityadmin"
        scala_password => "object00"
        scala_keystore_path => ""
        batch_size => 500000
        idle_flush_time => 5
        sequential_flush => true
        num_concurrent_writers => 20
        use_structured_api => false
        disk_cache_path => "/opt/logstashW/training/cache/basecache"
        date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
        log_file => "/opt/logstashW/logstash-2.2.1/log/scala_logstashW.log"
        log_level => "info"
    }#end scala output
```

c. Save and close the file when you are finished.

```
output {


        file {
            path => "/opt                    sh-2.2.1/log/logstashW-debug.log"
            codec => ruby
        }#end file

        scala {
            scala_url =>
            "https://log-analysis.csite.ibm.edu:9987/Unity/DataCollector"
            scala_user => "unityadmin"
            scala_password => "object00"
            scala_keystore_path => ""
            batch_size => 500000
            idle_flush_time => 5
            sequential_flush => true
            num_concurrent_writers => 20
            use_structured_api => false
            disk_cache_path => "/opt/logstashW/training/cache/basecache"
            date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
            log_file => "/opt/logstashW/logstash-2.2.1/log/scala_logstashW.log"
            log_level => "info"
        }#end scala output

}# end output section
```

scala is a custom output plug-in that is included with Log Analysis

These are details about where and how to send the data. Local cache file details are also set here.

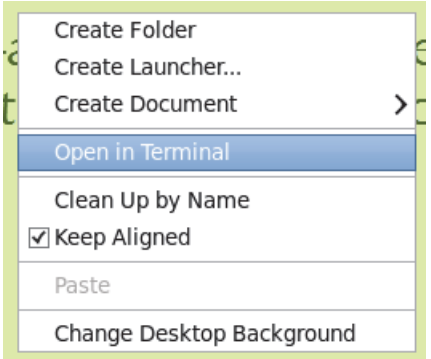
These are details about the log file of the scala plug-in

Figure 2 Key fields in logstashW.conf

3. Run the following command to create the base cache directory for logstashW.

```
mkdir -p /opt/logstashW/training/cache/basecache
```

4. Import the Java security certificate from the Log Analysis server and add it to the Logstash sender server.
   a. Go to the host named **log-analysis.csite.ibm.edu**.
   b. Open a terminal window on the host that is named **log-analysis.csite.ibm.edu**. Right-click the desktop and click **Open In Terminal**.



   c. Run the following command to verify that you are working on the correct host. You should be working on the host named **log-analysis.csite.ibm.edu**.

```
hostname

log-analysis.csite.ibm.edu
```

   d. Run the following command to add the Log Analysis security certificate to the Java runtime environment keystore. Run the entire command on one line. Enter **changeit** as the password and **yes** when you are prompted to trust the certificate.

```
sudo /usr/lib/jvm/jre-1.7.0-openjdk.x86_64/bin/keytool -import -alias scala
-keystore /usr/lib/jvm/jre-1.7.0-openjdk.x86_64/lib/security/cacerts -file
/opt/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security/client.crt

Enter keystore password: changeit

Trust this certificate? [no]:  yes
```

   e. Go to the host named **s-logstash.csite.ibm.edu**. You installed a Logstash sender instance on this host.
   f. Run the following command to change to the home directory of the **netcool** user.

```
cd
```

g. Run the following command to copy the `client.crt` file from the Log Analysis server. Run the entire command on one line. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**.

```
scp
netcool@192.168.100.180:/opt/IBM/LogAnalysis/wlp/usr/servers/Unity/resources
/security/client.crt .

The authenticity of host '192.168.100.180 (192.168.100.180)' can't be
established.
RSA key fingerprint is a7:09:f9:fc:ec:62:ad:6e:69:2a:d3:7a:2d:e5:d8:a0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.180' (RSA) to the list of known
hosts.
netcool@192.168.100.180's password: object00
```

h. Run the following command to import the certificate. Run the entire command on one line. Enter **yes** when you are prompted to trust the certificate.

```
sudo /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.91.x86_64/jre/bin/keytool -import
-file /home/netcool/client.crt -keystore
/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.91.x86_64/jre/lib/security/cacerts
-storepass changeit

Trust this certificate? [no]:  yes
```

5. Restart the logstashW server.

   a. Run the following command to stop the logstashW server.

   ```
   pkill -f logstashW
   ```

   b. Run the following command to start the logstashW server.

   ```
   startlogstashW
   ```

6. In two different terminal windows, watch for activity in the `logstashW-debug.log` and `scala_logstashW.log` files.

   a. Run the following command to watch for activity in the `logstashW-debug.log` file.

   ```
   tail -f /opt/logstashW/logstash-2.2.1/log/logstashW-debug.log
   ```

   b. Run the following command in a different terminal window to watch for activity in the `scala_logstashW.log` file.

   ```
   tail -f /opt/logstashW/logstash-2.2.1/log/scala_logstashW.log
   ```

Note: You can ignore any warning messages about the scala plug-in.

7. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

8. Run the following command to add more messages to the target log file.

   ```
   /software/log_samples/scripts/Dallas_Web_Logs.sh
   ```

9. Return to the host named **s-logstash.csite.ibm.edu**.

10. Look at the `logstashW-debug.log` file. Look for messages like the following example. Messages like these verify that the logstashW server is still receiving messages from the target log file.

    ```
    "message" => "Apache/IHS,192.168.2.94,-,-,[10/May/2016:15:41:37
    -0400],GET,\"GET /daytrader/app?action=buy&symbol=s%3A0&quantity=100
    HTTP/1.1\",200,3409,1184,\"-\",\"curl/7.21.3 (x86_64-unknown-linux-gnu)
    libcurl/7.21.3 OpenSSL/1.0.0 zlib/1.2.3\"",
          "@version" => "1",
        "@timestamp" => "2016-05-24T14:30:45.680Z",
              "beat" => {
            "hostname" => "collection.csite.ibm.edu",
                "name" => "collection.csite.ibm.edu"
    ...
    ```

11. Look at the `scala_logstashW.log` file. Look for messages like the following example. Messages like these verify that the logstashW server is sending messages to the Log Analysis server, but Log Analysis does not have a corresponding data source configured.

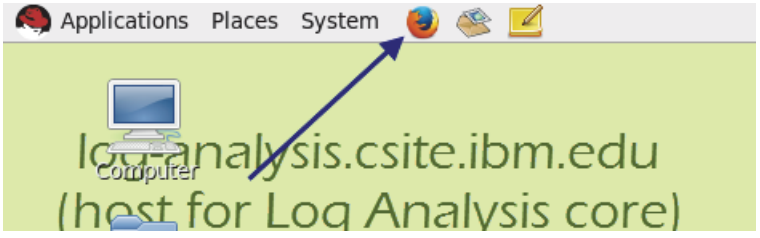
    ```
    ERROR ScalaCollector$CollectorRunnable - Error occurred while processing batch
    {"RESPONSE_CODE":404,"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":"CTGLA0401E :
    Missing data source"}
    ```

Note: In the next steps, you configure a data source in Log Analysis.

12. Press Ctrl + C to stop the tail of any log files that you are currently watching.

13. Add a data source for the target log file in Log Analysis.
    a. Go to the host named **log-analysis.csite.ibm.edu**.

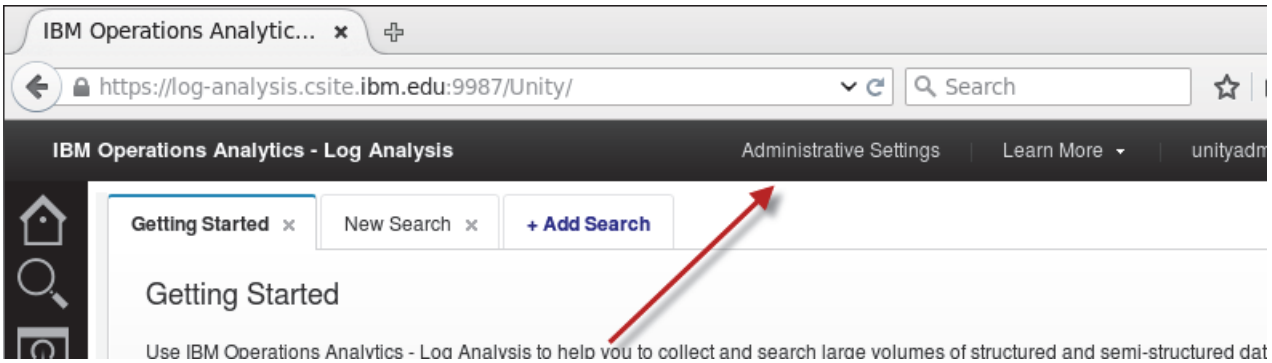
    b. Open a Firefox browser.



    c. Enter the following address:
    https://log-analysis.csite.ibm.edu:9987/Unity

d. Log in to the user interface with the user name **unityadmin** and the password **object00**.

e. Click **Administrative Settings**. The administration user interface opens in a new Firefox tab.

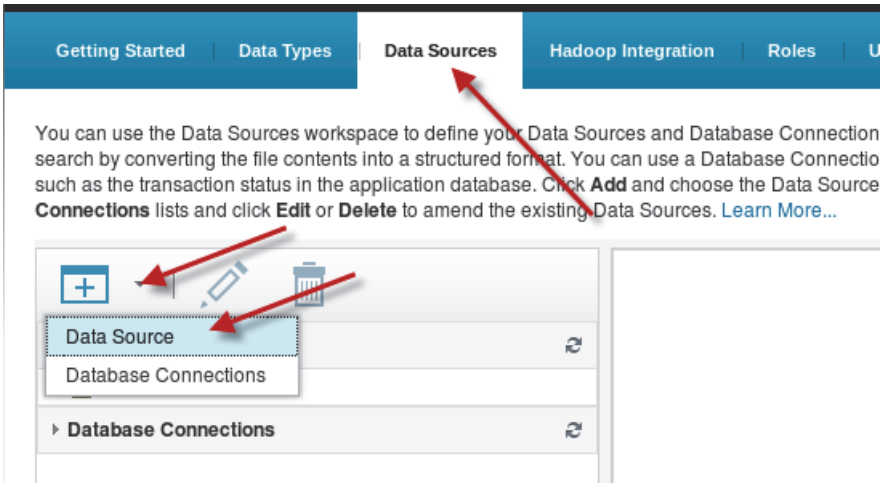


f. Create a data source named **Dallas_Web_Server**. Use the values in the following table to complete the data source wizard.

Field	Value
Location	Select custom
Host name	DEV_IBM-HTTP-Server
File Path	access-log
Type	WebAccessLog
Collection	Leave this field blank
Name	Dallas_Web_Server
Description	Leave this field blank
Group	Leave this field blank

g. Click the **Data Sources** tab in the administration user interface. The administration user interface is in the second Firefox tab.

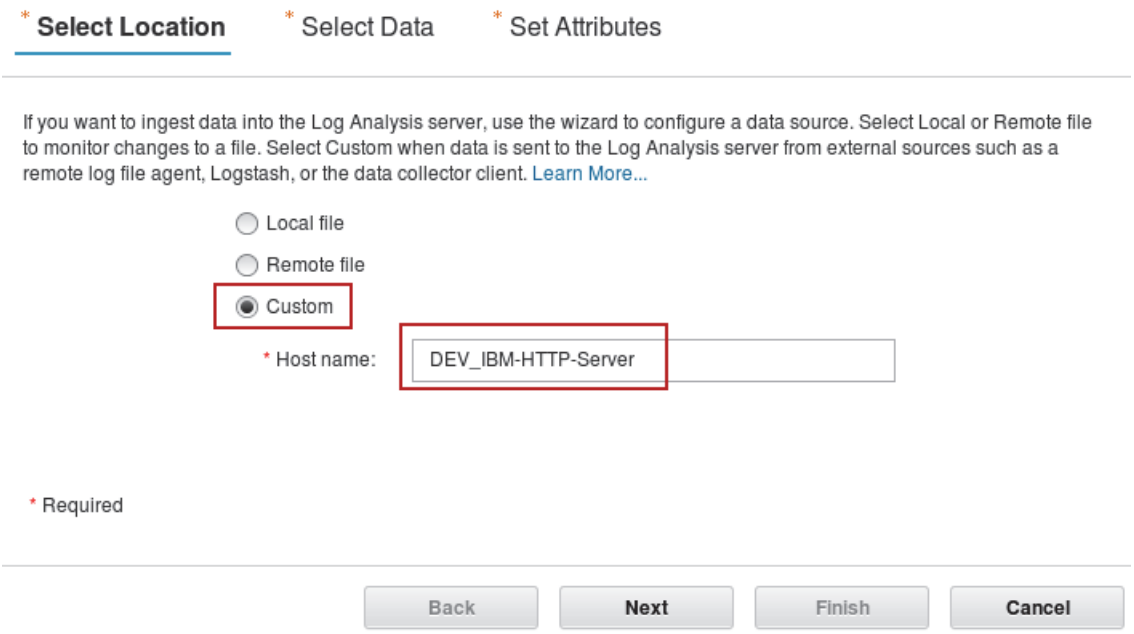
h. Click **Add > Data Source**.



i. Select **Custom**.

j. Enter **DEV_IBM-HTTP-Server** as the host name.

k. Click **Next**.



Note: Notice that the host name is **DEV_IBM-HTTP-Server**. This value corresponds to the value that you set with your mutate filter in the Logstash sender configuration and the metadata that you added with Filebeat.

- From your sender Logstash configuration:

```
replace => { "host" => "%{[fields][env]}_%{[fields][module]}" }
```

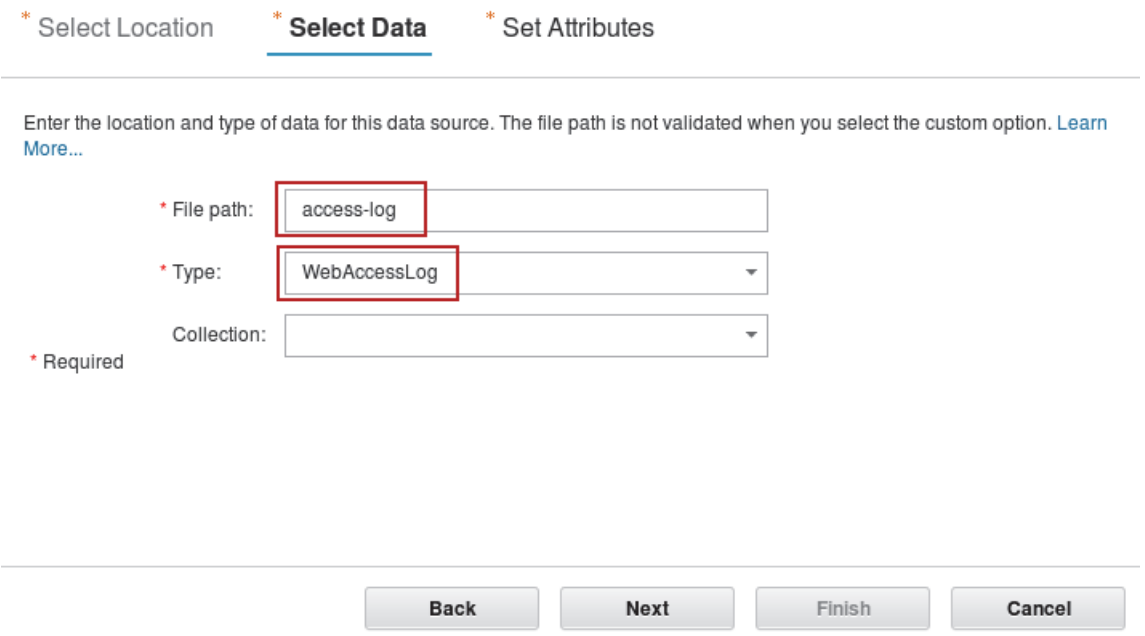
- From your Filebeat configuration:

```
fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: DALLAS
    platform: RHEL
```

l. Enter **access-log** as the file path.

m. Select **WebAccessLog** as the type.

n. Click **Next**.



Note: Notice that the file path is **access-log**. This value corresponds to the value you set with your mutate filter in the Logstash sender configuration and the metadata you added with Filebeat.

- From your sender Logstash configuration:

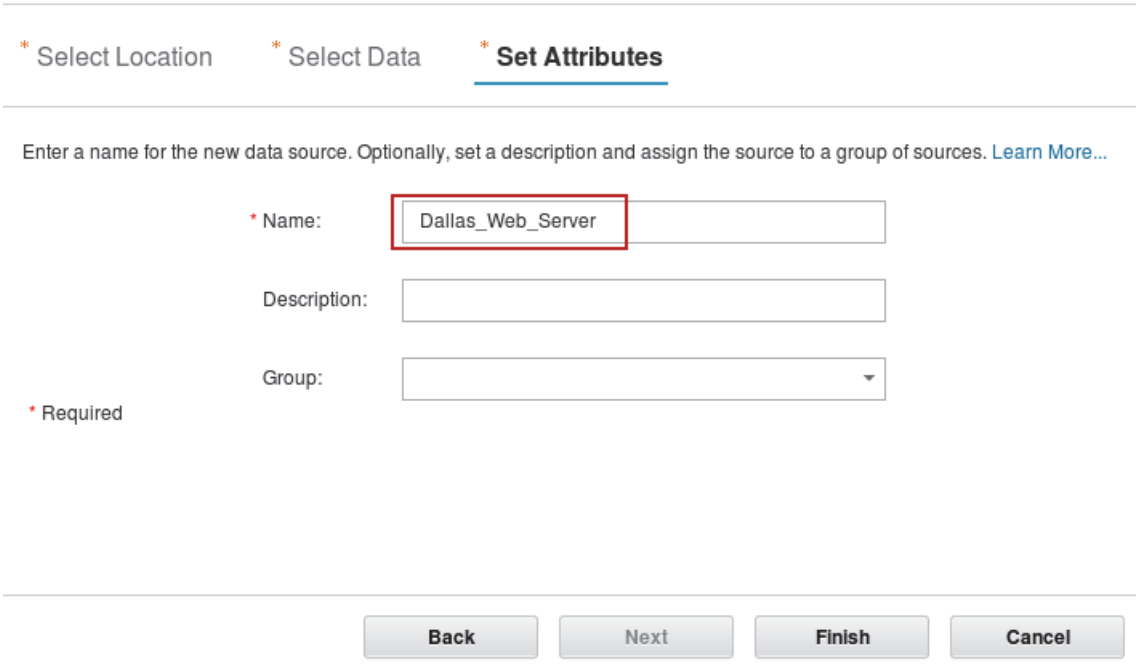
```
add_field => [ "path", "%{[fields][type]}" ]
```

- From your Filebeat configuration:

```
fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: DALLAS
    platform: RHEL
```

o. Enter **Dallas_Web_Server** as the name of the data source.

p. Click **Finish**.



q. Click **OK** in the confirmation windows.

r. Leave this Firefox page open. You use it again in a moment.

The `GenericReceiver.log` file shows all data coming in to the Log Analysis server.

14. Run the following command to watch for activity in the `GenericReceiver.log` file.

```
tail -f /opt/IBM/LogAnalysis/logs/GenericReceiver.log
```

15. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.

16. Run the following command to add more messages to the target log file.

    ```
    /software/log_samples/scripts/Dallas_Web_Logs.sh
    ```

17. Return to the host named **log-analysis.csite.ibm.edu**.

18. Look at the `GenericReceiver.log` file.

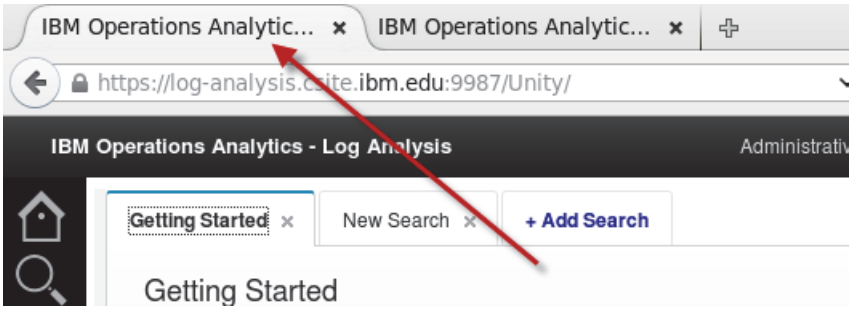
    a. Look for messages like the following example. Messages like these verify that data from the target log file is being processed by the Log Analysis software.

    ```
    05/24/16 16:17:08:986 UTC [Default Executor-thread-61963] INFO  -
    UnityFlowController :  Batch Status for -> Dallas_Web_Server , Size: 79 , Num
    successful: 79 , Num failures: 0 , Indexed Source volume: 0

    05/24/16 16:17:08:986 UTC [Default Executor-thread-61963] INFO  -
    DataCollectorRestServlet : Batch of Size 79 processed and encountered 0
    failures
    ```

    b. Press Ctrl + C to stop the `tail` of the `GenericReceiver.log` file.

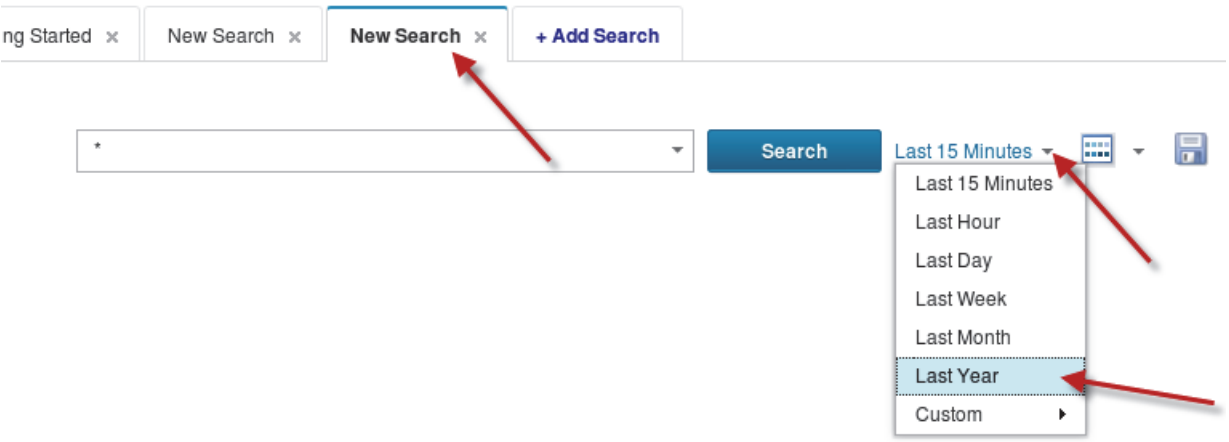
19. Verify that messages from the target log file are present in the Log Analysis search interface.

    a. Return to the Log Analysis user interface in the Firefox window. Go to the search interface by clicking the first Firefox tab.



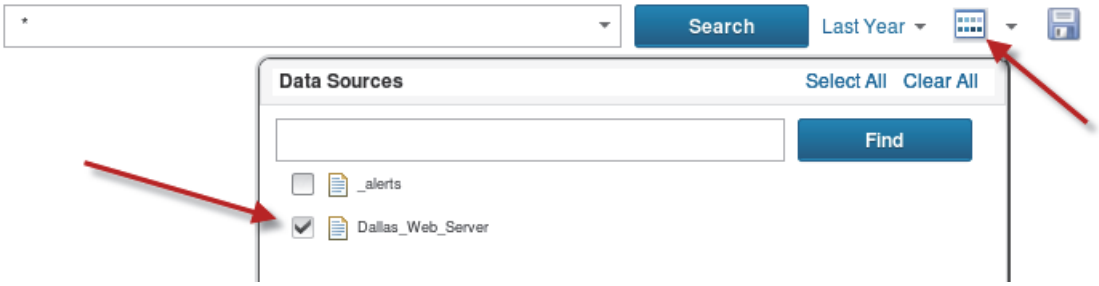
    b. Click the **Add Search** or the **New Search** tab.

c. Select **Last Year** as the time filter.



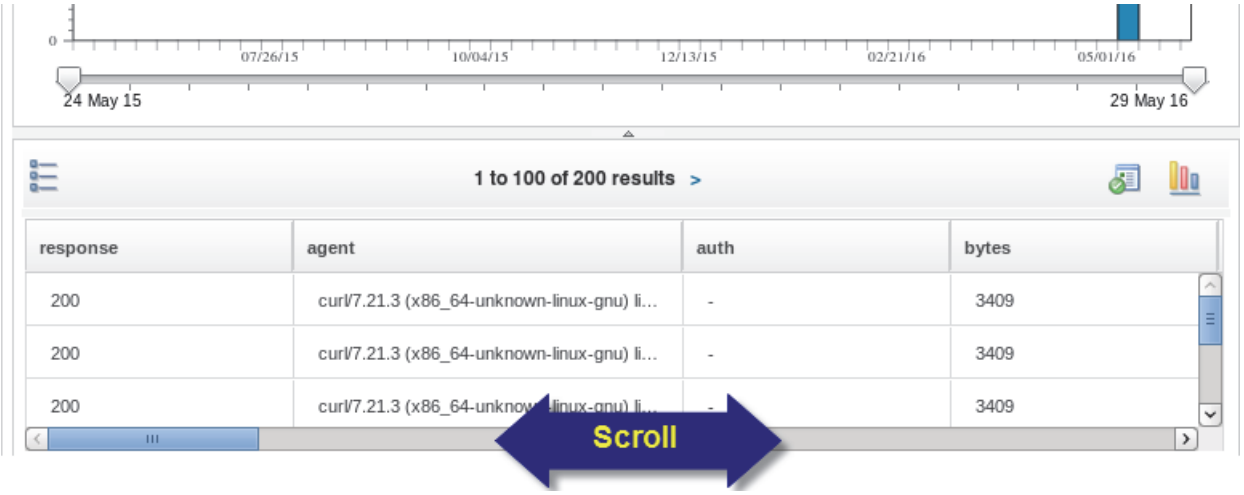
d. Select **Dallas_Web_Server** as the only data source.

e. Click **Search**.

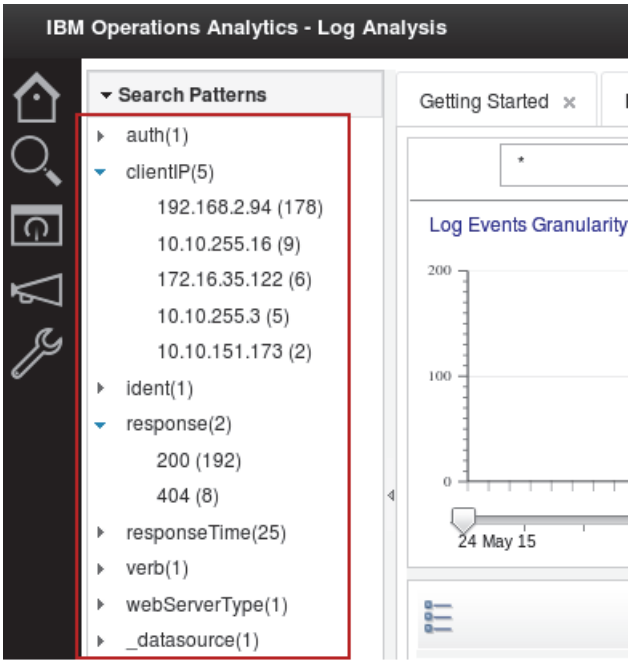


f. Log messages load in to the search interface. Click the **Grid View** button.

g. Scroll left and right to view the columns.



Look at the Search Patterns at the left of the search interface. Notice the facet counts and categories from the log file.



Exercise 3 Verifying Logstash sender resiliency

In this exercise, you stop the Logstash sender server named logstashW and verify that no messages from the target log file are lost.

Important: You use three different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **s-logstash.csite.ibm.edu**.
2. Run the following command to stop the logstashW server.

   ```
   pkill -f logstashW
   ```

3. Go to the host named **log-analysis.csite.ibm.edu**.
4. Run the following command to watch for activity in the `GenericReceiver.log` file. Leave the tail running.

   ```
   tail -f /opt/IBM/LogAnalysis/logs/GenericReceiver.log
   ```

5. Go to the host named **collection.csite.ibm.edu**. You installed Filebeat on this host.
6. Run the following command to add more messages to the target log file.

   ```
   /software/log_samples/scripts/Dallas_Web_Logs.sh
   ```

7. Go to the host named **log-analysis.csite.ibm.edu**. Look at the tail of the `GenericReceiver.log` file. No new messages have arrived. New data cannot arrive because the Logstash sender is stopped.
8. Start the Logstash sender.
   a. Go to the host named **s-logstash.csite.ibm.edu**.
   b. Run the following command to start the logstashW server.

      ```
      startlogstashW
      ```

9. Go to the host named **log-analysis.csite.ibm.edu**. Look at the `tail` of the `GenericReceiver.log` file again. Notice that new messages have arrived.

   ```
   05/24/16 17:25:55:032 UTC [Default Executor-thread-62261] INFO  -
   UnityFlowController :  Batch Status for -> Dallas_Web_Server , Size: 111 , Num
   successful: 111 , Num failures: 0 , Indexed Source volume: 46679

   05/24/16 17:25:55:033 UTC [Default Executor-thread-62261] INFO  -
   DataCollectorRestServlet : Batch of Size 111 processed and encountered 0
   failures
   ```

Important: This verifies that after recovering from a failure, a Logstash sender server reads any new messages in the Kafka queue. The Kafka server keeps messages in its queue until a consumer such as your Logstash sender retrieves them.

10. Press Ctrl + C to stop the `tail` of the `GenericReceiver.log` file.

Unit 5 Using the Log File Agent exercises

You can use the Log File Agent (LFA) as a collection agent to capture messages from a target log file. Earlier in this course, you installed and configured Filebeat as a collection agent. In these exercises, you install and configure a second type of log collector: the Log File Agent. You then update the other components in your lab environment to process messages from the Log File Agent.

Exercise 1 Installing and configuring the Log File Agent

In this exercise, you install and configure the Log File Agent.

Important: You use two different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **log-analysis.csite.ibm.edu**.
2. Use the remote installation tool to install the Log File Agent (LFA) on the host named collection.csite.ibm.edu.
   a. Change to the remote installation tool directory.
      `cd /opt/IBM/LogAnalysis/remote_install_tool`
   b. Open the `ssh-config.properties` file in a text editor.
      `vi config/ssh-config.properties`

c. Change the following values in bold typeface. These values are the details about how to connect to the collection host.

```
REMOTE_HOST=collection.csite.ibm.edu
PORT=22
TIME_OUT=60000
USER=netcool
#PASSWORD can be commented while using Public key based authentication
PASSWORD=object00
```

d. Save and close the file when you are finished.

e. Run the following command to start the remote installation tool.

```
./install.sh
```

f. Enter **/opt/LFA** as the installation directory.

```
Enter Remote Top Level Installation Directory absolute path:
[/home/netcool/LogAnalysis]

/opt/LFA
```

g. Enter **n** when you are prompted to install EIF Receiver instances.

```
Install EIF Receiver Instances (y|Y|n|N):[y]

n
```

h. Enter **y** when you are prompted to install LFA 6.3.

```
Install LFA 6.3 (y|Y|n|N):[y]

y
```

i. Enter **n** when you are prompted to install Logstash. The installation starts.

```
Install logstash 2.2.1 (y|Y|n|N):[y]

n
```

The installation takes several minutes to complete. The following message confirms that the installation was successful.

```
Response:==================================================
Response:    COMPONENT PIDSTATUS
Response:==================================================
Response:  Log File Agent 9784  UP
Response:==================================================
End Time:Thu May 26 16:06:54 UTC 2016
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Total install duration (seconds):109

+++++++++++++++++++++ Installation Ends +++++++++++++++++++++
```

3. Go to the host named **collection.csite.ibm.edu**.

4. Run the following command to verify that the LFA is running.

```
ps -ef | grep -i LFA

netcool   9784     1  0 16:06 ?        00:00:00
/opt/LFA/IBM-LFA-6.30/lx8266/lo/bin/kloagent
collection_default_workload_instance
```

5. You configure the LFA to monitor a log file by creating two files: a .conf file and a .fmt file. Create these two files and configure them to monitor a syslog log file.

   a. Run the following command to create a file named `lab-syslog.conf`.

   ```
   vi /opt/LFA/IBM-LFA-6.30/config/lo/lab-syslog.conf
   ```

   b. Add the following lines to the `lab-syslog.conf` file.

   ```
   LogSources=/software/log_samples/messages.log
   BufEvtPath=/opt/LFA/IBM-LFA-6.30/logs/lab-syslog.cache
   FileComparisonMode=CompareByAllMatches
   ServerLocation=ha-proxy.csite.ibm.edu
   ServerPort=5530
   FQDomain=yes
   BufferEvents=YES
   BufEvtMaxSize=102400
   EventMaxSize=32768
   ConnectionMode=CO
   PollInterval=3
   NumEventsToCatchUp=-1
   ServerSSL=NO
   ```

c. Save and close the file when you are finished.

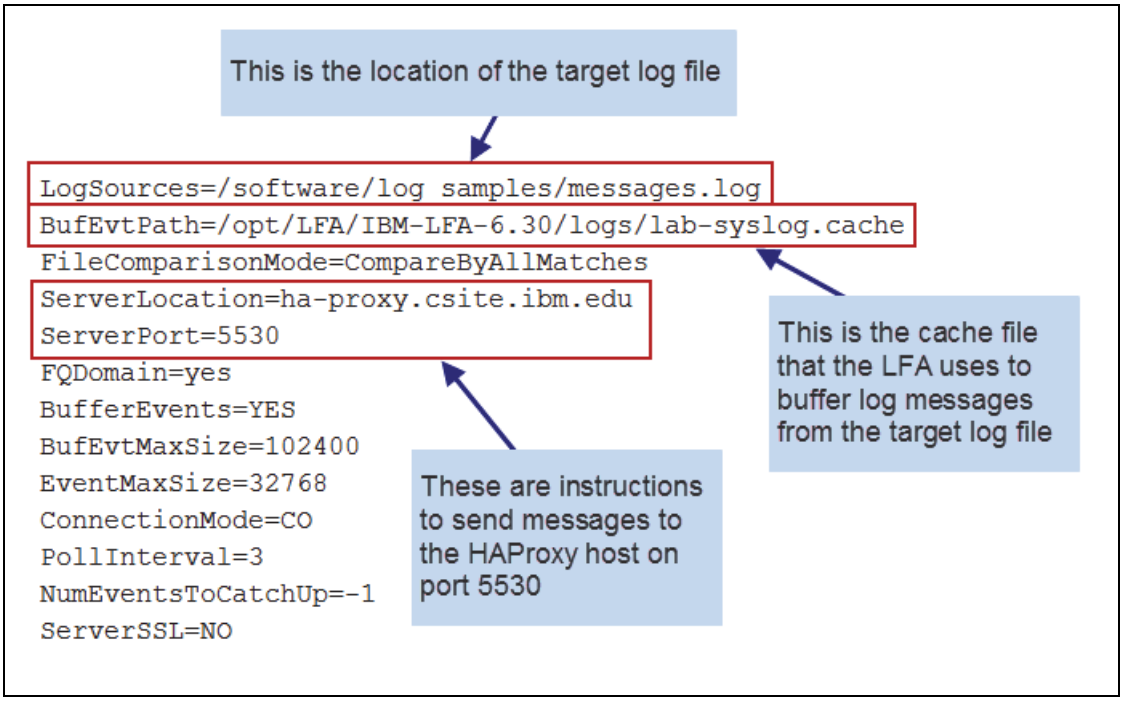


Figure 1 Key fields in the lab-syslog.conf file

d. Run the following command to create a file named `lab-syslog.fmt`.

```
vi /opt/LFA/IBM-LFA-6.30/config/lo/lab-syslog.fmt
```

e. Add the following lines to the `lab-syslog.fmt` file.

```
REGEX AllRecords
(.*)
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
type syslog
instance labInstance
cluster NONE
module syslog
env DEV
functional NONE
site NONE
text $1
END
```

f. Save and close the file when you are finished.

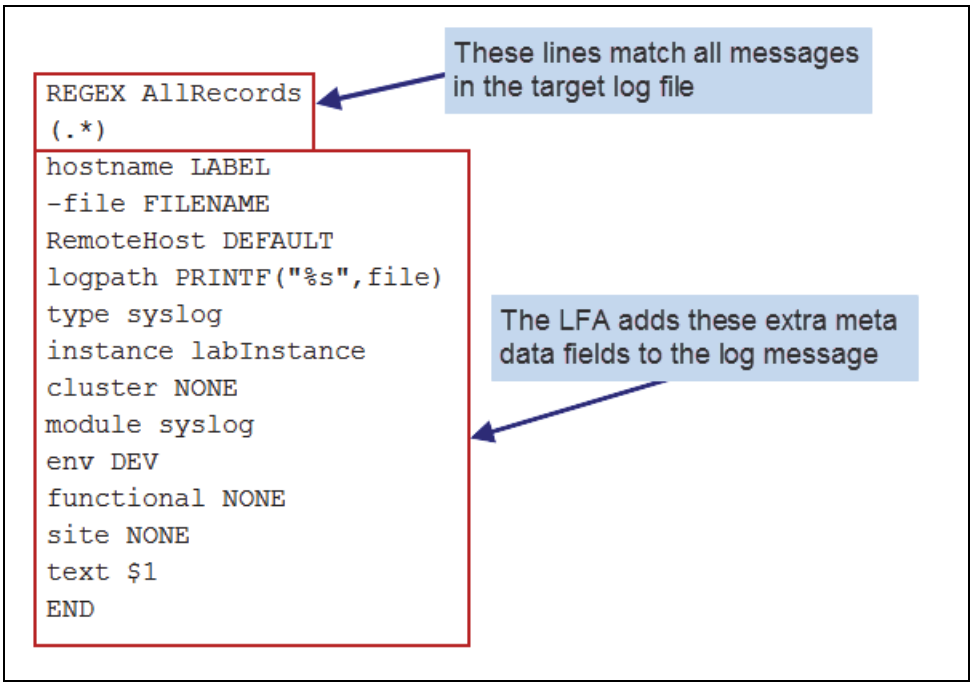


Figure 2 Key fields in the lab-syslog.fmt file

After you create the .conf and .fmt files, the LFA automatically starts monitoring the target log. Verify that the LFA is monitoring the `messages.log` **file.**

a. Run the following command to change to the LFA log directory.

```
cd /opt/LFA/IBM-LFA-6.30/logs
```

b. List the contents of the `/opt/LFA/IBM-LFA-6.30/logs` directory. Look for a log file with a name like the following example. The name of your log file is slightly different than this example.

```
ls
...

collection_lo_default_workload_instance_kloagent_57471f13-01.log
```

Hint: Look for the log file name with the format:
`collection_lo_default_workload_instance_kloagent``_NNNNNaNN-NN.log`

c. Run the following command to look at the most recent messages in the log file. You must change the command to match the name of your log file.

```
tail -50 collection_lo_default_workload_instance_kloagent_57471f13-01.log
```

d. Look for messages like the following example. Messages like these verify that the LFA is monitoring the messages.log file.

```
(574725F4.0001-7:logmonitorqueryclass.cpp,3456,"LogMonitorQueryClass::firstC
ollectDataInit") initializing mdlName=lab-syslogLogfileProfileEvents
/opt/LFA/IBM-LFA-6.30/config/lo/lab-syslog.conf and
/opt/LFA/IBM-LFA-6.30/config/lo/lab-syslog.fmt parsed successfully.

(574725F5.0006-A:kumpthrd.c,119,"KUMP_MarkThreadStarted") File server is
started

(574725F5.0007-A:kumpfile.c,84,"KUMP_FileServer") >>>>> DP file server
ThreadID A started for local file  /software/log_samples/messages.log
```

Note: You can ignore any messages about connection errors to the HAProxy host. You configure HAProxy later in these exercises.

```
(574725F5.0000-7:sockeif.c,390,"_imp_connect") KDE1 connection returned 0x1DE00045
errno 107 for ha-proxy.csite.ibm.edu port 5530

(574725F5.0001-7:sockeif.c,537,"_imp_eipc_create_remote_client") Cannot connect to
ha-proxy.csite.ibm.edu<192.168.100.176> port 5530, rc -1
```

Exercise 2 Configuring HAProxy

In this exercise, you configure HAProxy to listen for traffic from the LFA and forward traffic to the Logstash receiver instances.

Important: Run all of the steps in this exercise on the host named **ha-proxy.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **ha-proxy.csite.ibm.edu**.
2. Configure HAProxy to listen for traffic from the LFA on port and forward traffic to the Logstash receiver instances.
   a. Open the HAProxy configuration file with a text editor.

   ```
   sudo vi /etc/haproxy/haproxy.cfg
   ```

   b. Find the following line.

   ```
   listen stats :9000
   ```

c. Add the following lines above the `listen stats :9000` line:

```
## Listener for LFA Logstash cluster
## Specify Logstash instances for processing data from LFA
listen  LFA_Cluster ha-proxy.csite.ibm.edu:5530
        mode tcp
        balance roundrobin
        server receiver-logstashA r-logstash.csite.ibm.edu:5540 check fall 2
rise 3 inter 1000
        server receiver-logstashB r-logstash.csite.ibm.edu:5541 check fall 2
rise 3 inter 1000
```

d. Save and close the file when you are finished.

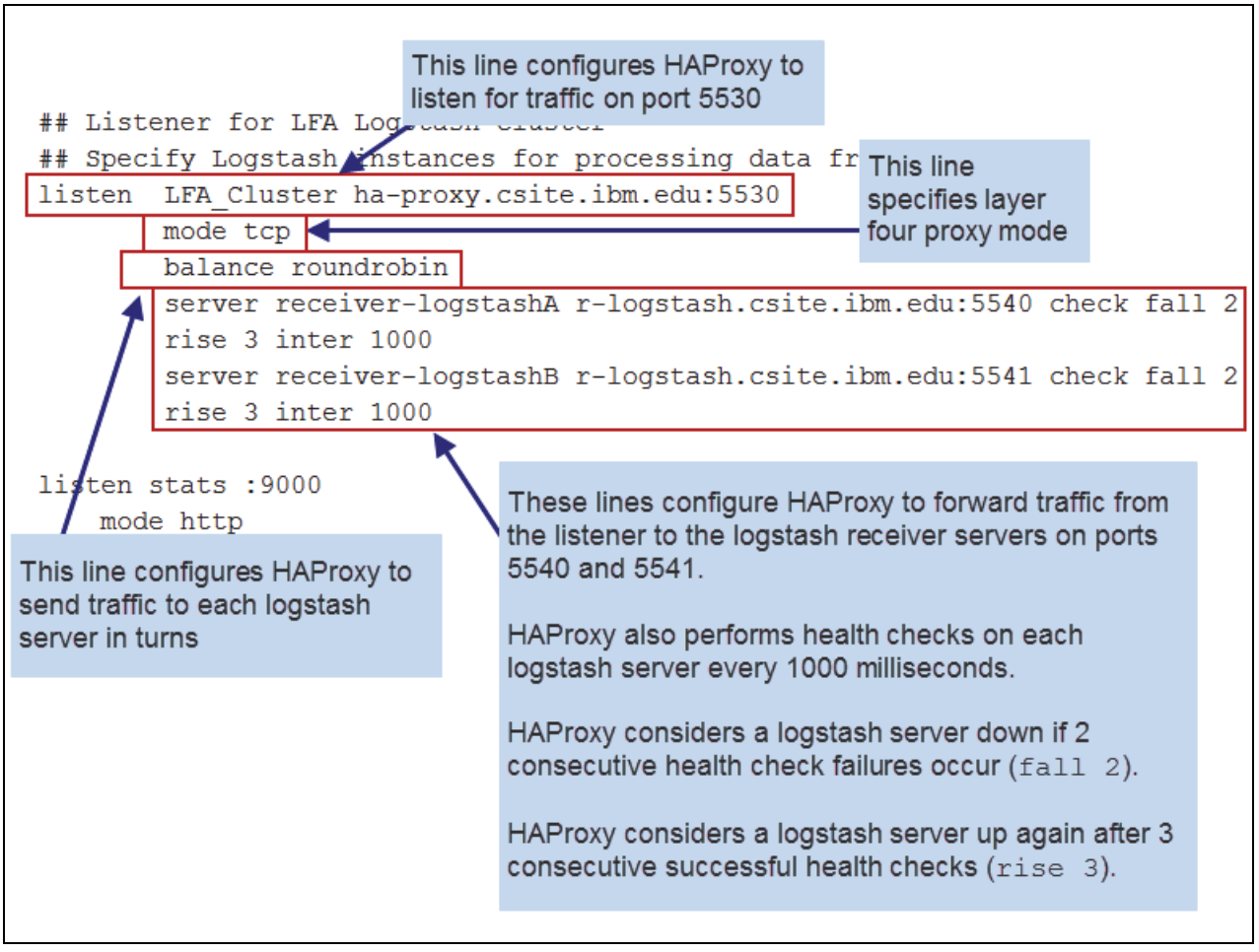


Figure 3 Key fields in the haproxy.cfg file

3. Run the following command to restart HAProxy.

```
sudo service haproxy restart
```

Hint: You can ignore the message about `LFA_Cluster has no server available!` This message is present because you have not configured your Logstash receivers yet.

Exercise 3 Configuring the logstashA receiver

You installed two Logstash receiver instances earlier in this course: logstashA and logstashB. In this exercise, you configure the logstashA receiver instance to process messages from the LFA log file.

Important: You use three different hosts in this exercise. Pay careful attention to the host you are working on when you complete each step.

1. Go to the host named **r-logstash.csite.ibm.edu**.
2. Configure the logstashA server to use traffic from TCP port 5540 as input.
   a. Open the logstashA configuration file in a text editor.
   ```
   vi /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
   ```
   b. Add the following lines in bold typeface to the input section.
   ```
   input {

       beats {
               port => 18737
       } #end beats input

       tcp {
           port => 5540
           type => "lfa"
           codec => line { charset => "US-ASCII" }
       } #end lfa input

   } #end input section
   ```
   c. Save and close the file when you are finished.

Note: These lines invoke the **tcp** input plug-in, set the port to **5540**, and set the type of any messages that arrive on this port to **lfa**. The last line in the tcp input configuration converts the character set to US-ASCII.

3. Restart the logstashA server and watch for activity in the logstashA log file.
   a. Run the following command to stop the logstashA server.

      ```
      pkill -f logstashA
      ```

   b. Run the following command to start the logstashA server.

      ```
      startlogstashA
      ```

   c. Run the following command to watch for activity in the logstashA log file. Leave the `tail` command running.

      ```
      tail -f /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
      ```

4. Go to the host named **collection.csite.ibm.edu**. This is the host where you installed the LFA.
5. Start a tail of the LFA log file. You use this log file to verify that the LFA is sending messages to HAProxy.
   a. Run the following command to change to the LFA log directory.

      ```
      cd /opt/LFA/IBM-LFA-6.30/logs
      ```

   b. List the contents of the `/opt/LFA/IBM-LFA-6.30/logs` directory. Look for a log file with a name like the following example. The name of your log file is slightly different than this example.

      ```
      ls
      ...

      collection_lo_default_workload_instance_kloagent_57471f13-01.log
      ```

Hint: Look for the log file name with the format:
`collection_lo_default_workload_instance_kloagent``_NNNNNaNN-NN.log`

   c. Run the following command to watch for activity in the log file. You must change the command to match the name of your log file.

      ```
      tail -f collection_lo_default_workload_instance_kloagent_57471f13-01.log
      ```

   d. Leave the tail command running.

6. In a different terminal window, run the following command to add more messages to the target log file.

   `/software/log_samples/scripts/Syslog.sh`

7. Verify that the LFA is sending messages to HAProxy.
   a. Return to the terminal window where you are watching the LFA log file.
   b. Look for messages like the following example. Messages like these verify that the LFA is monitoring the `messages.log` file.

   ```
   (57473E54.0000-B:sockeif.c,550,"_imp_eipc_create_remote_client") Connect
   successful to ha-proxy.csite.ibm.edu port 5530
   (57473E54.0001-B:sockeif.c,726,"_imp_do_send") Note: EIF events to
   destination <ha-proxy.csite.ibm.edu> being sent over any local interface
   (57474583.0000-B:sockeif.c,1209,"_imp_eipc_recv_data") KDE1_ReceiveOn
   returned 0x1DE0000B
   ```

   c. Press Ctrl + C to stop the tail command.

8. Go to the host named **r-logstash.csite.ibm.edu**.

9. Look for messages like the following example. Messages like these verify that logstashA is receiving log messages from the LFA.

Important: It might take up to 90 seconds for messages from the LFA to arrive in Logstash.

```
"message" =>
"\\u0001<START>>\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0
000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u
0000\\u0000\\u0000\\u0001\\xBD\\u0000\\u0000\\u0000\\u0004\\u0003AllRecords;
type='syslog';text='2016-05-12T23:39:50.764527+00:00
host=analysishost,relayHost=analysishost, tag=kernel:,
programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
msg=Reserving 129MB of memory at 48MB for crashkernel (System RAM:
9216MB)';RemoteHost='';site='NONE';instance='labInstance';hostname='collecti
on';cluster='NONE';module='syslog';env='DEV';logpath='/software/log_samples/
messages.log';functional='NONE';END",
      "@version" => "1",
    "@timestamp" => "2016-05-26T18:20:08.149Z",
          "host" => "192.168.100.176",
          "port" => 45437,
          "type" => "lfa"
```

The following figures explain the different portions of this message.

```
    "message" =>
    "\\u0001<START>>\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u000
    0\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000
    \\u0000\\u0000\\u0000\\u0000\\u0001\\xBD\\u0000\\u0000\\u0000\\u0004\\u
    0003AllRecords;type='syslog';text='2016-05-12T23:39:50.764527+00:00
    host=analysishost,relayHost=analysishost, tag=kernel:,
    programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
    msg=Reserving 129MB                                  kernel (System RAM:
    9216MB)';RemoteHost=                                 bInstance';hostname='col
    lection';cluster='NO                                 V';logpath='/software/lo
    g_samples/messages.l
  "@version" => "1",
"@timestamp" => "2016-05-26T18:20:08.149Z",
      "host" => "192.168.100.176",
      "port" => 45437,
      "type" => "lfa"
```

This part of the message is automatically added by the LFA. ***You do not need this text.***

```
    "message" =>
    "\\u0001<START>>\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u000
    0\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000
    \\u0000\\u0000\\u0000\\u0000\\u0001\\xBD\\u0000\\u0000\\u0000\\u0004\\u
    0003AllRecords;type='syslog';text='2016-05-12T23:39:50.764527+00:00
    host=analysishost,relayHost=analysishost, tag=kernel:,
    programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
    msg=Reserving 129MB of memory at 48MB for crashkernel (System RAM:
    9216MB)';RemoteHost='';site='NONE';instance='labInstance';hostname='col
    lection';cluster='NONE';module='syslog';env='DEV';logpath='/software/lo
    g_samples/messages.log';fun
  "@version" => "1",
"@timestamp" => "2016-05-26T18
      "host" => "192.168.100.176",
      "port" => 45437,
      "type" => "lfa"
```

This is the actual message from the target log file.

```
    "message" =>
    "\\u0001<START>>\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u000
    0\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000\\u0000
    \\u0000\\u0000\\u0000\\u0000\\u0001\\u0000\\u0000\\u0000\\u0000\\u0004\\u
    0003AllRecords;type='syslog';text='2[illegible]00
    host=analysishost,relayHost=analysis[illegible]
    programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
    msg=Reserving 129MB of memory at 48MB for crashkernel (System RAM:
    9216MB)';RemoteHost='';site='NONE';instance='labInstance';hostname='col
    lection';cluster='NONE';module='syslog';env='DEV';logpath='/software/lo
    g_samples/messages.log';functional='NONE';END",
  "@version" => "1",
"@timestamp" => "2016-05-26T18:20:08.149Z",
      "host" => "192.168.100.176",
      "port" => 45437,
      "type" => "lfa"
```

These meta data fields were added by the LFA because of the .fmt configuration file you created

10. Configure the logstashA server to remove the unwanted text from the message and to use the extra metadata fields.
    a. Open the logstashA configuration file in a text editor.

    ```
    vi /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
    ```

    b. Add the following lines in bold typeface to the filter section.

    ```
    }# end mutate
        } #end filebeat condition

        if [type] == "lfa" {
            grok {
                patterns_dir => "/opt/logstashA/logstash-2.2.1/patterns"
                match => [ "message", "%{LFAMESSAGE}" ]
                add_tag => ["grok_lfa"]
            } #end initial LFA grok
        } #end initial LFA condition

    } #end filter section
    ```

    c. Keep the file open. You add more lines to this file in the next steps.

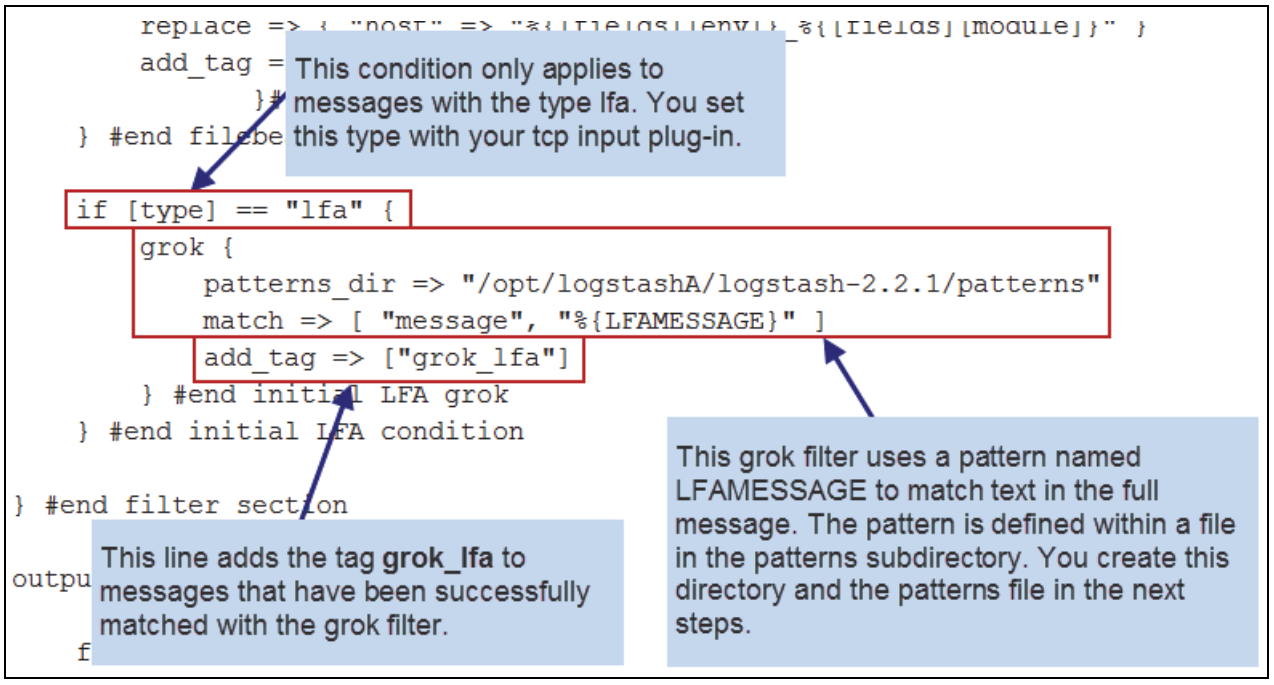


Figure 4 Key fields in the logstashA.conf file

d. Add the following lines to the filter section. Add them below the lines you added in the preceding step.

```
if "grok_lfa" in [tags] {
    mutate {
        replace => ["message","%{LFA_ORIG_MSG}"]
        add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
        add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1"]
    }# end mutate
}# end grok_lfa condition
```

e. Save and close the file when you are finished.

This condition only applies to messages with the tag **grok_lfa**. This tag is set by the grok filter you added earlier.

```
        grok {
            patterns_dir => "/opt/
            match => [ "message",
            add_tag => ["grok_lfa"
        } #end initial LFA grok
    } #end initial LFA condition

    if "grok_lfa" in [tags] {
        mutate {
            replace => ["message","%{LFA_ORIG_MSG}"]
            add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
            add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1"]
        }# end mutate
    }# end grok_lfa condition

} #end filter section

output {

    file {
```

This line replaces the full message from the LFA, including the unwanted characters, with only the actual message from the target log file.

The actual message from the target log file was matched by the grok pattern you added earlier, and identified as LFA_ORIG_MSG.

These two lines create two new fields: **datasource** and **resourceID**. These fields are used to organize data in Kafka. The value of datasource becomes the Kafka topic ID.

Figure 5 Key fields in the logstashA.conf file

In your logstashA configuration, you added a grok filter that used a pattern from a file. You also set the directory where Logstash looks for the pattern file.

11. Create the pattern file directory and copy the pattern file into the new directory.
    a. Run the following commands to create the patterns directory and change into the new directory.

    ```
    mkdir /opt/logstashA/logstash-2.2.1/patterns

    cd /opt/logstashA/logstash-2.2.1/patterns
    ```

    The pattern file that contains the LFAMESSAGE pattern is included with the Log Analysis core software.

    b. Run the following command to copy the pattern file from the Log Analysis server. Enter **yes** if you are prompted about the authenticity of the host. Use the password **object00**.

    ```
    scp
    netcool@192.168.100.180:/opt/IBM/LogAnalysis/kafka/test-configs/logstash-con
    figs/SCALAPATTERNS .

    netcool@192.168.100.180's password: object00
    ```

12. Restart the logstashA server and watch for activity in the logstashA log file.
    a. Run the following command to stop the logstashA server.

    ```
    pkill -f logstashA
    ```

    b. Run the following command to start the logstashA server.

    ```
    startlogstashA
    ```

    c. Run the following command to watch for activity in the logstashA log file. Leave the tail command running.

    ```
    tail -f /opt/logstashA/logstash-2.2.1/log/logstashA-debug.log
    ```

13. Go to the host named **collection.csite.ibm.edu**. You installed the LFA on this host.

14. Run the following command to add more messages to the target log file.

    ```
    /software/log_samples/scripts/Syslog.sh
    ```

15. Go to the host named **r-logstash.csite.ibm.edu**. Verify that the unwanted text from the LFA was removed and that the extra meta fields are present.
    a. Return to the terminal window where the tail command is running.

    b. Look for messages like the following example. Confirm that the unwanted text from the LFA was removed and that the extra meta fields are present.

    ```
    "message" => "2016-05-12T23:39:50.764527+00:00
    host=analysishost,relayHost=analysishost, tag=kernel:,
    programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
    msg=Reserving 129MB of memory at 48MB for crashkernel (System RAM: 9216MB)",
                   "@version" => "1",
                 "@timestamp" => "2016-05-26T13:26:59.627Z",
                       "host" => "192.168.100.176",
                       "port" => 43979,
                       "type" => "lfa",
                   "LFA_TYPE" => "syslog",
               "LFA_ORIG_MSG" => "2016-05-12T23:39:50.764527+00:00
    host=analysishost,relayHost=analysishost, tag=kernel:,
    programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
    msg=Reserving 129MB of memory at 48MB for crashkernel (System RAM: 9216MB)",
                   "LFA_SITE" => "NONE",
               "LFA_INSTANCE" => "labInstance",
               "LFA_HOSTNAME" => "collection",
                 "LFA_MODULE" => "syslog",
        "LFA_ENVIRONMENTNAME" => "DEV",
                "LFA_LOGNAME" => "/software/log_samples/messages.log",
         "LFA_FUNCTIONALNAME" => "NONE",
                       "tags" => [
             [0] "grok_lfa"
    ```

```
],
          "datasource" => "labInstance_syslog",
         "resourceID" => "collection_/software/log_samples/messages.log_1"
```

16. Configure the logstashA server to send messages from the LFA to the Kafka server.
    a. Open the logstashA configuration file in a text editor.

       ```
       vi /opt/logstashA/logstash-2.2.1/conf/logstashA.conf
       ```

    b. Find the following line in the output section:

       ```
       if ("mutate_filebeat" in [tags]) and ! ("_grokparsefailure" in [tags]) {
       ```

    c. Change the line to look like the following example. This change allows messages from Filebeat or the LFA to use the Kafka output plug-in.

       ```
       if ("mutate_filebeat" or "grok_lfa" in [tags]) and ! ("_grokparsefailure"
       in [tags]) {
       ```

    d. Save and close the file when you are finished.

    e. Run the following command to stop the logstashA server.

       ```
       pkill -f logstashA
       ```

    f. Run the following command to start the logstashA server.

       ```
       startlogstashA
       ```

17. Go to the host named **collection.csite.ibm.edu**. You installed the LFA on this host.

18. Run the following command to add more messages to the target log file.

    ```
    /software/log_samples/scripts/Syslog.sh
    ```

19. Go to the host named **kafka.csite.ibm.edu**.

20. Verify that the Logstash receiver created a new topic for the log file that the LFA is monitoring.
    a. Open a terminal window, if you do not already have one open.

    b. Run the following commands to list all topics. Confirm that the topic named **labInstance_syslog** is present.

       ```
       cd /opt/kafka_2.9.1-0.8.2.2/bin/

       ./kafka-topics.sh --list --zookeeper kafka.csite.ibm.edu:17981

       DEV_IBM-HTTP-Server_access-log
       labInstance_syslog
       ```

Note: Notice that the name of the topic is **labInstance_syslog**. This name came from your Logstash receiver configuration and the metadata you added with your LFA configuration:

- From your Logstash receiver filter section:

```
if "grok_lfa" in [tags] {
        mutate {
            replace => ["message","%{LFA_ORIG_MSG}"]
            add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
```

- From your Logstash receiver output section:

```
if ("mutate_filebeat" or "grok_lfa" in [tags]) and ! ("_grokparsefailure" in
[tags]) {
        kafka {
            bootstrap_servers =>"kafka.csite.ibm.edu:17991"
            topic_id => "%{datasource}"
            message_key => "%{resourceID}"
        } #end Kafka output
    } #end Kafka condition
```

The value for %{datasource} is set by the metadata you added in your `lab-syslog.fmt` LFA configuration file:

```
...
instance labInstance
cluster NONE
module syslog
env DEV
functional NONE
site NONE
text $1
END
```

21. Verify that the new topic contains messages from the target log file.
    a. Run the following command on one line. Notice the messages from the syslog target log file.

```
./kafka-console-consumer.sh --zookeeper kafka.csite.ibm.edu:17981 --topic
labInstance_syslog --from-beginning

{"message":"2016-05-12T23:39:50.764527+00:00
host=analysishost,relayHost=analysishost, tag=kernel:,
programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
msg=Reserving 129MB of memory at 48MB for crashkernel (System RAM: 9216MB)
...
```

    b. Press Ctrl + C to stop the output of the messages.

Exercise 4 Configuring the logstashB receiver

You installed two Logstash receiver instances earlier in this course: logstashA and logstashB. In this exercise, you configure the logstashB receiver instance to process messages from the LFA log file.

Important: You use three different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the **host named r-logstash.csite.ibm.edu**.
2. Configure the logstashB server to use traffic from TCP port 5541 as input.
   a. Open the logstashB configuration file in a text editor.

      ```
      vi /opt/logstashB/logstash-2.2.1/conf/logstashB.conf
      ```

   b. Add the following lines in bold typeface to the input section.

      ```
      input {

          beats {
                  port => 18738
          } #end beats input

          tcp {
              port => 5541
              type => "lfa"
              codec => line { charset => "US-ASCII" }
          } #end lfa input

      } #end input section
      ```

   c. Keep the file open. You add more lines to this file in the next steps.

3. Configure the logstashB server to remove the unwanted text from the message and to use the extra metadata fields.
   a. Add the following lines to the filter section.

```
if [type] == "lfa" {
    grok {
        patterns_dir => "/opt/logstashB/logstash-2.2.1/patterns"
        match => [ "message", "%{LFAMESSAGE}" ]
        add_tag => ["grok_lfa"]
    } #end initial LFA grok
} #end initial LFA condition

if "grok_lfa" in [tags] {
    mutate {
        replace => ["message","%{LFA_ORIG_MSG}"]
        add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
        add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1"]
    }# end mutate
}# end grok_lfa condition
```

   b. Keep the file open. You continue to edit this file in the next step.

4. Configure the logstashB server to send messages from the LFA to the Kafka server.
   a. Find the following line in the output section:

```
if ("mutate_filebeat" in [tags]) and ! ("_grokparsefailure" in [tags]) {
```

   b. Change the line to look like the following example. This change allows messages from Filebeat or the LFA to use the Kafka output plug-in.

```
if ("mutate_filebeat" or "grok_lfa" in [tags]) and ! ("_grokparsefailure"
in [tags]) {
```

   c. Save and close the file when you are finished.

5. In your logstashB configuration, you added a grok filter that used a pattern from a file. You also set the directory where Logstash looks for the pattern file. Create the pattern file directory and copy the pattern file into the new directory.
   a. Run the following commands to create the patterns directory and change into the new directory.

```
mkdir /opt/logstashB/logstash-2.2.1/patterns

cd /opt/logstashB/logstash-2.2.1/patterns
```

   The pattern file that contains the LFAMESSAGE pattern is included with the Log Analysis core software. You copied this pattern file in the preceding exercise to your logstashA server.

b. Run the following command to copy the pattern file from the logstashA server to the logstashB server.

```
cp /opt/logstashA/logstash-2.2.1/patterns/SCALAPATTERNS
/opt/logstashB/logstash-2.2.1/patterns
```

6. Restart the logstashB server and watch for activity in the logstashB log file.
   a. Run the following command to stop the logstashB server.

   ```
   pkill -f logstashB
   ```

   b. Run the following command to start the logstashB server.

   ```
   startlogstashB
   ```

   c. Run the following command to watch for activity in the logstashB log file. Leave the tail command running.

   ```
   tail -f /opt/logstashB/logstash-2.2.1/log/logstashB-debug.log
   ```

7. Run the following command to stop the logstashA server. You are stopping the logstashA server to force messages to flow through the logstashB server. This verifies that logstashB can send messages to Kafka.

   ```
   pkill -f logstashA
   ```

8. Go to the host named **kafka.csite.ibm.edu**.

9. Watch the labInstance_syslog topic in Kafka. In the next steps, you confirm that Kafka can receive messages from the logstashB server.
   a. Run the following command to change to the Kafka bin directory.

   ```
   cd /opt/kafka_2.9.1-0.8.2.2/bin
   ```

   b. Run the following command on one line. This command shows new messages in the labInstance_syslog topic.

   ```
   ./kafka-console-consumer.sh --zookeeper kafka.csite.ibm.edu:17981 --topic
   labInstance_syslog
   ```

   c. Leave the command running. You view the output in the next steps to verify that Kafka is receiving messages from logstashB

10. Go to the host named **collection.csite.ibm.edu**. This is the host where you installed the LFA.

11. Run the following command to add more messages to the target log file.

    ```
    /software/log_samples/scripts/Syslog.sh
    ```

12. Go to the host named **r-logstash.csite.ibm.edu**. Verify that the unwanted text from the LFA was removed and that the extra meta fields are present.
    a. Return to the terminal window where the tail command is running for the `logstashB-debug.log` file.
    b. Look for messages like the following example. Confirm that the unwanted text from the LFA was removed and that the extra meta fields are present.

```
"message" => "2016-05-12T23:39:50.764513+00:00
host=analysishost,relayHost=analysishost, tag=kernel:,
programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
msg=found SMP MP-table at [ffff8800000f69b0] f69b0",
               "@version" => "1",
             "@timestamp" => "2016-05-26T13:44:08.550Z",
                   "host" => "192.168.100.176",
                   "port" => 55806,
                   "type" => "lfa",
               "LFA_TYPE" => "syslog",
           "LFA_ORIG_MSG" => "2016-05-12T23:39:50.764513+00:00
host=analysishost,relayHost=analysishost, tag=kernel:,
programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
msg=found SMP MP-table at [ffff8800000f69b0] f69b0",
               "LFA_SITE" => "NONE",
           "LFA_INSTANCE" => "labInstance",
           "LFA_HOSTNAME" => "collection",
             "LFA_MODULE" => "syslog",
    "LFA_ENVIRONMENTNAME" => "DEV",
            "LFA_LOGNAME" => "/software/log_samples/messages.log",
     "LFA_FUNCTIONALNAME" => "NONE",
                   "tags" => [
        [0] "grok_lfa"
    ],
             "datasource" => "labInstance_syslog",
           "resourceID" => "collection_/software/log_samples/messages.log_1"
```

    c. Press Ctrl + C to stop the tail of the `logstashB-debug.log` file.

13. Go to the host named **kafka.csite.ibm.edu**.

14. Verify that the **labInstance_syslog** topic received new messages from the target log file.
    a. Return to the terminal window where the `kafka-console-consumer.sh` command is running.

    Notice the messages from the syslog target log file. Messages like these verify that the logstashB server is receiving messages from the LFA and sending them to Kafka.

```
{"message":"2016-05-12T23:39:50.764513+00:00
host=analysishost,relayHost=analysishost, tag=kernel:,
```

```
programName=kernel,procid=-, facility=kern, sev=info,appName=kernel,
msg=found SMP MP-table at [ffff8800000f69b0]
...
```

b. Press Ctrl + C to stop the output of the messages.

15. Go to the host named **r-logstash.csite.ibm.edu**.

16. Run the following command to start the logstashA server.

```
startlogstashA
```

Exercise 5 Configuring the Logstash sender and the Log Analysis data source

In a preceding exercise, you installed a Logstash sender instance named logstashW. In this exercise, you configure the logstashW server to retrieve messages from the new Kafka topic. You also add a data source in the Log Analysis core software to process messages from the syslog target log file.

Important: You use three different hosts in this exercise. Pay careful attention to the host you are working on when you complete each step.

1. Go to the host named **s-logstash.csite.ibm.edu**.

2. Configure the logstashW server to retrieve messages from the Kafka topic ID: **labInstance_syslog**. Also, add a mutate filter to add **host** and **path** fields.

   a. Open the logstashW configuration file in a text editor.

```
vi /opt/logstashW/logstash-2.2.1/conf/logstashW.conf
```

b. Add the following lines in bold typeface to the input section. These lines pull messages from the Kafka topic named labInstance_syslog.

```
}# end HTTP server log Kafka input

kafka {
    zk_connect => "kafka.csite.ibm.edu:17981"
    group_id => "G-labInstance_syslog"
    topic_id => "labInstance_syslog"
    consumer_threads => 1
    consumer_restart_on_error => true
    consumer_restart_sleep_ms => 100
    decorate_events => true
}# end syslog server log Kafka input

}# end input section
```

c. Add the following lines in bold typeface to the filter section. This mutate filter sets two fields: **path** and **host**. These fields are used when you configure the data source in Log Analysis.

```
        } # end mutate
    } # end filebeat mutate condition

    if "grok_lfa" in [tags] {
        mutate {
          add_field => [ "path", "%{LFA_INSTANCE}_%{LFA_MODULE}" ]
          replace => ["host", "%{LFA_ENVIRONMENTNAME}_%{LFA_MODULE}"]
        } # end mutate
    } # end lfa mutate condition

}# end filter section
```

d. Save and close the file when you are finished.

3. Restart the logstashW server.

   a. Run the following command to stop the logstashW server.

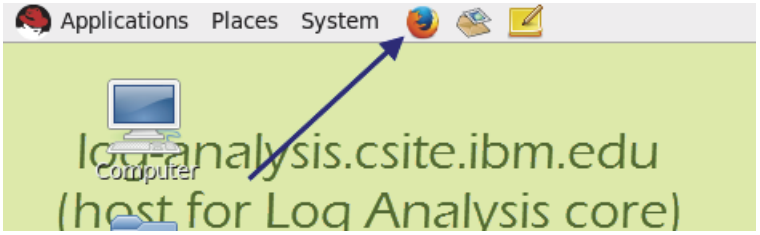
   ```
   pkill -f logstashW
   ```

   b. Run the following command to start the logstashW server.

   ```
   startlogstashW
   ```

4. Go to the host named **log-analysis.csite.ibm.edu**.

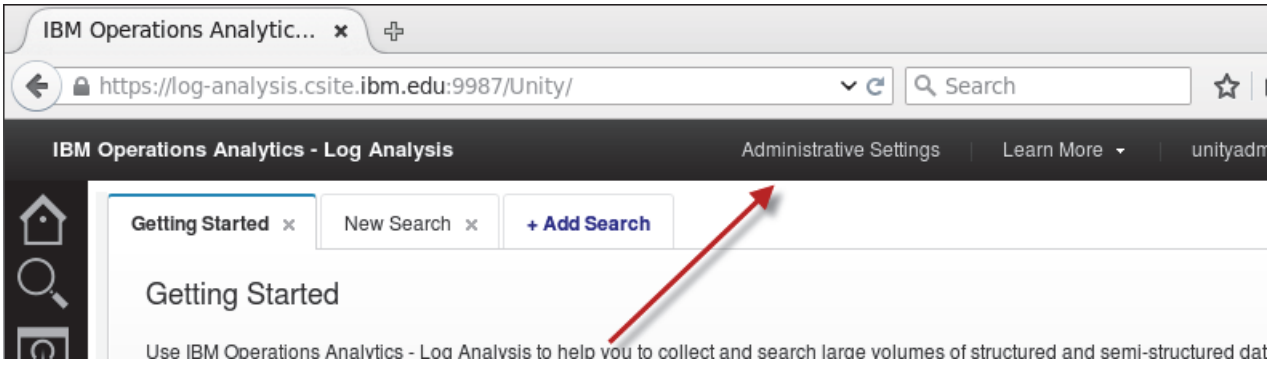
5. Add a data source for the target log file in Log Analysis.
   a. Open a Firefox browser.



   b. Enter the following address:
      https://log-analysis.csite.ibm.edu:9987/Unity

   c. Log in to the user interface with the user name **unityadmin** and the password **object00**.

   d. Click **Administrative Settings**. The administration user interface opens in a new Firefox tab.

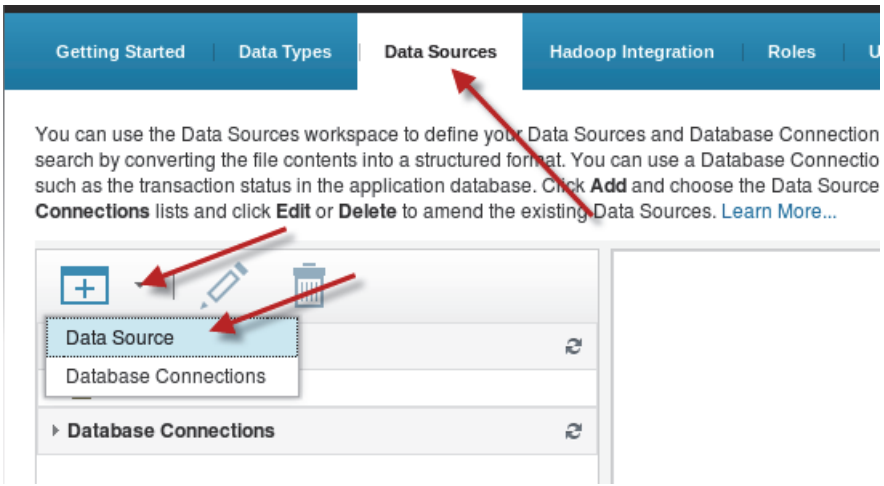


e. Create a data source named **Lab_Syslog**. Use the values in the following table to complete the data source wizard.

Field	Value
Location	Select custom
Host name	dev_syslog
File Path	labInstance_syslog
Type	Syslog_custom
Collection	Leave this field blank
Name	Lab_Syslog
Description	Leave this field blank
Group	Leave this field blank

f. Click the **Data Sources** tab in the administration user interface. The administration user interface is in the second Firefox tab.

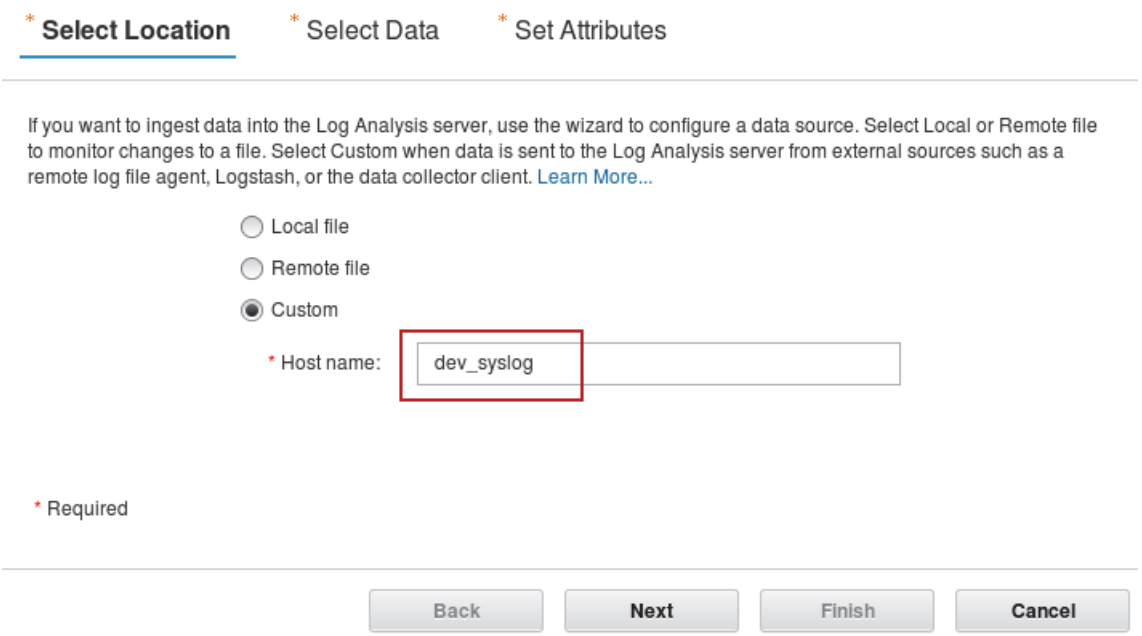
g. Click **Add > Data Source**.



h. Select **Custom**.

i. Enter **dev_syslog** as the host name.

j. Click **Next**.



Note: Notice that the host name is **dev_syslog**. This value corresponds to the value you set with your mutate filter in the Logstash sender configuration and the metadata you added with the LFA.

- From your sender Logstash configuration:

```
replace => ["host", "%{LFA_ENVIRONMENTNAME}_%{LFA_MODULE}"]
```

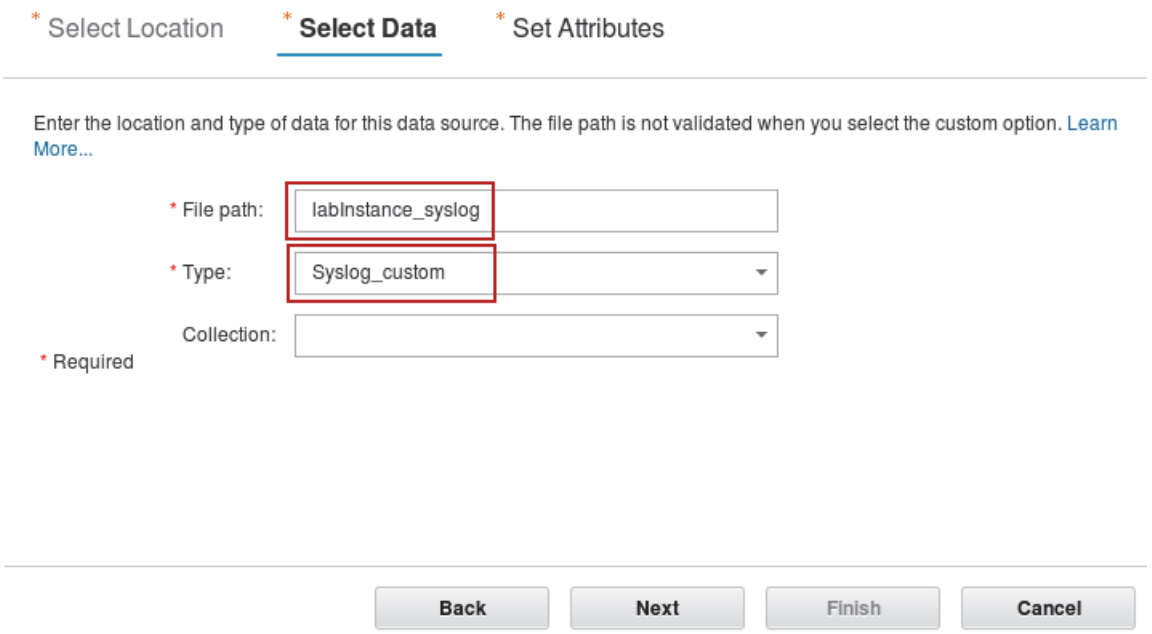
- From your lab-syslog.fmt LFA configuration file:

```
...
type syslog
instance labInstance
cluster NONE
module syslog
env DEV
functional NONE
site NONE
text $1
END
```

k. Enter **labInstance_syslog** as the file path.

l. Select **Syslog_custom** as the type.

m. Click **Next**.



Note: Notice that the file path is **labInstance_syslog**. This value corresponds to the value you set with your mutate filter in the Logstash sender configuration and the metadata you added with the LFA.

- From your sender Logstash configuration:

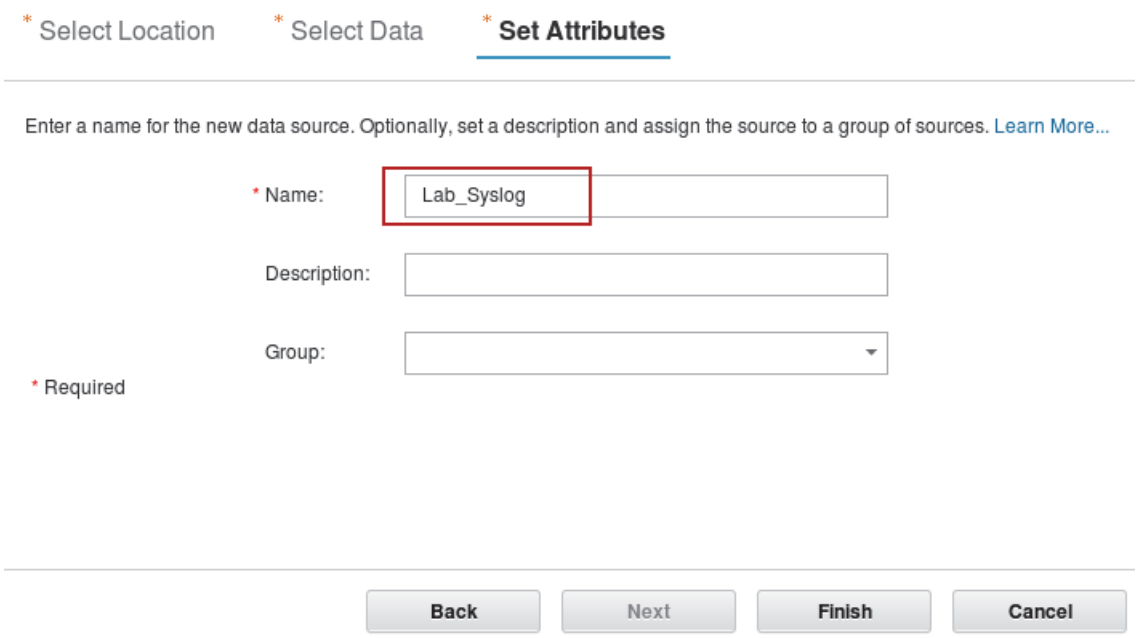
```
add_field => [ "path", "%{LFA_INSTANCE}_%{LFA_MODULE}" ]
```

- From your `lab-syslog.fmt` LFA configuration file:

```
...
type syslog
instance labInstance
cluster NONE
module syslog
env DEV
functional NONE
site NONE
text $1
END
```

n. Enter **Lab_Syslog** as the name of the data source.

o. Click **Finish**.



p. Click **OK** in the confirmation windows.

q. Leave this Firefox page open. You use it again in a moment.

The `GenericReceiver.log` file shows all data coming in to the Log Analysis server.

6. Run the following command to watch for activity in the `GenericReceiver.log` file.

```
tail -f /opt/IBM/LogAnalysis/logs/GenericReceiver.log
```

7. Go to the host named **collection.csite.ibm.edu**.

8. Run the following command to add more messages to the target log file.

```
/software/log_samples/scripts/Syslog.sh
```

9. Return to the host named **log-analysis.csite.ibm.edu**.

10. Look at the `GenericReceiver.log` file.

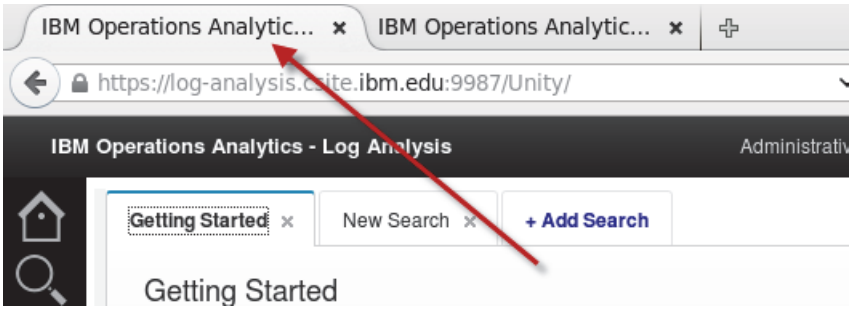
    a. Look for messages like the following example. Messages like these verify that data from the target log file is being processed by the Log Analysis software.

```
5/27/16 18:52:25:204 UTC [Default Executor-thread-15195] INFO  -
UnityFlowController :  Batch Status for -> Lab_Syslog , Size: 113 , Num
successful: 113 , Num failures: 0 , Indexed Source volume: 0

05/27/16 18:52:25:208 UTC [Default Executor-thread-15195] INFO  -
DataCollectorRestServlet : Batch of Size 113 processed and encountered 0
failures
```

b. Press Ctrl + C to stop the tail of the `GenericReceiver.log` file.

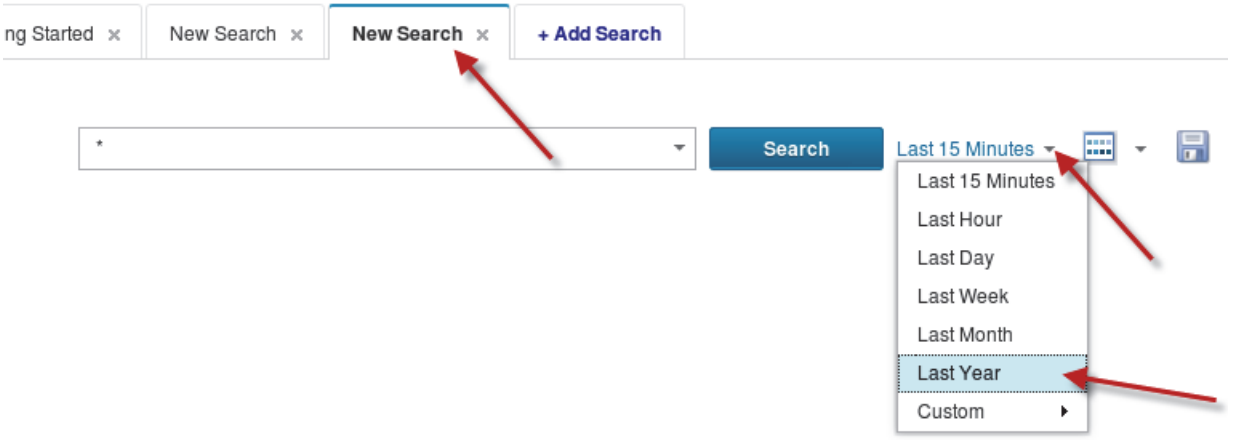
Note: It might take up to 90 seconds for messages to arrive.

11. Verify that messages from the target log file are present in the Log Analysis search interface.
    a. Return to the Log Analysis user interface in the Firefox window. Go to the search interface by clicking the first Firefox tab.



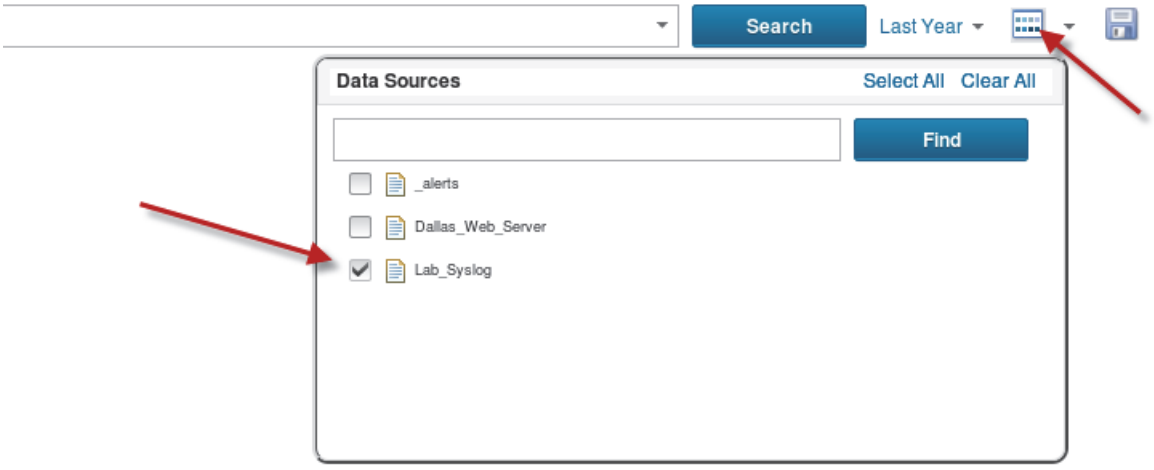
    b. Click the **Add Search** or the **New Search** tab.

    c. Select **Last Year** as the time filter.

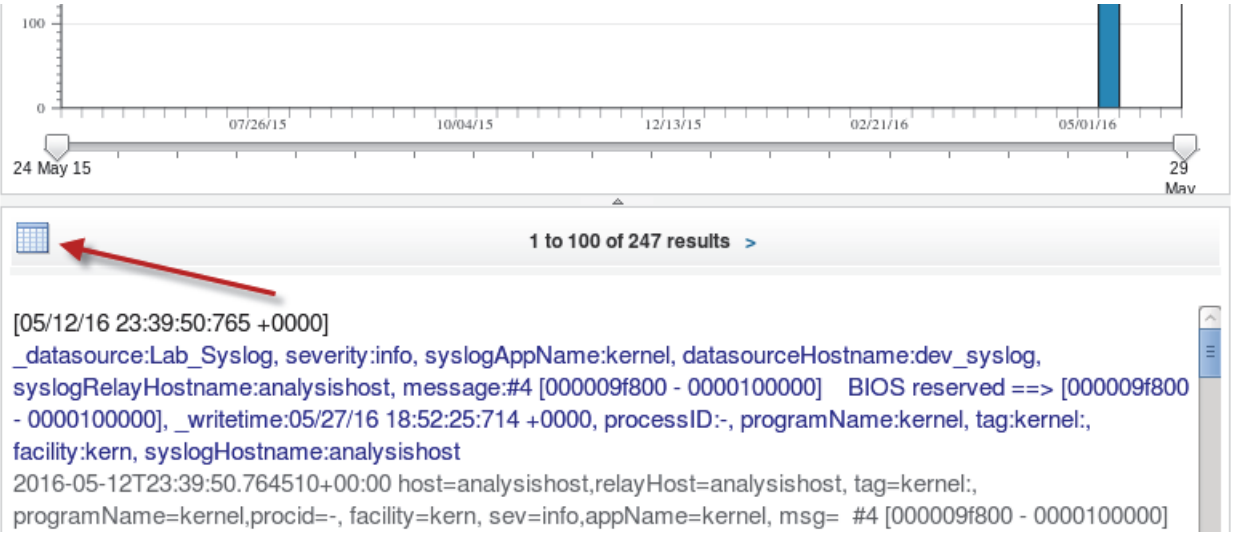


d. Select **Lab_Syslog** as the only data source.

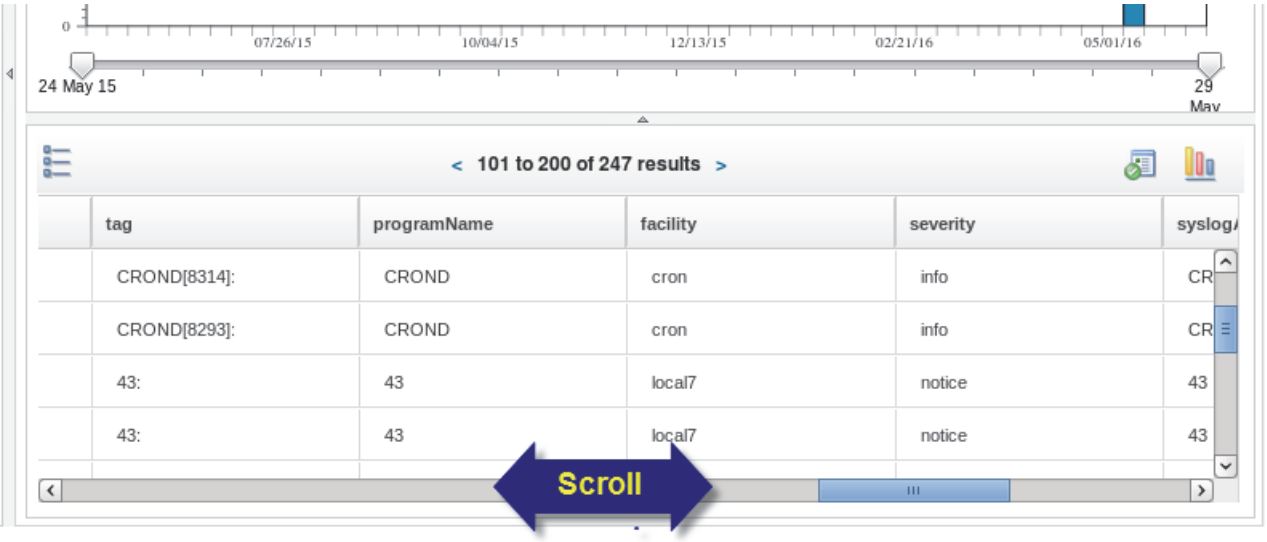
e. Click **Search**.



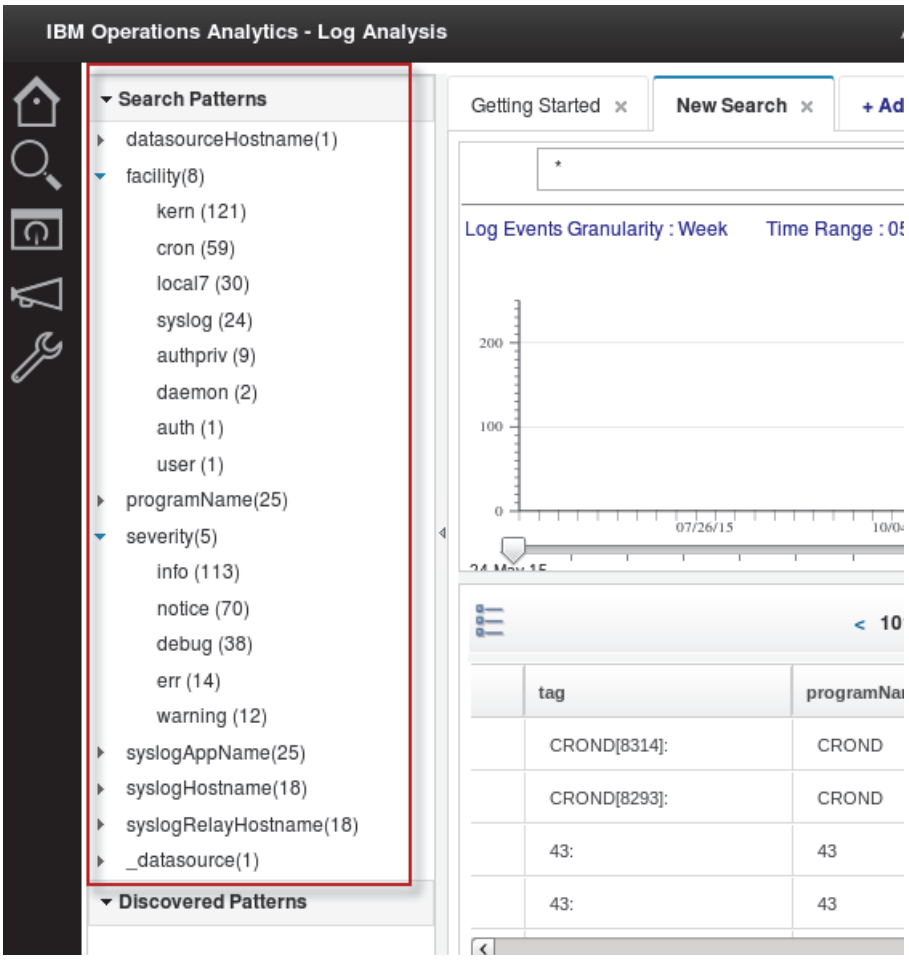
f. Log messages load in to the search interface. Click the **Grid View** button.



g. Scroll left and right to view the columns.



h. Look at the Search Patterns at the left of the search interface. Notice the facet counts and categories from the log file.



Unit 6 Multiline logs exercises

Many log files are in a multiline format. Multiline log records are single log messages that span multiple lines. To process multiline logs in the scalable collection architecture, you must ensure that all of the lines in a single message are processed together, and in the correct sequence. In the exercises for this unit, you add support for a multiline log file.

Exercise 1 Configuring the Log File Agent

In a preceding exercise, you installed the Log File Agent on the host named **collection.csite.ibm.edu**. In this exercise, you configure that Log File Agent to start monitoring a DB2® log file, which is in a multiline format.

Important: Run all of the steps in this exercise on the host named **collection.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **collection.csite.ibm.edu**.
2. Run the following command to verify that the LFA is running.

   ```
   ps -ef | grep -i LFA
   ```

   ```
   netcool   9784     1  0 16:06 ?        00:00:00
   /opt/LFA/IBM-LFA-6.30/lx8266/lo/bin/kloagent
   collection_default_workload_instance
   ```

   If the LFA is not running, use the following command to start it:

   ```
   /opt/LFA/IBM-LFA-6.30/bin/itmcmd agent -o default_workload_instance -f start lo
   ```

3. You configure the LFA to monitor a log file by creating two files: a .conf file and a .fmt file. Create these two files and configure them to monitor a DB2 log file.

   a. Run the following command to create a file named `lab-db2diag.conf`.

   ```
   vi /opt/LFA/IBM-LFA-6.30/config/lo/lab-db2diag.conf
   ```

b. Add the following lines to the `lab-db2diag.conf` file.

```
LogSources=/software/log_samples/DB2_logs/db2diag.log
BufEvtPath=/opt/LFA/IBM-LFA-6.30/logs/lab-db2diag.cache
FileComparisonMode=CompareByAllMatches
ServerLocation=ha-proxy.csite.ibm.edu
ServerPort=5980
FQDomain=yes
BufferEvents=YES
BufEvtMaxSize=102400
EventMaxSize=32768
ConnectionMode=CO
PollInterval=3
NumEventsToCatchUp=-1
ServerSSL=NO
```

c. Save and close the file when you are finished.

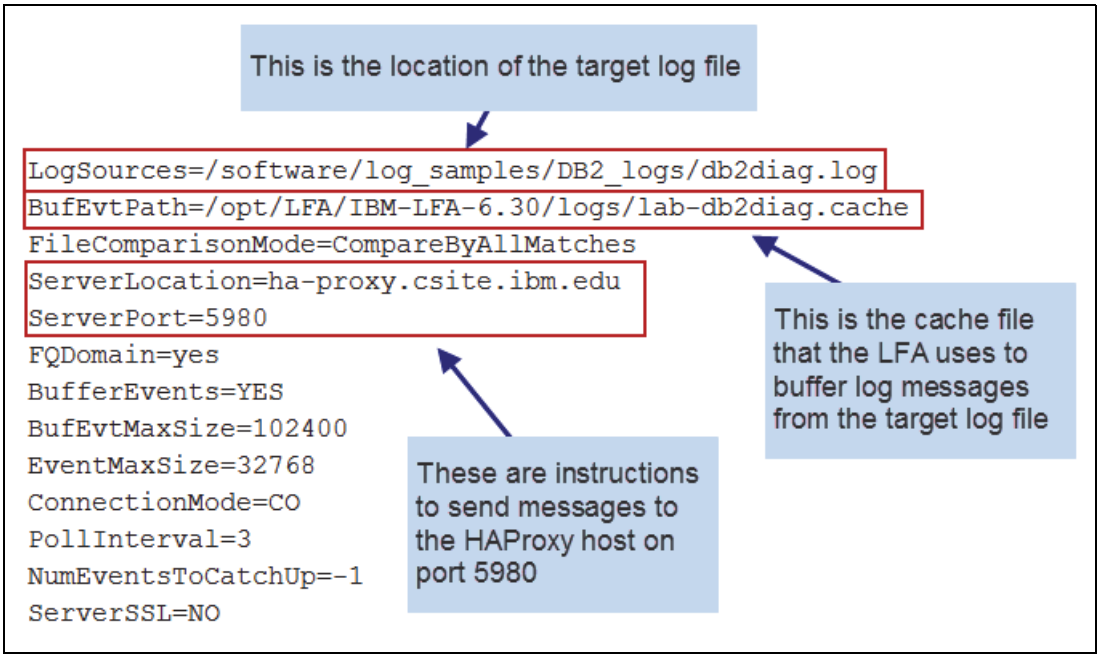


Figure 1 Key fields in the lab-db2diag.conf file

d. Run the following command to create a file named `lab-db2diag.fmt`.

```
vi /opt/LFA/IBM-LFA-6.30/config/lo/lab-db2diag.fmt
```

e. Add the following lines to the `lab-db2diag.fmt` file.

```
REGEX AllRecords
(.*)
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
type db2diag
instance LabDB2Instance
cluster NONE
module db2diag
env TEST
functional NONE
site NONE
text $1
END
```

f. Save and close the file when you are finished.

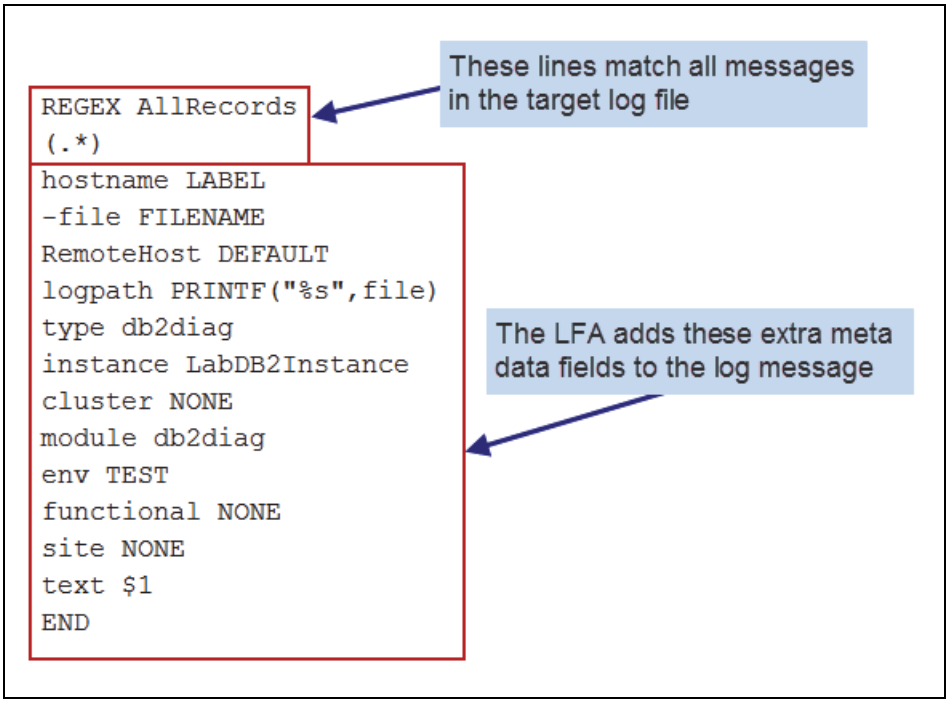


Figure 2 Key fields in the lab-db2diag.fmt file

4. After you create the .conf and .fmt files, the LFA automatically starts monitoring the target log. Verify that the LFA is monitoring the `db2diag.log` file.

   a. Run the following command to change to the LFA log directory.

   ```
   cd /opt/LFA/IBM-LFA-6.30/logs
   ```

b. List the contents of the `/opt/LFA/IBM-LFA-6.30/logs` directory. Look for a log file with a name like the following example. The name of your log file is slightly different than this example.

```
ls
...

collection_lo_default_workload_instance_kloagent_57471f13-01.log
```

Hint: Look for the log file name with this format:
`collection_lo_default_workload_instance_kloagent``_NNNNNNNN-NN.log`

c. Run the following command to look at the most recent messages in the log file. You must change the command to match the name of your log file.

```
tail -50 collection_lo_default_workload_instance_kloagent_57471f13-01.log
```

d. Look for messages like the following example. Messages like these verify that the LFA is monitoring the `db2diag.log` file.

```
(575577D0.0007-A:logmonitorqueryclass.cpp,840,"initLFA")
/opt/LFA/IBM-LFA-6.30/config/lo/lab-db2diag.conf and
/opt/LFA/IBM-LFA-6.30/config/lo/lab-db2diag.fmt parsed successfully.

(575577D1.0000-B:kumpthrd.c,119,"KUMP_MarkThreadStarted") File server is
started

(575577D1.0001-B:kumpfile.c,84,"KUMP_FileServer") >>>>> DP file server
ThreadID B started for local file  /software/log_samples/DB2_logs/db2diag.log
```

Note: You can ignore any messages about connection errors to the HAProxy host. You configure HAProxy later in these exercises.

```
(575577D0.0004-A:sockeif.c,390,"_imp_connect") KDE1 connection returned 0x1DE00045
errno 107 for ha-proxy.csite.ibm.edu port 5980

(575577D0.0005-A:sockeif.c,537,"_imp_eipc_create_remote_client") Cannot connect to
ha-proxy.csite.ibm.edu<192.168.100.176> port 5980, rc -1
```

Exercise 2 Configuring HAProxy

In this exercise, you configure HAProxy to listen for traffic from the LFA and forward traffic to the Logstash receiver instances.

Important: Run all of the steps in this exercise on the host named **ha-proxy.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **ha-proxy.csite.ibm.edu**.
2. Configure HAProxy to listen for traffic from the LFA on port and forward traffic to the Logstash receiver instances.
   a. Open the HAProxy configuration file with a text editor.
   ```
   sudo vi /etc/haproxy/haproxy.cfg
   ```
   b. Find the following line.
   ```
   listen stats :9000
   ```
   c. Add the following lines above the `listen stats :9000` line.
   ```
   ## Listener for LFA Multiline cluster
   ## Specify Logstash instances for processing multiline messages from LFA
   listen  Multiline_LFA_Cluster ha-proxy.csite.ibm.edu:5980
           mode tcp
           balance source
           hash-type consistent
          server receiver-logstashC r-logstash.csite.ibm.edu:5990 check fall 2
   rise 3 inter 1000
          server receiver-logstashD r-logstash.csite.ibm.edu:5991 check fall 2
   rise 3 inter 1000
   ```

d. Save and close the file when you are finished.

```
        server receiver-logstashB r-logstash.csite.ibm.edu:5541 check fall 2
        rise 3 inter 1000

## Listener for LFA Multiline cluster
## Specify Logstash instances for processing multiline mes
listen  Multiline_LFA_Cluster ha-proxy.csite.ibm.edu:5980
        mode tcp
        balance source
        hash-type consistent
        server receiver-logstashC r-logstash.csite.ibm.edu:5990 check fall 2
        rise 3 inter 1000
        server receiver-logstashD r-logstash.csite.ibm.edu:5991 check fall 2
        rise 3 inter 1000

listen stats :9000
    mode http
```

This line configures HAProxy to listen for traffic on port 5980

This line specifies layer four proxy mode

These two lines ensure that all traffic from a collection agent is sent to the same logstash instance, as long as that logstash instance is up.

This is important for multi-line logs because you want all lines from the same log message to go to the same logstash instance.

These lines configure HAProxy to forward traffic from the listener to the logstash receiver servers on ports 5990 and 5591.

HAProxy also performs health checks on each logstash server every 1000 milliseconds.

HAProxy considers a logstash server down if 2 consecutive health check failures occur (`fall 2`).

HAProxy considers a logstash server up again after 3 consecutive successful health checks (`rise 3`).

Figure 3 Key fields in the haproxy.cfg file

3. Run the following command to restart HAProxy.

```
sudo service haproxy restart
```

Note: You can ignore messages like this: `proxy Multiline_LFA_Cluster has no server available!` You install and configure the logstashC and logstashD servers later in this exercise.

Exercise 3 Installing and configuring the first multiline Logstash receiver

Multiline Logstash receivers must run in single-threaded mode so that they process each line of a single message in the correct sequence. In this exercise, you install and configure the first of two Logstash receiver servers.

Important: You use three different hosts in this exercise. Pay careful attention to the host you are working on when you complete each step.

1. Go to the host named **r-logstash.csite.ibm.edu**.
2. Run the following command to create a directory for the first multiline Logstash receiver. In the topology you are building for this lab, the first multiline Logstash receiver instance is called **logstashC**.

   ```
   mkdir /opt/logstashC
   ```

3. Run the following commands to decompress the Logstash installation file and install Logstash into the **/opt/logstashC** directory. Remember, you copied the Logstash installation file from the Log Analysis server in a preceding exercise.

   ```
   cd /software/logstash/

   tar -zxvf logstash-2.2.1.tar.gz -C /opt/logstashC/
   ```

4. Create a directory for the logstashC configuration file.

   ```
   mkdir /opt/logstashC/logstash-2.2.1/conf
   ```

5. Create a directory for the logstashC log file.

   ```
   mkdir /opt/logstashC/logstash-2.2.1/log
   ```

6. Configure the logstashC server.
   a. Open the logstashC configuration file in a text editor.

      ```
      vi /opt/logstashC/logstash-2.2.1/conf/logstashC.conf
      ```

b. Add the following lines to create an input section. These lines configure the logstashC server to use traffic from TCP port 5990 as input.

```
input {

    tcp {
        port => 5990
        type => "lfa"
        codec => line { charset => "US-ASCII" }
    } #end lfa input

} #end input section
```

c. Keep the file open. You add more lines to this file in the next steps.

Note: These lines invoke the **tcp** input plug-in, set the port to **5990**, and set the type of any messages that arrive on this port to **lfa**. The last line in the tcp input configuration converts the character set to US-ASCII.

d. Add the following lines to create a filter section. Add them below the lines you added in the preceding step.

```
filter {

    if [type] == "lfa" {
        grok {
            patterns_dir => "/opt/logstashC/logstash-2.2.1/patterns"
            match => [ "message", "%{LFAMESSAGE}" ]
            add_tag => ["grok_lfa"]
        } #end initial LFA grok
    } #end initial LFA condition

    if "grok_lfa" in [tags] {
        mutate {
            replace => ["message","%{LFA_ORIG_MSG}"]
            add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
            add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1"]
        }# end mutate
    }# end grok_lfa condition

} #end filter section
```

e. Keep the file open. You add more lines to this file in the next steps.

Note: Your filter section performs the following operations:

- The grok filter configures Logstash to remove the unwanted LFA text from the message and to use the extra metadata fields.
- The replace option in the mutate filter replaces the full message from the LFA, including the unwanted characters, with only the actual message from the target log file.
- The add field options in the mutate filter create two new fields: **datasource** and **resourceID**. These fields are used to organize data in Kafka. The value of **datasource** becomes the Kafka topic ID. The value of **resourceID** is used by the Logstash sender later in these exercises.

f. Add the following lines to create an output section. Add them to the bottom of the file.

```
output {

    file {
        path => "/opt/logstashC/logstash-2.2.1/log/logstashC-debug.log"
        codec => rubydebug
    } #end file

    if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
        kafka {
            bootstrap_servers =>"kafka.csite.ibm.edu:17991"
            topic_id => "%{datasource}"
            message_key => "%{resourceID}"
        } #end Kafka output
    } #end Kafka condition

} # end output section
```

g. Save and close the file when you are finished.

Note: These lines configure Logstash to send output to two destinations: a debug log file for logstashC and the Kafka server. Notice that the value of `%{datasource}` from your mutate filter is used as the Kafka topic ID.

7. In your logstashC configuration, you added a grok filter that used a pattern from a file. You also set the directory where Logstash looks for the pattern file. Create the pattern file directory and copy the pattern file into the new directory.
   a. Run the following commands to create the patterns directory and change into the new directory.

      ```
      mkdir /opt/logstashC/logstash-2.2.1/patterns

      cd /opt/logstashC/logstash-2.2.1/patterns
      ```

      The pattern file that contains the LFAMESSAGE pattern is included with the Log Analysis core software. You copied this pattern file in a preceding exercise to your logstashA server.

   b. Run the following command to copy the pattern file from the logstashA server to the logstashC server.

      ```
      cp /opt/logstashA/logstash-2.2.1/patterns/SCALAPATTERNS
      /opt/logstashC/logstash-2.2.1/patterns
      ```

8. In these exercises, you start and stop the logstashC server many times as you test your configuration. Add a command alias to make it easier to start Logstash. You must start and run this Logstash instance in single-threaded mode so that it processes each line of a single message in the correct sequence.
   a. Open the `.bashrc` file of the netcool user in a text editor.

      ```
      vi /home/netcool/.bashrc
      ```

   b. Add the following line to the bottom of the file.

      ```
      alias startlogstashC='/opt/logstashC/logstash-2.2.1/bin/logstash -w 1 -f
      /opt/logstashC/logstash-2.2.1/conf/logstashC.conf -l
      /opt/logstashC/logstash-2.2.1/log/logstashC-debug.log &'
      ```

   c. Save and close the file when you are finished.

Important: Notice the option **`-w 1`**. This option starts Logstash in single-threaded mode so that it processes each line of a single message in the correct sequence.

   d. Run the following command to source the modified environment file.

      ```
      source /home/netcool/.bashrc
      ```

   e. Run the following command to start the logstashC server.

      ```
      startlogstashC
      ```

9. Run the following command to watch for activity in the logstashC log file. Leave the `tail` command running.

   ```
   tail -f /opt/logstashC/logstash-2.2.1/log/logstashC-debug.log
   ```

10. Go to the host named **collection.csite.ibm.edu**. You installed the LFA on this host.

11. Run the following command to add more messages to the target log file.

    `/software/log_samples/scripts/DB2_Logs.sh`

12. Go to the host named **r-logstash.csite.ibm.edu**. Verify that the unwanted text from the LFA was removed and that the extra meta fields are present.

    a. Return to the terminal window where the `tail` command is running.

    b. Look for messages like the following example. Confirm that the unwanted text from the LFA was removed and that the extra meta fields are present.{

```
                 "message" => "AUTHID  : NCIM    ",
                "@version" => "1",
              "@timestamp" => "2016-06-06T15:50:49.294Z",
                    "host" => "192.168.100.176",
                    "port" => 59885,
                    "type" => "lfa",
                "LFA_TYPE" => "db2diag",
            "LFA_ORIG_MSG" => "AUTHID  : NCIM    ",
                "LFA_SITE" => "NONE",
            "LFA_INSTANCE" => "LabDB2Instance",
            "LFA_HOSTNAME" => "collection",
              "LFA_MODULE" => "db2diag",
     "LFA_ENVIRONMENTNAME" => "TEST",
             "LFA_LOGNAME" => "/software/log_samples/DB2_logs/db2diag.log",
      "LFA_FUNCTIONALNAME" => "NONE",
                    "tags" => [
         [0] "grok_lfa"
     ],
              "datasource" => "LabDB2Instance_db2diag",
              "resourceID" =>
"collection_/software/log_samples/DB2_logs/db2diag.log_1"
```

Note: It might take up to 90 seconds for messages from the LFA to arrive in Logstash.

    c. Press Ctrl + C to stop the `tail` command.

13. Go to the host named **kafka.csite.ibm.edu**.

14. Verify that the Logstash receiver created a new topic for the log file that the LFA is monitoring.
    a. Open a terminal window, if you do not already have one open.
    b. Run the following commands to list all topics. Confirm that the topic named **LabDB2Instance_db2diag** is present.

```
cd /opt/kafka_2.9.1-0.8.2.2/bin/

./kafka-topics.sh --list --zookeeper kafka.csite.ibm.edu:17981

DEV_IBM-HTTP-Server_access-log
LabDB2Instance_db2diag
labInstance_syslog
```

Note: Notice that the name of the topic is **LabDB2Instance_db2diag**. This name came from your Logstash receiver configuration and the metadata you added with your LFA configuration:

- From your Logstash receiver filter section:

```
if "grok_lfa" in [tags] {
        mutate {
            replace => ["message","%{LFA_ORIG_MSG}"]
            add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
```

- From your Logstash receiver output section:

```
    if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
        kafka {
            bootstrap_servers =>"kafka.csite.ibm.edu:17991"
            topic_id => "%{datasource}"
            message_key => "%{resourceID}"
        } #end Kafka output
    } #end Kafka condition
```

The value for `%{datasource}` is set by the metadata you added in your `lab-db2diag.fmt` LFA configuration file:

```
...
instance LabDB2Instance
cluster NONE
module db2diag
env TEST
functional NONE
site NONE
text $1
END
```

15. Verify that the new topic contains messages from the target log file.
    a. Run the following command on one line. Notice the messages from the DB2 target log file.

```
./kafka-console-consumer.sh --zookeeper kafka.csite.ibm.edu:17981 --topic
LabDB2Instance_db2diag --from-beginning

...
{"message":"AUTHID  : NCIM
","@version":"1","@timestamp":"2016-06-06T15:50:49.294Z","host":"192.168.100.17
6","port":59885,"type":"lfa","LFA_TYPE":"db2diag","LFA_ORIG_MSG":"AUTHID  :
NCIM
","LFA_SITE":"NONE","LFA_INSTANCE":"LabDB2Instance","LFA_HOSTNAME":"collection"
,"LFA_MODULE":"db2diag","LFA_ENVIRONMENTNAME":"TEST","LFA_LOGNAME":"/software/l
og_samples/DB2_logs/db2diag.log","LFA_FUNCTIONALNAME":"NONE","tags":["grok_lfa"
],"datasource":"LabDB2Instance_db2diag","resourceID":"collection_/software/log_
samples/DB2_logs/db2diag.log_1"}
...
```

    b. Press **Ctrl + C** to stop the output of the messages.

Exercise 4 Installing and configuring the second multiline Logstash receiver

In a preceding exercise, you installed and configured a Logstash receiver instance named logstashC. In this exercise, you add redundancy to your environment by installing and configuring a second Logstash receiver named logstashD.

Important: Run all of the steps in this exercise on the host named **r-logstash.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **r-logstash.csite.ibm.edu**.
2. Run the following command to create a directory for the second multiline Logstash receiver. In the topology you are building for this lab, the second multiline Logstash receiver instance is named **logstashD**.

```
mkdir /opt/logstashD
```

3. Run the following commands to decompress the Logstash installation file and install Logstash into the **/opt/logstashD** directory. Remember, you copied the Logstash installation file from the Log Analysis server in a preceding exercise.

```
cd /software/logstash/

tar -zxvf logstash-2.2.1.tar.gz -C /opt/logstashD/
```

4. Create a directory for the logstashD configuration file.

```
mkdir /opt/logstashD/logstash-2.2.1/conf
```

5. Create a directory for the logstashD log file.

```
mkdir /opt/logstashD/logstash-2.2.1/log
```

6. Configure the logstashD server.
   a. The configuration of the logstashD server is almost identical to the logstashC configuration. Run the following command to copy the logstashC configuration to logstashD. Run the entire command on one line.

```
cp /opt/logstashC/logstash-2.2.1/conf/logstashC.conf
/opt/logstashD/logstash-2.2.1/conf/logstashD.conf
```

   b. Open the logstashD configuration file in a text editor.

```
vi /opt/logstashD/logstash-2.2.1/conf/logstashD.conf
```

   c. Find the following line in the input section:

```
port => 5990
```

   d. Change the line to use port number 5991.

```
port => 5991
```

   e. Find the following line in the filter section:

```
patterns_dir => "/opt/logstashC/logstash-2.2.1/patterns"
```

   f. Change the line to use the logstashD directory.

```
patterns_dir => "/opt/logstashD/logstash-2.2.1/patterns"
```

   g. Find the following line in the output section:

```
path => "/opt/logstashC/logstash-2.2.1/log/logstashC-debug.log"
```

   h. Change the line to use the logstashD debug file.

```
path => "/opt/logstashD/logstash-2.2.1/log/logstashD-debug.log"
```

   i. Save and close the file when you are finished.

7. In your logstashD configuration, you added a grok filter that used a pattern from a file. You also set the directory where Logstash looks for the pattern file. Create the pattern file directory and copy the pattern file into the new directory.
   a. Run the following commands to create the patterns directory and change into the new directory.

      ```
      mkdir /opt/logstashD/logstash-2.2.1/patterns

      cd /opt/logstashD/logstash-2.2.1/patterns
      ```

      The pattern file that contains the LFAMESSAGE pattern is included with the Log Analysis core software. You copied this pattern file in a preceding exercise to your logstashA server.

   b. Run the following command to copy the pattern file from the logstashA server to the logstashD server.

      ```
      cp /opt/logstashA/logstash-2.2.1/patterns/SCALAPATTERNS
      /opt/logstashD/logstash-2.2.1/patterns
      ```

8. In these exercises, you start and stop the logstashD server many times as you test your configuration. Add a command alias to make it easier to start Logstash. You must start and run this Logstash instance in single-threaded mode so that it processes each line of a single message in the correct sequence.
   a. Open the `.bashrc` file of the netcool user in a text editor.

      ```
      vi /home/netcool/.bashrc
      ```

   b. Add the following line to the bottom of the file.

      ```
      alias startlogstashD='/opt/logstashD/logstash-2.2.1/bin/logstash -w 1 -f
      /opt/logstashD/logstash-2.2.1/conf/logstashD.conf -l
      /opt/logstashD/logstash-2.2.1/log/logstashD-debug.log &'
      ```

   c. Save and close the file when you are finished.

Important: Notice the option **`-w 1`**. This option starts Logstash in single-threaded mode so that it processes each line of a single message in the correct sequence.

   d. Run the following command to source the modified environment file.

      ```
      source /home/netcool/.bashrc
      ```

   e. Run the following command to start the logstashD server.

      ```
      startlogstashD
      ```

Exercise 5 Installing and configuring the multiline Logstash sender

Logstash senders pull messages from the Kafka brokers, process the log messages, and send them to the Log Analysis core software. In this exercise, you install and configure a Logstash sender server for multiline logs.

Important: You use two different hosts in this exercise. Pay careful attention to the host you are working on when you complete each step.

1. Go to the host named **s-logstash.csite.ibm.edu**.
2. Run the following command to create a directory for the multiline Logstash sender. In the topology you are building for this lab, the multiline Logstash sender instance is named **logstashX**.

   ```
   mkdir /opt/logstashX
   ```

3. Run the following commands to decompress the installation file and install Logstash into the **/opt/logstashX** directory. Remember, you copied the Logstash installation file from the Log Analysis server in a preceding exercise.

   ```
   cd /software/logstash/

   tar -zxvf logstash-2.2.1.tar.gz -C /opt/logstashX/
   ```

4. Create a directory for the logstashX configuration file.

   ```
   mkdir /opt/logstashX/logstash-2.2.1/conf
   ```

5. Create a directory for the logstashX log file.

   ```
   mkdir /opt/logstashX/logstash-2.2.1/log
   ```

6. Configure the logstashX server.
   a. Open the logstashX configuration file in a text editor.

      ```
      vi /opt/logstashX/logstash-2.2.1/conf/logstashX.conf
      ```

b. Add the following lines to create an input section. These lines configure the logstashX server to retrieve messages from the Kafka topic named **LabDB2Instance_db2diag**.

```
input {

    kafka {
        zk_connect => "kafka.csite.ibm.edu:17981"
        group_id => "G-LabDB2Instance_db2diag"
        topic_id => "LabDB2Instance_db2diag"
        consumer_threads => 1
        consumer_restart_on_error => true
        consumer_restart_sleep_ms => 100
        decorate_events => true
    }# end DB2 server log Kafka input

}# end input section
```

c. Keep the file open. You add more lines to this file in the next steps.

Note: These lines in your output configuration perform the following operations.

- Connect to the Zookeeper server on port 17981
- Pull messages from the topic named LabDB2Instance_db2diag
- Set the Kafka consumer group for this Logstash instance to G-LabDB2Instance_db2diag
- Add metadata to the incoming messages, such as topic ID

d. Add the following lines to create a filter section. Add them below the lines you added in the preceding step.

```
filter {

    if "grok_lfa" in [tags] {
        mutate {
          add_field => [ "path", "%{LFA_INSTANCE}_%{LFA_MODULE}" ]
          replace => ["host", "%{LFA_ENVIRONMENTNAME}_%{LFA_MODULE}"]
        } # end mutate
    } # end lfa mutate condition

}# end filter section
```

e. Keep the file open. You add more lines to this file in the next steps.

Note: Your filter section sets two fields: **path** and **host**. These fields are used when you configure the data source in Log Analysis.

f. Add the following lines to create an output section. Add them to the bottom of the file.

```
output {

        file {
            path => "/opt/logstashX/logstash-2.2.1/log/logstashX-debug.log"
            codec => rubydebug
        }#end file

        scala {
            scala_url =>
    "https://log-analysis.csite.ibm.edu:9987/Unity/DataCollector"
            scala_user => "unityadmin"
            scala_password => "object00"
            scala_keystore_path => ""
            batch_size => 500000
            idle_flush_time => 5
            sequential_flush => true
            num_concurrent_writers => 20
            use_structured_api => false
            disk_cache_path => "/opt/logstashX/training/cache/basecache"
            date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
            log_file => "/opt/logstashX/logstash-2.2.1/log/scala_logstashX.log"
            log_level => "info"
            metadata_fields => {
              "TEST_db2diag@LabDB2Instance_db2diag" => {
                "field_names" => "resourceID"
                "field_paths" => "resourceID"
              } # end db2diag meta data
            } # end meta data fields

        }#end scala output

    }# end output section
```

g. Save and close the file when you are finished.

Important: These lines configure Logstash to send output to two destinations: a debug log file for logstashX and the Log Analysis server.

Notice the following lines in your scala output plug-in configuration:

```
metadata_fields => {
  "TEST_db2diag@LabDB2Instance_db2diag" => {
    "field_names" => "resourceID"
    "field_paths" => "resourceID"
```

These lines ensure that the Logstash server sends messages from the same physical data source together to Log Analysis in the same batch. Notice that the field names and paths use the value of **resourceID**. Remember, you set the value of **resourceID** with a filter in your multiline Logstash receiver configuration.

- From your multiline Logstash receiver filter configuration:

```
mutate {
    replace => ["message","%{LFA_ORIG_MSG}"]
    add_field => [ "datasource", "%{LFA_INSTANCE}_%{LFA_MODULE}"]
    add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1"]
```

In this example, the value of **resourceID** is: `collection_/software/log_samples/DB2_logs/db2diag.log_1`.

You also used **resourceID** as the Kafka message key in your Logstash receiver output configuration.

- From your multiline Logstash receiver output configuration:

```
kafka {
    bootstrap_servers =>"kafka.csite.ibm.edu:17991"
    topic_id => "%{datasource}"
    message_key => "%{resourceID}"
} #end Kafka output
```

This ensures that the Logstash sender can identify messages from Kafka for this specific physical data source, even if messages from other physical data sources are in the same Kafka topic and partition.

The `metadata_fields` configuration in your Logstash sender ensures that messages with the same **resourceID** are sent from the Logstash sender to the Log Analysis server together in the same batches. This is important for multiline logs, because multiple lines from the same log message must arrive in sequence and only with other lines from the same actual log file.

A best practice is to configure the value of **resourceID** to be a string that uniquely identifies the physical log source within your entire environment.

7. Logstash senders use a custom output plug-in named scala to send messages to Log Analysis. Run the following commands to copy the custom output plug-in to the Logstash working directory.

```
cd /opt/logstashX/logstash-2.2.1/vendor/bundle/jruby/1.9/gems/
logstash-core-2.2.1-java/lib/logstash/outputs/

cp /software/scala_plugin/* .
```

8. Run the following command to create the base cache directory for logstashX.

```
mkdir -p /opt/logstashX/training/cache/basecache
```

9. In these exercises, you start and stop the logstashX server many times as you test your configuration. Add a command alias to make it easier to start Logstash. You must start and run this Logstash instance in single-threaded mode so that it processes each line of a single message in the correct sequence.

   a. Open the `.bashrc` file of the **netcool** user in a text editor.

   ```
   vi /home/netcool/.bashrc
   ```

   b. Add the following line to the bottom of the file.

   ```
   alias startlogstashX='/opt/logstashX/logstash-2.2.1/bin/logstash -w 1 -f
   /opt/logstashX/logstash-2.2.1/conf/logstashX.conf -l
   /opt/logstashX/logstash-2.2.1/log/logstashX-debug.log &'
   ```

   c. Save and close the file when you are finished.

Important: Notice the option **`-w 1`**. This option starts Logstash in single-threaded mode so that it processes each line of a single message in the correct sequence.

   d. Run the following command to source the modified environment file.

   ```
   source /home/netcool/.bashrc
   ```

   e. Run the following command to start the logstashX server.

   ```
   startlogstashX
   ```

Note: You can ignore any warning messages about the scala plug-in.

10. Run the following command to watch for activity in the logstashX log file. Leave the `tail` command running.

```
tail -f /opt/logstashX/logstash-2.2.1/log/logstashX-debug.log
```

11. Go to the host named **collection.csite.ibm.edu**. You installed the LFA on this host.

12. Run the following command to add more messages to the target log file.

    `/software/log_samples/scripts/DB2_Logs.sh`

Note: It might take up to 90 seconds for messages from the LFA to arrive in Logstash.

13. Go to the host named **s-logstash.csite.ibm.edu**. Verify that the multiline Logstash sender is processing messages.
    a. Return to the terminal window where the `tail` command is running.
    b. Look at the `logstashX-debug.log` file. Look for messages like the following example. Messages like these verify that Logstash is pulling log messages from the Kafka server.
       - Notice the metadata about Kafka at the bottom of each message.
       - Notice the host and path that were set by the mutate filter:

```
                 "message" => "AUTHID  : NCIM    ",
                "@version" => "1",
              "@timestamp" => "2016-06-06T19:05:49.209Z",
                    "host" => "TEST_db2diag",
                    "port" => 36109,
                    "type" => "lfa",
                "LFA_TYPE" => "db2diag",
            "LFA_ORIG_MSG" => "AUTHID  : NCIM    ",
                "LFA_SITE" => "NONE",
            "LFA_INSTANCE" => "LabDB2Instance",
            "LFA_HOSTNAME" => "collection",
              "LFA_MODULE" => "db2diag",
    "LFA_ENVIRONMENTNAME" => "TEST",
             "LFA_LOGNAME" => "/software/log_samples/DB2_logs/db2diag.log",
     "LFA_FUNCTIONALNAME" => "NONE",
                    "tags" => [
        [0] "grok_lfa"
    ],
              "datasource" => "LabDB2Instance_db2diag",
              "resourceID" =>
"collection_/software/log_samples/DB2_logs/db2diag.log_1",
                   "kafka" => {
                "msg_size" => 541,
                   "topic" => "LabDB2Instance_db2diag",
           "consumer_group" => "G-LabDB2Instance_db2diag",
               "partition" => 0,
                     "key" => byte[99, 111, 108, 108, 101, 99, 116, 105, 111,
110, 95, 47, 115, 111, 102, 116, 119, 97, 114, 101, 47, 108, 111, 103, 95,
```

```
115, 97, 109, 112, 108, 101, 115, 47, 68, 66, 50, 95, 108, 111, 103, 115, 47,
100, 98, 50, 100, 105, 97, 103, 46, 108, 111, 103, 95, 49]@170a52ce
    },
                    "path" => "LabDB2Instance_db2diag"
```

c. Press Ctrl + C to stop the `tail` command.

14. Run the following command to look at the `scala_logstashX.log` file. Look for messages like the following example. Messages like these verify that the logstashX server is sending messages to the Log Analysis server, but Log Analysis does not have a corresponding data source configured.

```
tail -20 /opt/logstashX/logstash-2.2.1/log/scala_logstashX.log

06/06/16 19:06:10:633 UTC [Thread-12] ERROR ScalaCollector$CollectorRunnable -
Error occurred while processing batch
{"RESPONSE_CODE":404,"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":"CTGLA0401E :
Missing data source"}
```

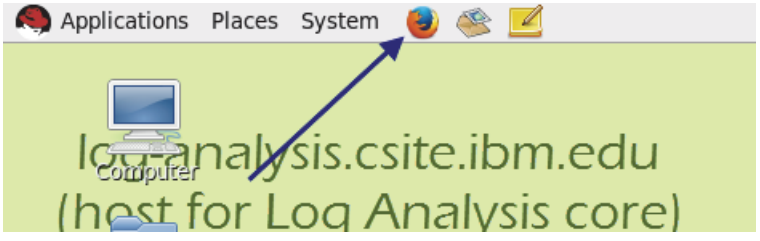
Note: In the next steps, you configure a data source in Log Analysis.

Exercise 6 Sending data to Log Analysis

In this exercise, you configure Log Analysis to accept data from the DB2 multiline log file.

Important: You use two different hosts in this exercise. Pay careful attention to the host you are working on when you complete each step.

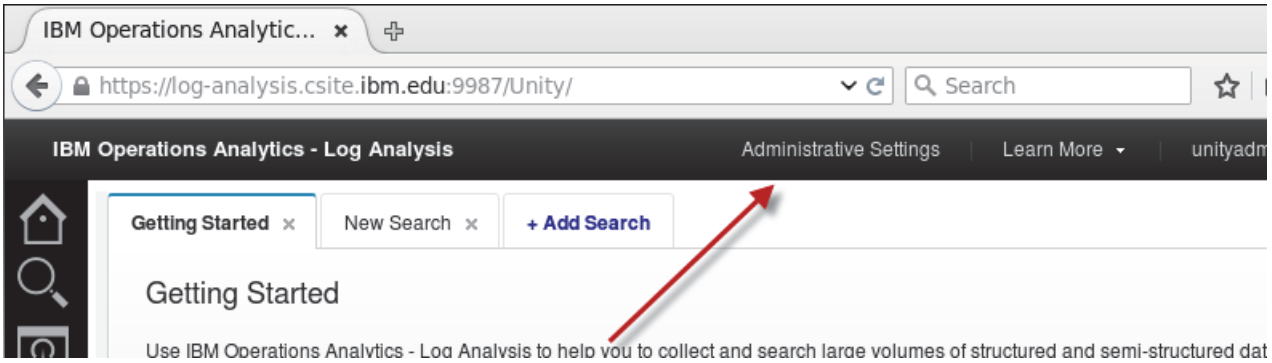
1. Go to the host named **log-analysis.csite.ibm.edu**.
2. Add a data source for the target log file in Log Analysis.
   a. Open a Firefox browser.



b. Enter the following address:
https://log-analysis.csite.ibm.edu:9987/Unity

c. Log in to the user interface with the user name **unityadmin** and the password **object00**.

d. Click **Administrative Settings**. The administration user interface opens in a new Firefox tab.

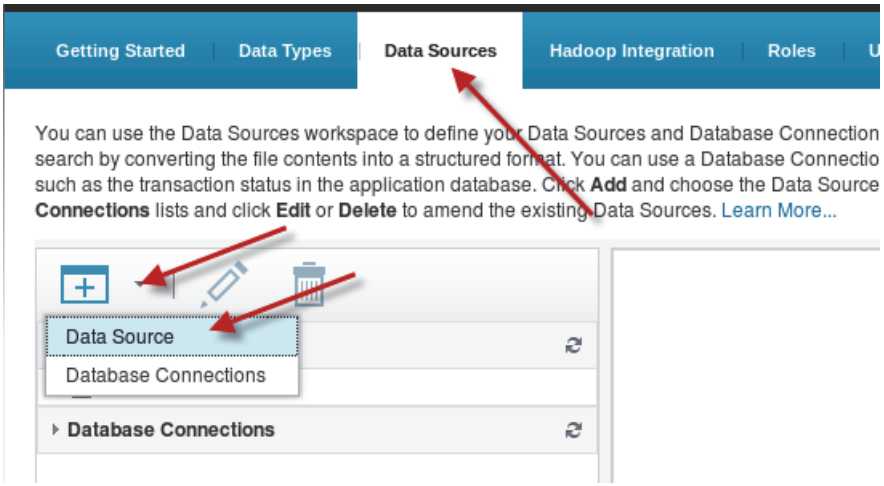


e. Create a data source named **Lab_DB2**. Use the values in the following table to complete the data source wizard.

Field	Value
Location	Select custom
Host name	TEST_db2diag
File Path	LabDB2Instance_db2diag
Type	DB2Diag
Collection	Leave this field blank
Name	Lab_DB2
Description	Leave this field blank
Group	Leave this field blank

f. Click the **Data Sources** tab in the administration user interface. The administration user interface is in the second Firefox tab.

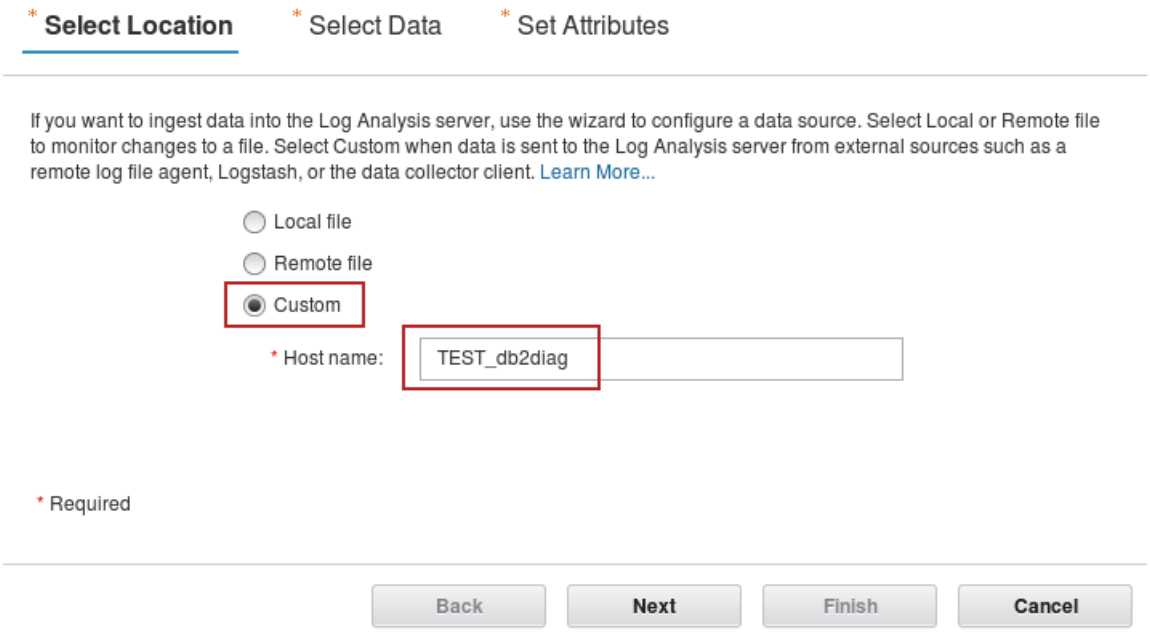
g. Click **Add > Data Source**.



h. Select **Custom**.

i. Enter **TEST_db2diag** as the host name.

j. Click **Next**.



Note: Notice that the host name is **TEST_db2diag**. This value corresponds to the value that you set with your mutate filter in the Logstash sender configuration and the metadata that you added with the LFA.

- From your sender Logstash configuration:

```
replace => ["host", "%{LFA_ENVIRONMENTNAME}_%{LFA_MODULE}"]
```

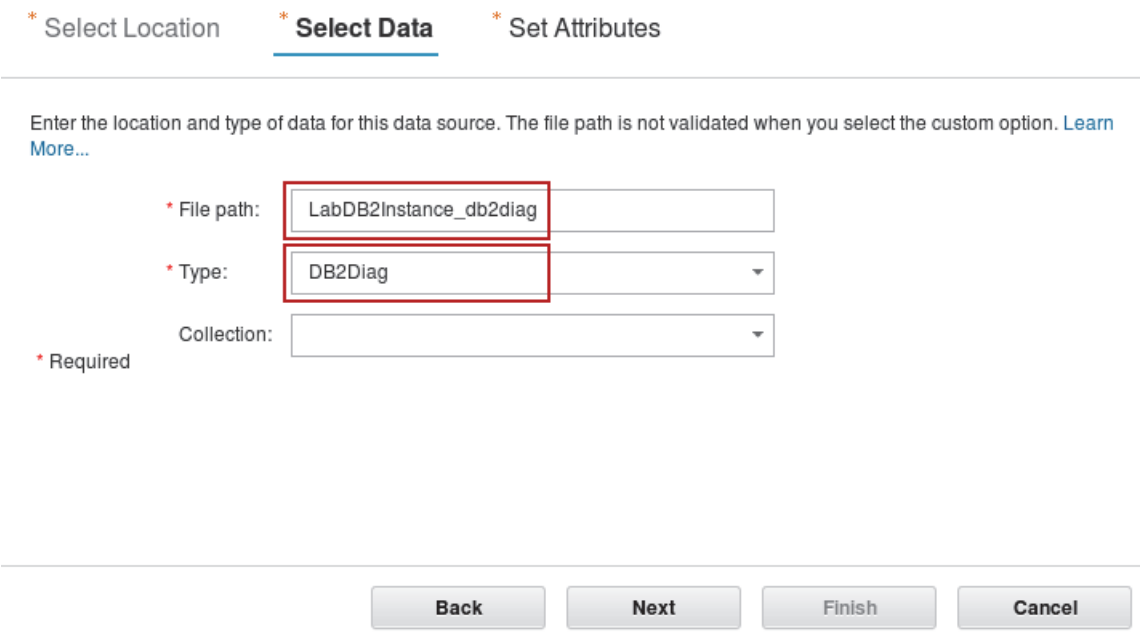
- From your `lab-db2diag.fmt` LFA configuration file:

```
...
type db2diag
instance LabDB2Instance
cluster NONE
module db2diag
env TEST
functional NONE
site NONE
text $1
END
```

k. Enter **LabDB2Instance_db2diag** as the file path.

l. Select **DB2Diag** as the type.

m. Click **Next**.



Note: Notice that the file path is **LabDB2Instance_db2diag**. This value corresponds to the value you set with your mutate filter in the Logstash sender configuration and the metadata you added with the LFA.

- From your sender Logstash configuration:

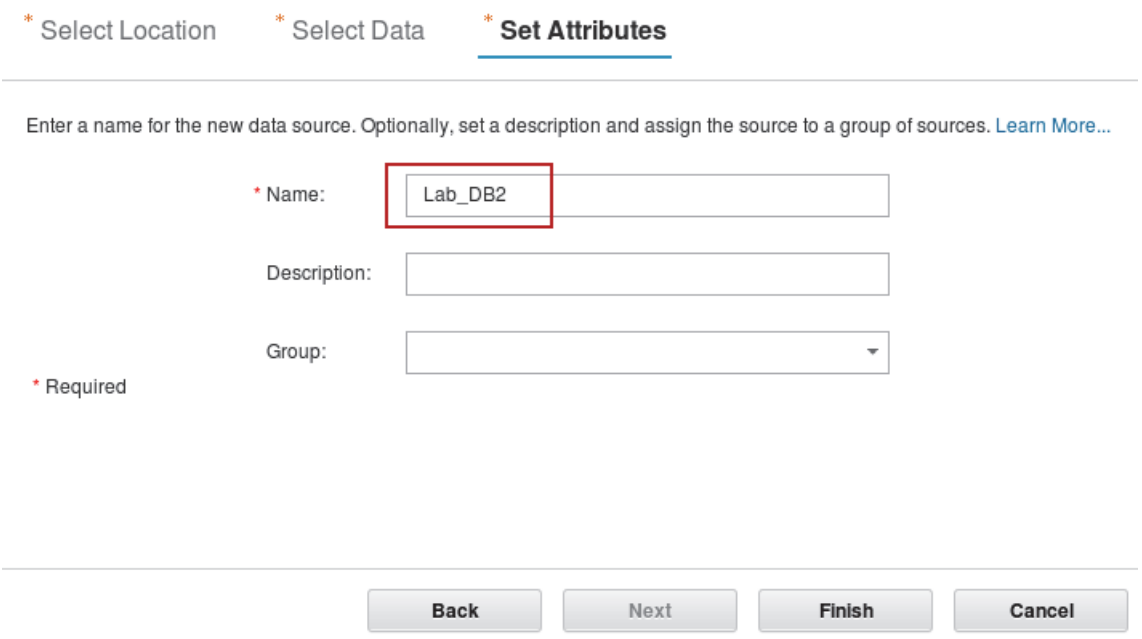
```
add_field => [ "path", "%{LFA_INSTANCE}_%{LFA_MODULE}" ]
```

- From your lab-db2diag.fmt LFA configuration file:

```
...
type db2diag
instance LabDB2Instance
cluster NONE
module db2diag
env TEST
functional NONE
site NONE
text $1
END
```

n. Enter **Lab_DB2** as the name of the data source.

o. Click **Finish**.



p. Click **OK** in the confirmation windows.

q. Leave this Firefox page open. You use it again in a moment.

The `GenericReceiver.log` file shows all data coming in to the Log Analysis server.

3. Run the following command to watch for activity in the `GenericReceiver.log` file.

   ```
   tail -f /opt/IBM/LogAnalysis/logs/GenericReceiver.log
   ```

4. Go to the host named **collection.csite.ibm.edu**.

5. Run the following command to add more messages to the target log file.

   ```
   /software/log_samples/scripts/DB2_Logs.sh
   ```

6. Return to the host named **log-analysis.csite.ibm.edu**.

7. Look at the `GenericReceiver.log` file.

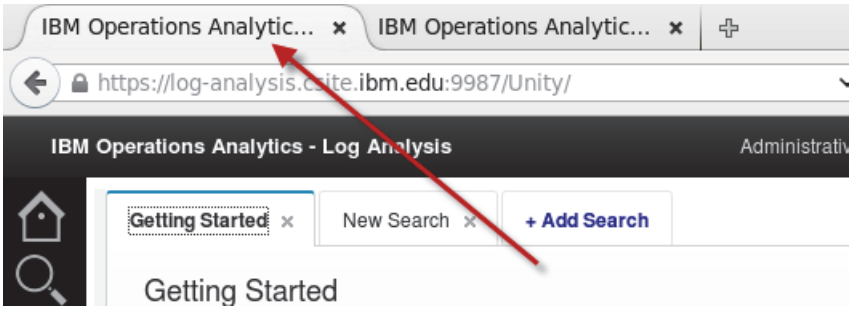
   a. Look for messages like the following example. Messages like these verify that data from the target log file is being processed by the Log Analysis software.

   ```
   06/06/16 19:32:51:543 UTC [Default Executor-thread-3386] INFO  -
   UnityFlowController :  Batch Status for -> Lab_DB2 , Size: 352 , Num
   successful: 352 , Num failures: 0 , Indexed Source volume: 79098

   06/06/16 19:32:51:543 UTC [Default Executor-thread-3386] INFO  -
   DataCollectorRestServlet : Batch of Size 352 processed and encountered 0
   failures
   ```

   b. Press Ctrl + C to stop the `tail` of the `GenericReceiver.log` file.

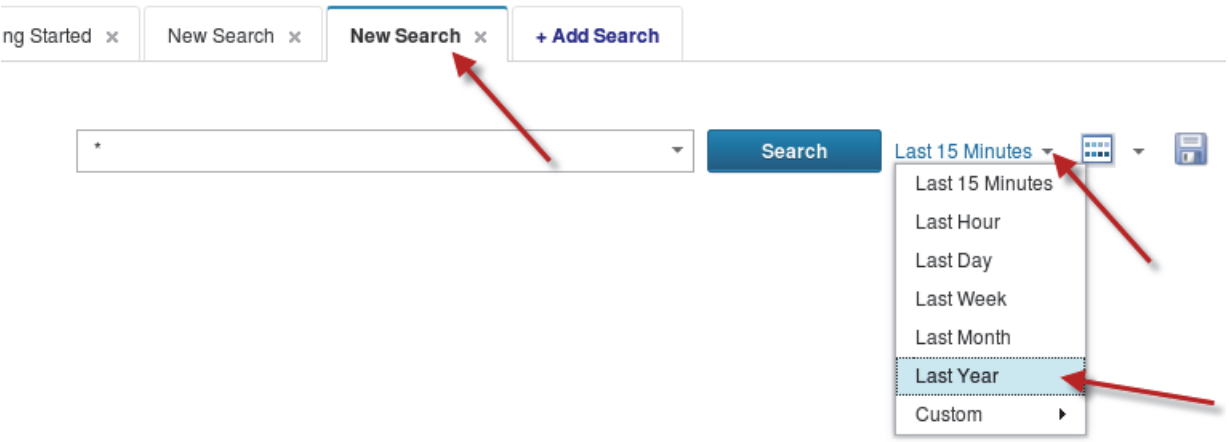
8. Verify that messages from the target log file are present in the Log Analysis search interface.

   a. Return to the Log Analysis user interface in the Firefox window. Go to the search interface by clicking the first Firefox tab.



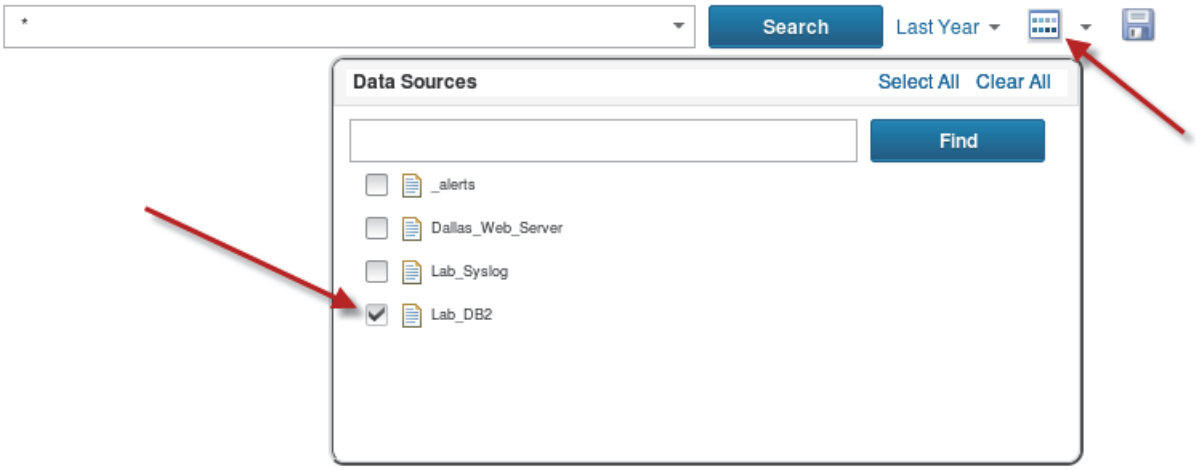
   b. Click the **Add Search** or the **New Search** tab.

c. Select **Last Year** as the time filter.

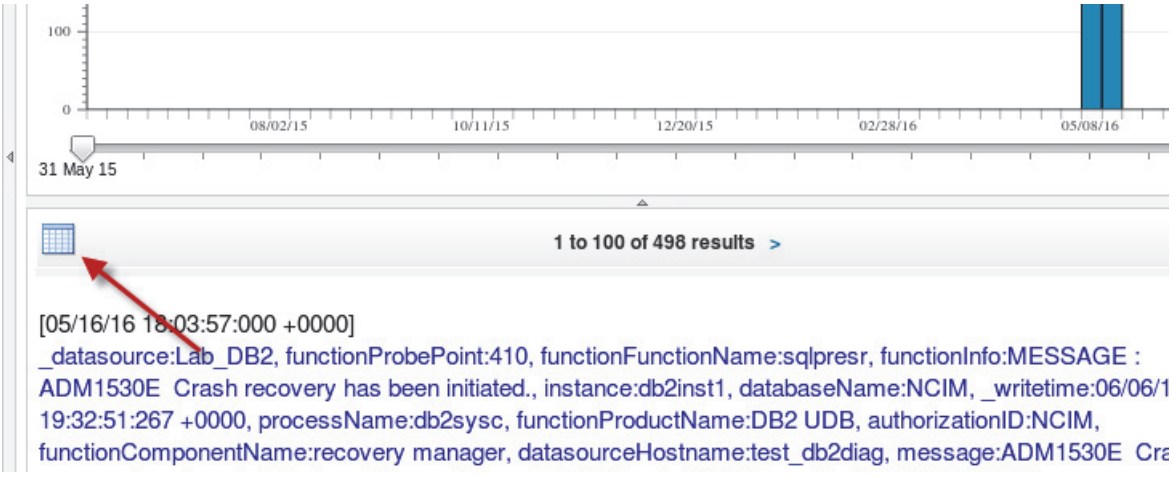


d. Select **Lab_DB2** as the only data source.

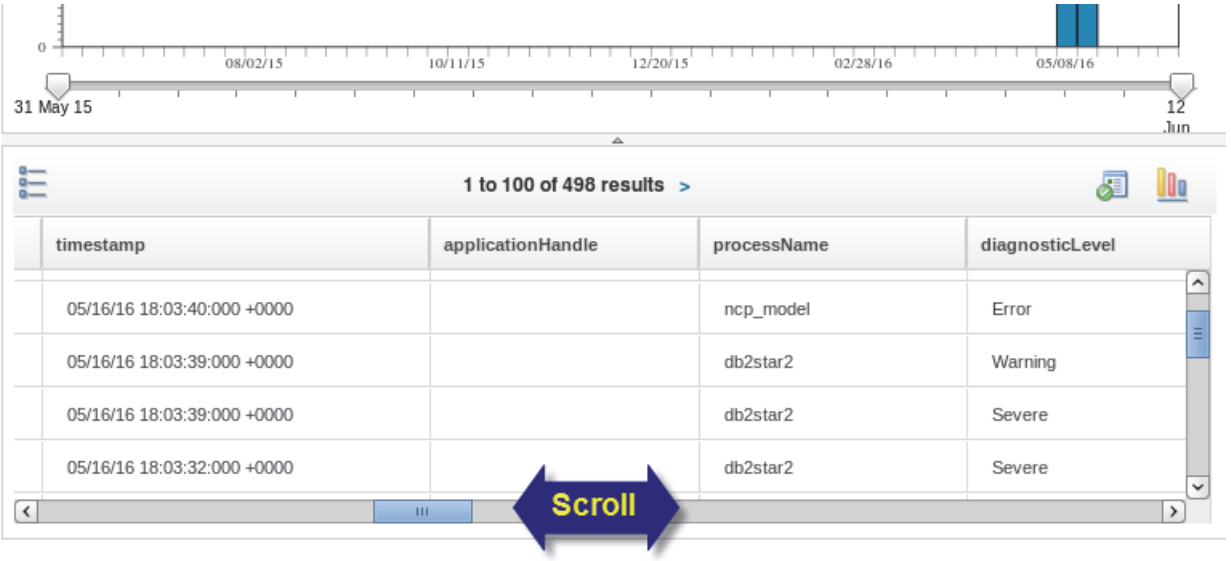
e. Click **Search**.



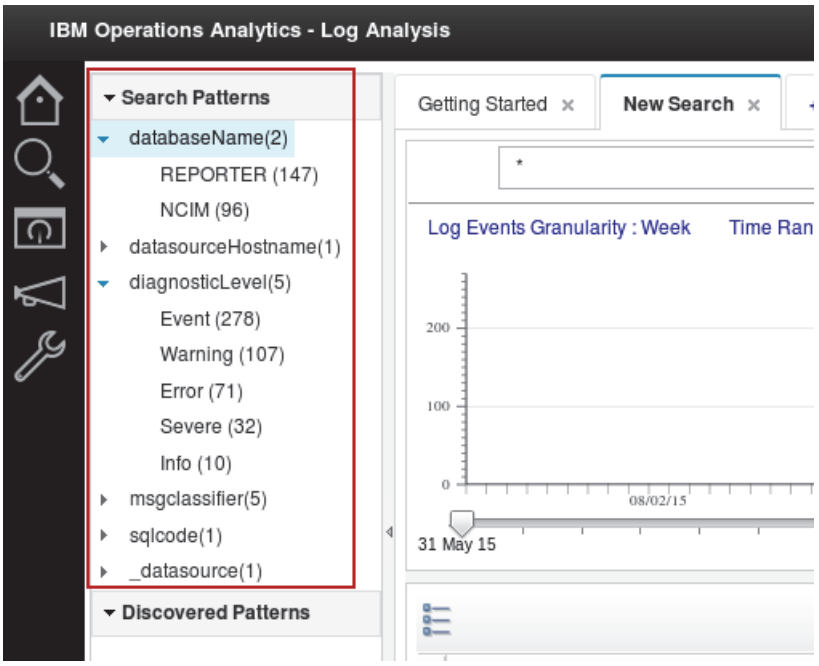
f. Log messages load in to the search interface. Click the **Grid View** button.



g. Scroll left and right to view the columns.



h. Look at the Search Patterns at the left of the search interface. Notice the facet counts and categories from the log file.



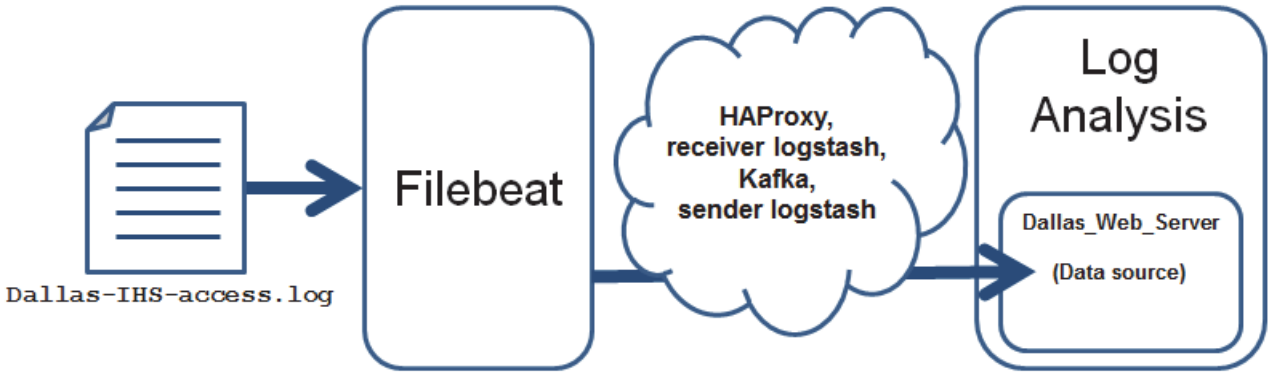
Unit 7 Log consolidation exercises

In a simple configuration, IBM Log Analysis manages input from a single target log file with a single data source. Environments with many log files to monitor (hundreds or thousands) require just as many data sources. It can be problematic for administrators to create, manage, navigate, and support so many data sources. In the exercises for this unit, you alter your lab environment to consolidate several web server logs into a single Log Analysis data source.

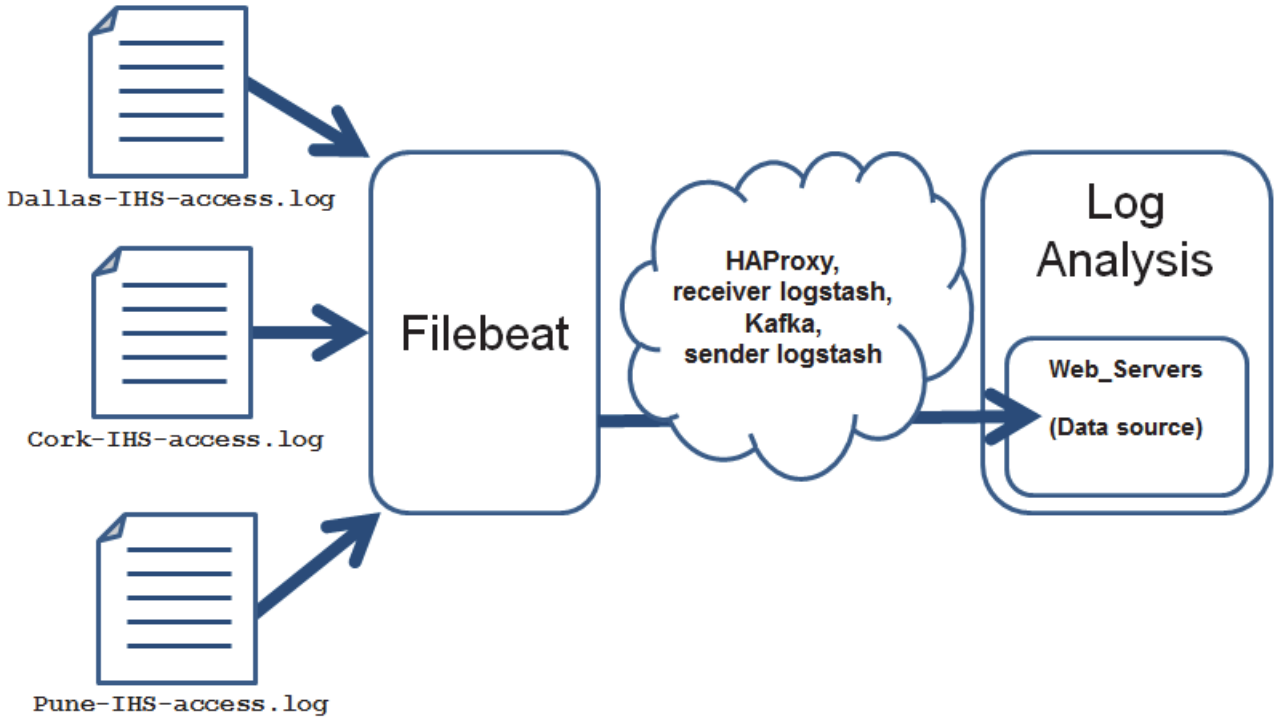
Earlier in this course, you configured your environment to monitor a web access log from an IBM HTTP Server (IHS). You configured Filebeat to monitor one physical log file named Dallas-IHS-access.log. You also configured a data source in Log Analysis named Dallas_Web_Server to index messages from this target log file.

In these exercises, you use meta data added by Filebeat to consolidate messages from three different web access log files into one Log Analysis data source. The following figures describe the changes that you make in this unit:

Your web server data flow before you make changes:



Your web server data flow after you make changes



Exercise 1 Deleting the current Log Analysis data source

Earlier in this course, you configured a data source in Log Analysis named Dallas_Web_Server to index messages from the Dallas-IHS-access.log target log file. In this exercise, you delete that data source.

Note: Run all of the steps in this exercise on the host named **log-analysis.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **log-analysis.csite.ibm.edu**.

Before you can delete a data source, you must delete all of the log data within the data source.

2. Delete all data in the Dallas_Web_Server data source.
   a. Run the following command to change to the Log Analysis delete utility directory.

```
cd /opt/IBM/LogAnalysis/utilities/deleteUtility
```

b. Edit the delete.properties file in a text editor.

```
vi delete.properties
```

c. Find the following line.

```
dataSourceName = SCALA_DATASOURCE
```

d. Change the name of the data source to Dallas_Web_Server.

```
dataSourceName = Dallas_Web_Server
```

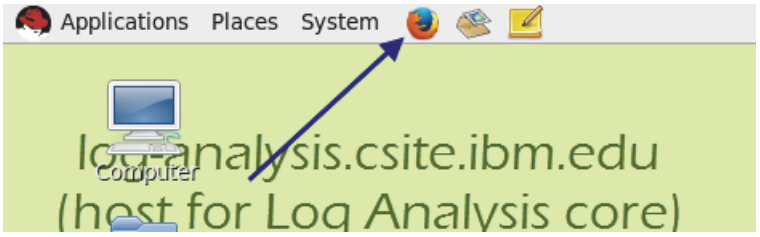
e. Save and close the file when you are finished.

f. Run the following command to delete the log data from the Dallas_Web_Server data source. The command runs for several minutes.

```
/usr/bin/python2.6 deleteUtility.py object00
```

3. Go to the Log Analysis administrator interface.

a. Open a Firefox browser.

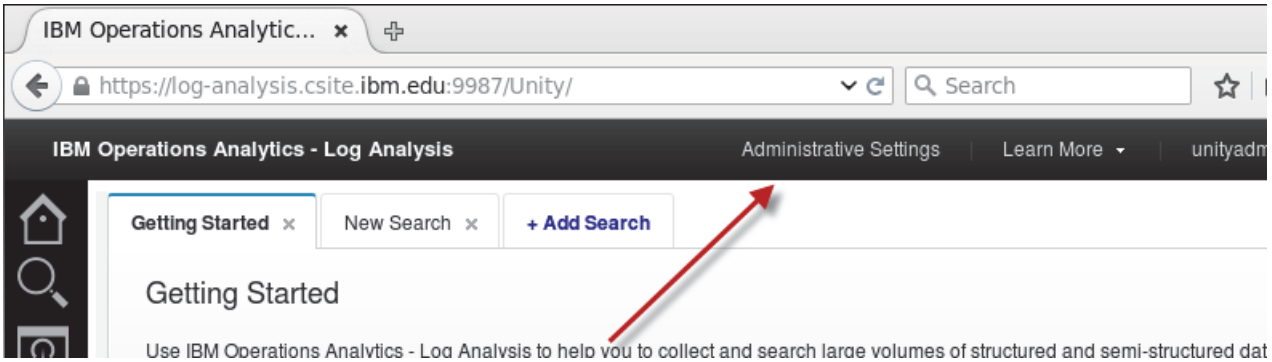


b. Enter the following address:

https://log-analysis.csite.ibm.edu:9987/Unity

c. Log in to the user interface with the user name **unityadmin** and the password **object00**.

d. Click **Administrative Settings**. The administration user interface opens in a new Firefox tab.

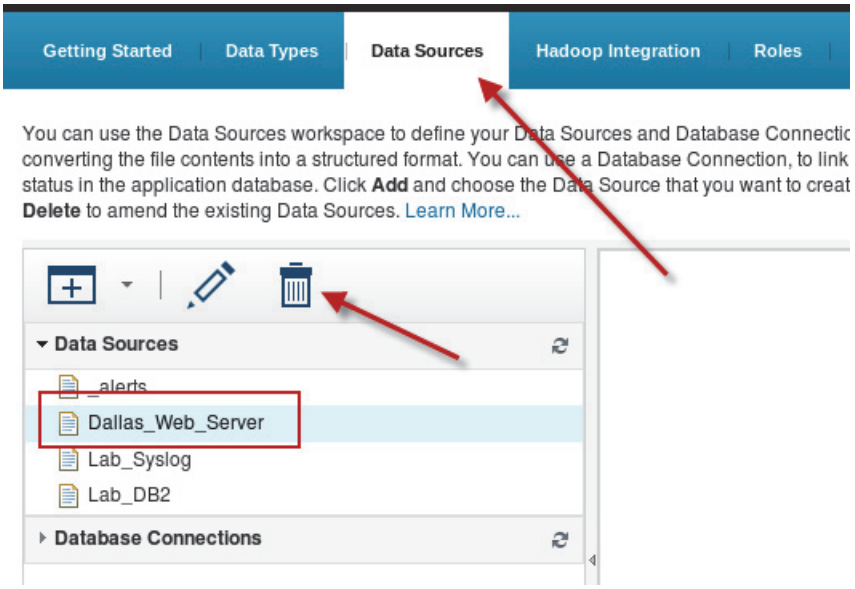


4. Delete the Dallas_Web_Server data source.

   a. Click the **Data Sources** tab.

   b. Select the **Dallas_Web_Server** data source.

   c. Click **Delete**.



   d. Click **OK** to confirm.

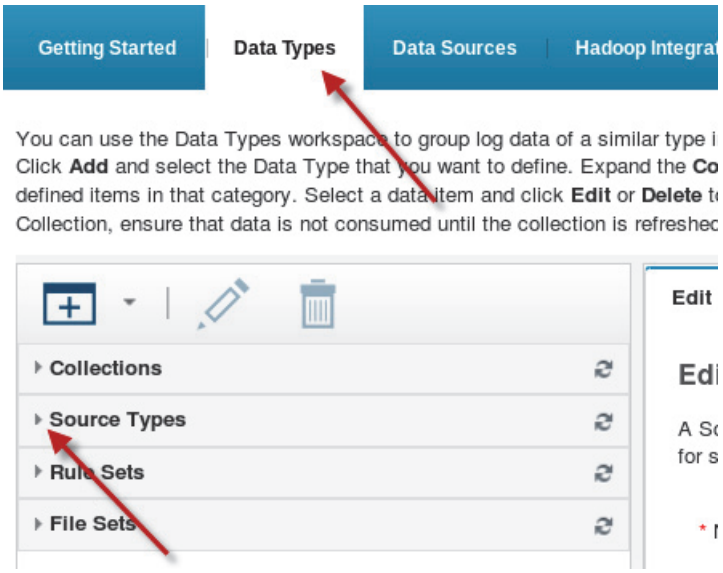
   e. Leave the Log Analysis administration user interface open. You use it in the next steps.

Exercise 2 Adding a source type and a new data source

In this exercise, you add a Log Analysis source type and an index configuration. You customize your source type to index three extra fields: **location**, **platform**, and **logfile**. You also create a new data source for web access logs that uses your new source type.

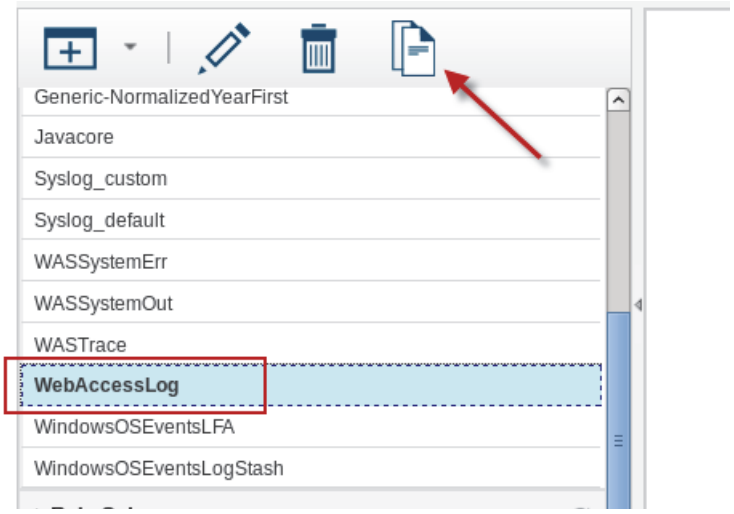
Important: Run all of the steps in this exercise on the host named **log-analysis.csite.ibm.edu** as the **netcool** user.

1. Make a copy of the source type named **WebAccessLog**.
   a. Click the **Data Types** tab in the Log Analysis administration user interface.
   b. Expand **Source Types**.

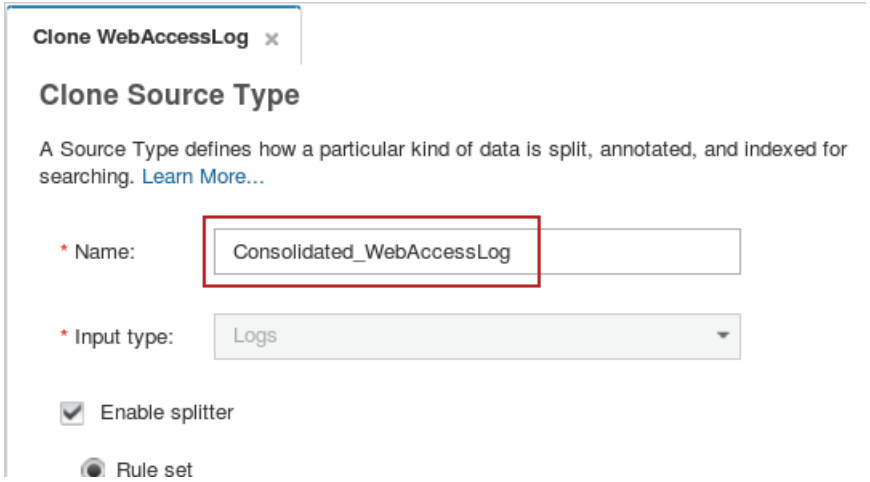


c. Click the **WebAccessLog** source type.

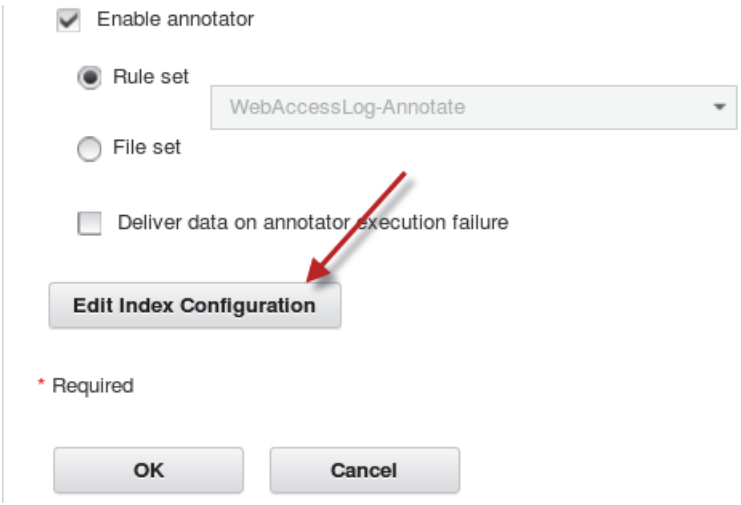
d. Click the clone button.



e. Enter **Consolidated_WebAccessLog** as the name.

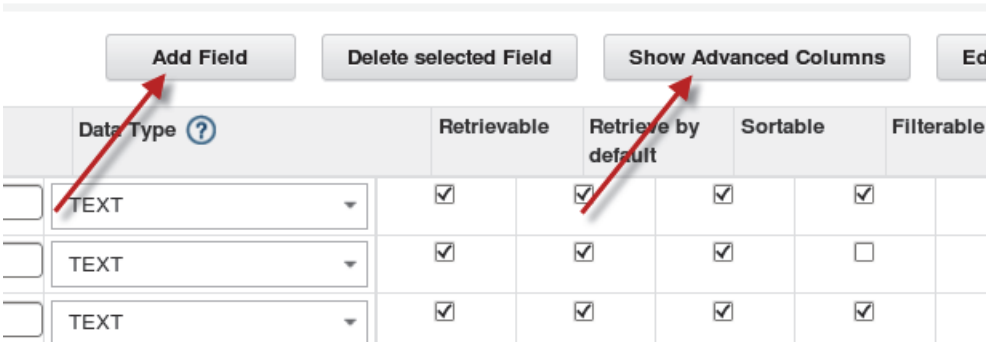


2. Change the index configuration for your data source to include three extra fields: **location**, **platform**, and **logfile**.
   a. Scroll down and click **Edit Index Configuration**.



   b. Click **Show Advanced Columns**.

   c. Add the first field. Click **Add Field**.



d. Enter **location** as the field name.

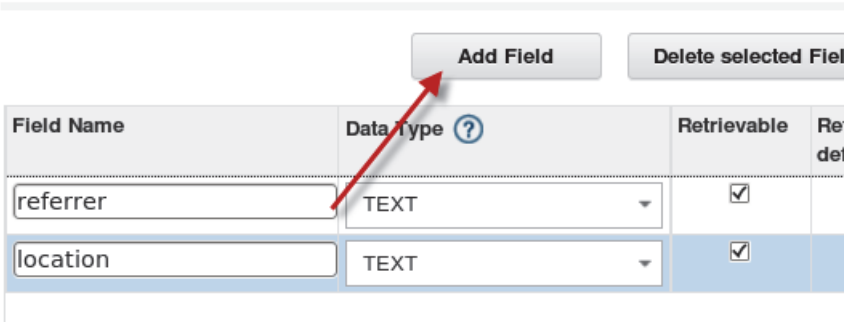
e. Select these options:

- **Retrievable**
- **Retrieve by default**
- **Sortable**
- **Filterable**
- **Searchable**

f. Enter **metadata.Location** as the path.

Hint: You might have to scroll from left to right as you customize your index configuration.

g. Add another field. Click **Add Field**.



h. Enter **platform** as the field name.

i. Select these options:

- **Retrievable**
- **Retrieve by default**
- **Sortable**
- **Filterable**
- **Searchable**

j. Enter **metadata.Platform** as the path.

Field Name	Data Type	Retrievable	Retrieve by default	Sortable	Filterable	Searchable	Combine	Paths
referrer	TEXT	☑	☑	☑	☐	☑	ALL	annotations.WebAcce
location	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Location
platform	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Platform

k. Add another field. Click **Add Field**.

l. Enter **logfile** as the field name.

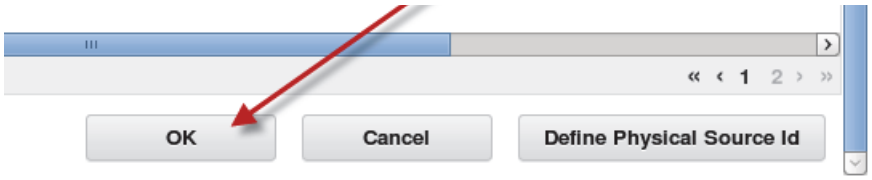
m. Select these options:

- **Retrievable**
- **Retrieve by default**
- **Sortable**
- **Filterable**
- **Searchable**

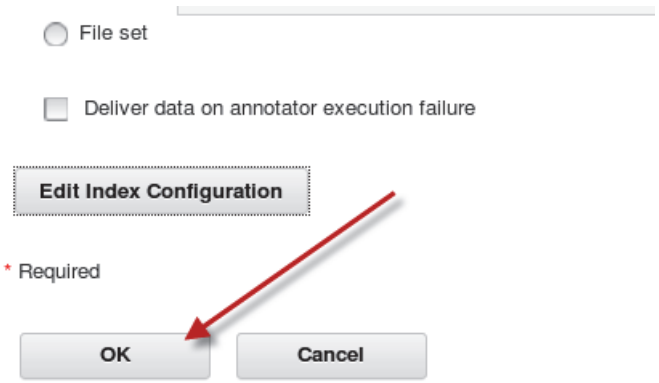
n. Enter **metadata.Logfile** as the path.

Field Name	Data Type	Retrievable	Retrieve by default	Sortable	Filterable	Searchable	Combine	Paths
referrer	TEXT	☑	☑	☑	☐	☑	ALL	annotations.WebAcc
location	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Location
platform	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Platform
logfile	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Logfile

o. Click **OK** at the bottom of the Edit Index Configuration pane.



p. Click **OK** to save the source type.



q. Click **OK** to confirm.

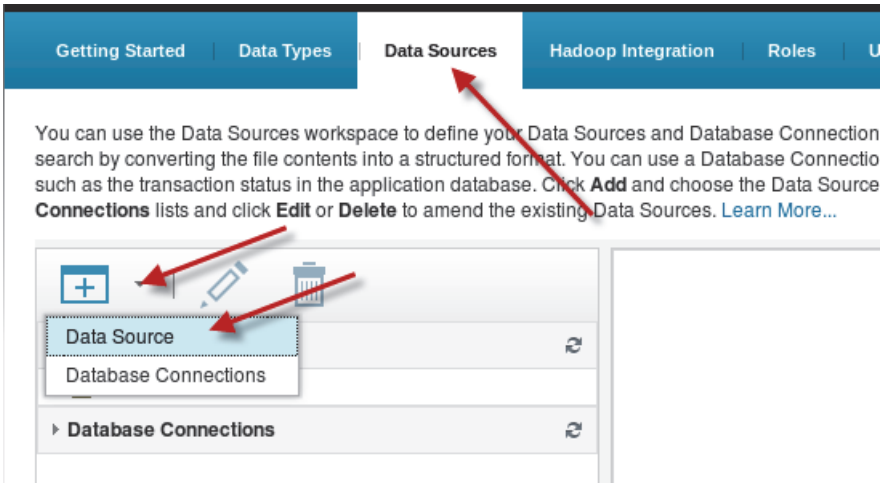
r. Leave the Log Analysis administration user interface open. You use it in the next steps.

3. Create a data source named **Web_Servers**. Use the values in the following table to complete the data source wizard.

Field	Value
Location	Select custom
Host name	DEV_IBM-HTTP-Server
File Path	access-log
Type	Consolidated_WebAccessLog
Collection	Leave this field blank
Name	Web_Servers
Description	Leave this field blank
Group	Leave this field blank

a. Click the **Data Sources** tab in the administration user interface.

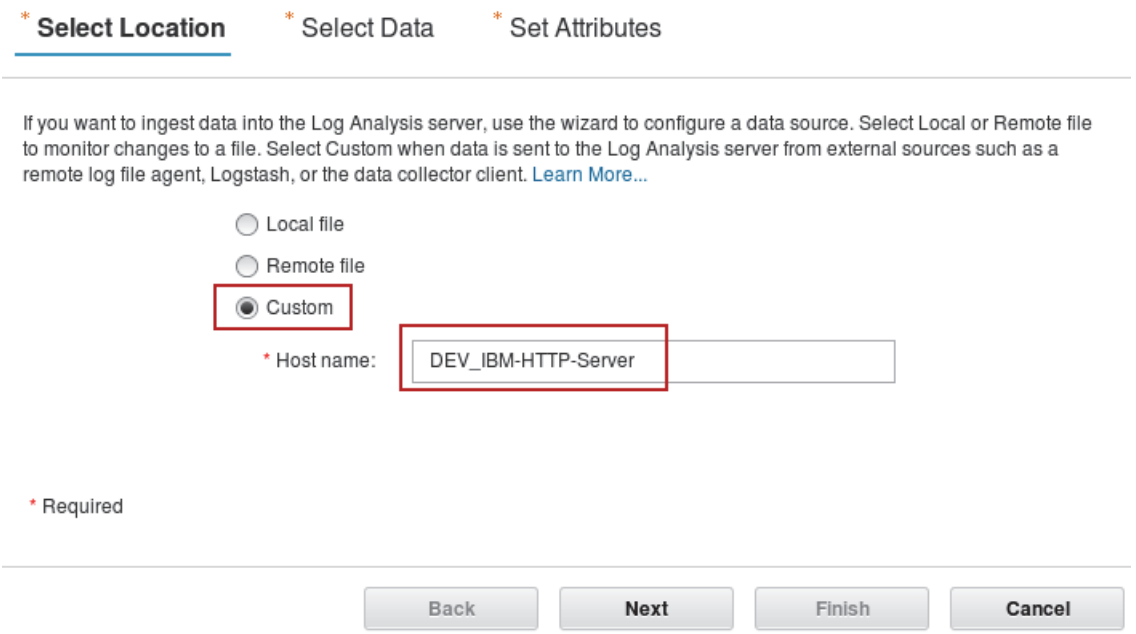
b. Click **Add > Data Source**.



c. Select **Custom**.

d. Enter **DEV_IBM-HTTP-Server** as the host name.

e. Click **Next**.



Note: Notice that the host name is **DEV_IBM-HTTP-Server**. This value corresponds to the value you set with your mutate filter in the Logstash sender configuration and the metadata you added with Filebeat.

- From your sender Logstash configuration:

```
replace => { "host" => "%{[fields][env]}_%{[fields][module]}" }
```

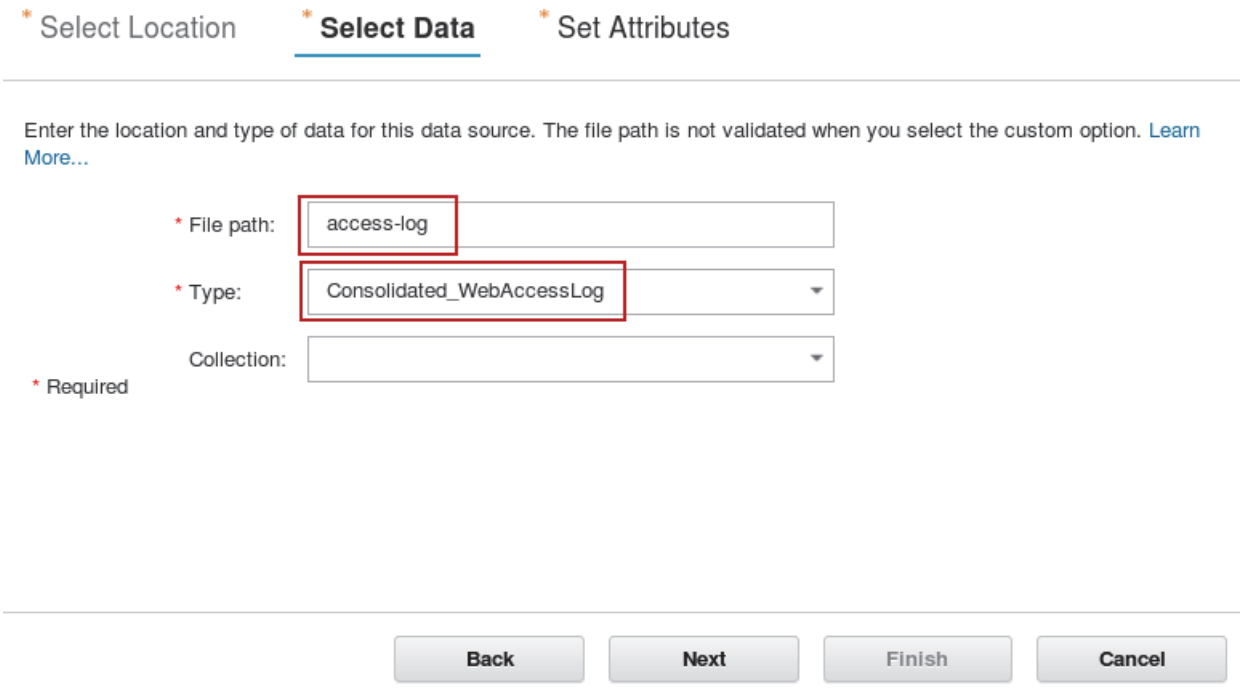
- From your Filebeat configuration:

```
fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: DALLAS
    platform: RHEL
```

f. Enter **access-log** as the file path.

g. Select **Consolidated_WebAccessLog** as the type. You created this source type in the preceding steps.

h. Click **Next**.



Note: Notice that the file path is **access-log**. This value corresponds to the value that you set with your mutate filter in the Logstash sender configuration and the metadata that you added with Filebeat.

- From your sender Logstash configuration:

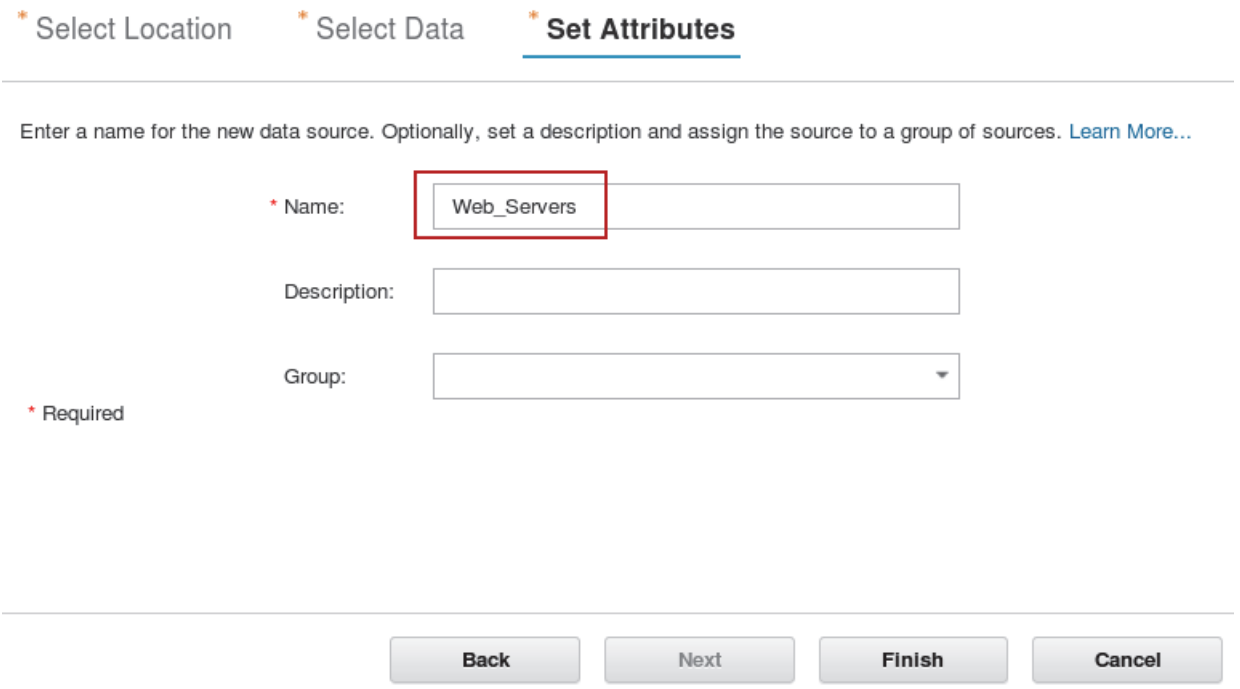
```
add_field => [ "path", "%{[fields][type]}" ]
```

- From your Filebeat configuration:

```
fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: DALLAS
    platform: RHEL
```

i. Enter **Web_Servers** as the name of the data source.

j. Click **Finish**.



k. Click **OK** in the confirmation windows.

Exercise 3 Configuring the Logstash sender

In the previous exercise, you configured Log Analysis to index the three extra metadata fields: location, platform, and logfile. In this exercise, you configure the Logstash sender server to create these extra fields and send them to the Log Analysis server.

Important: Run all of the steps in this exercise on the host named **s-logstash.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **s-logstash.csite.ibm.edu**.

2. Configure the logstashW server to create three fields, **Location**, **Platform**, and **Logfile** from the metadata that was added by Filebeat.

   a. Open the logstashW configuration file in a text editor.

   ```
   vi /opt/logstashW/logstash-2.2.1/conf/logstashW.conf
   ```

b. Add the following lines in bold typeface to the filter section.

```
} # end filebeat mutate condition

if [fields][type] == "access-log" {
    mutate {
      add_field => [ "Location", "%{[fields][site]}" ]
      add_field => [ "Platform", "%{[fields][platform]}" ]
      add_field => [ "Logfile", "%{resourceID}"]
    } # end web server mutate
} # end web server mutate condition

if "grok_lfa" in [tags] {
```

c. Keep the file open. You add more lines to this file in the next steps.

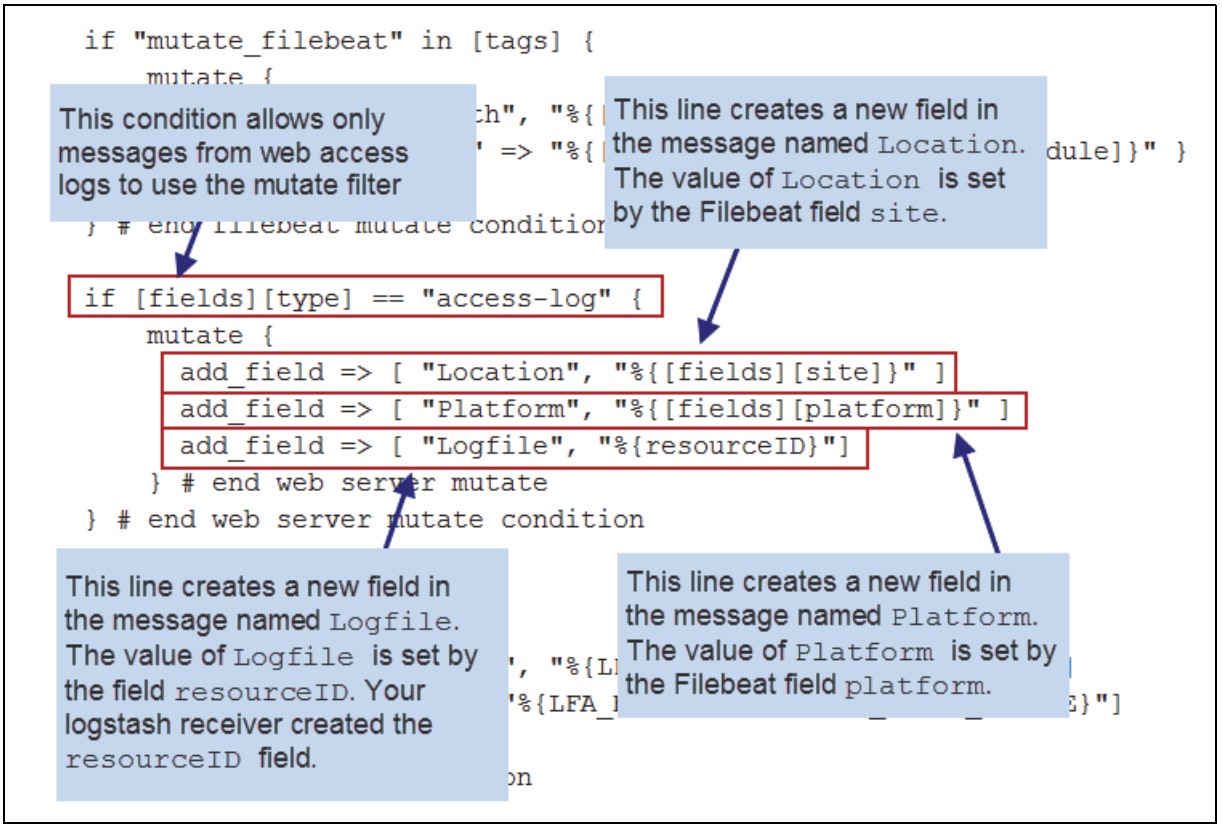


Figure 1 Key fields in logstashW.conf

Note: Remember, you added extra metadata fields **site** and **platform** with your Filebeat configuration earlier in this course.

From your Filebeat configuration:

```
paths:
 - /software/log_samples/IHS_logs/Dallas-IHS-access.log
input_type: log
fields:
 collector: filebeats-collection.csite.ibm.edu
 env: DEV
 module: IBM-HTTP-Server
 type: access-log
 site: DALLAS
 platform: RHEL
```

3. Configure the logstashW server to send the extra metadata fields to Log Analysis.
   a. Add the following lines in bold typeface to the output section.

```
        log_file => "/opt/logstashW/logstash-2.2.1/log/scala_logstashW.log"
        log_level => "info"
        metadata_fields => {
          "DEV_IBM-HTTP-Server@access-log" => {
            "field_names" => "Location,Platform,Logfile"
            "field_paths" => "Location,Platform,Logfile"
          } # end web server meta data
        } # end meta data fields
    }#end scala output

}# end output section
```

   b. Save and close the file when you are finished.

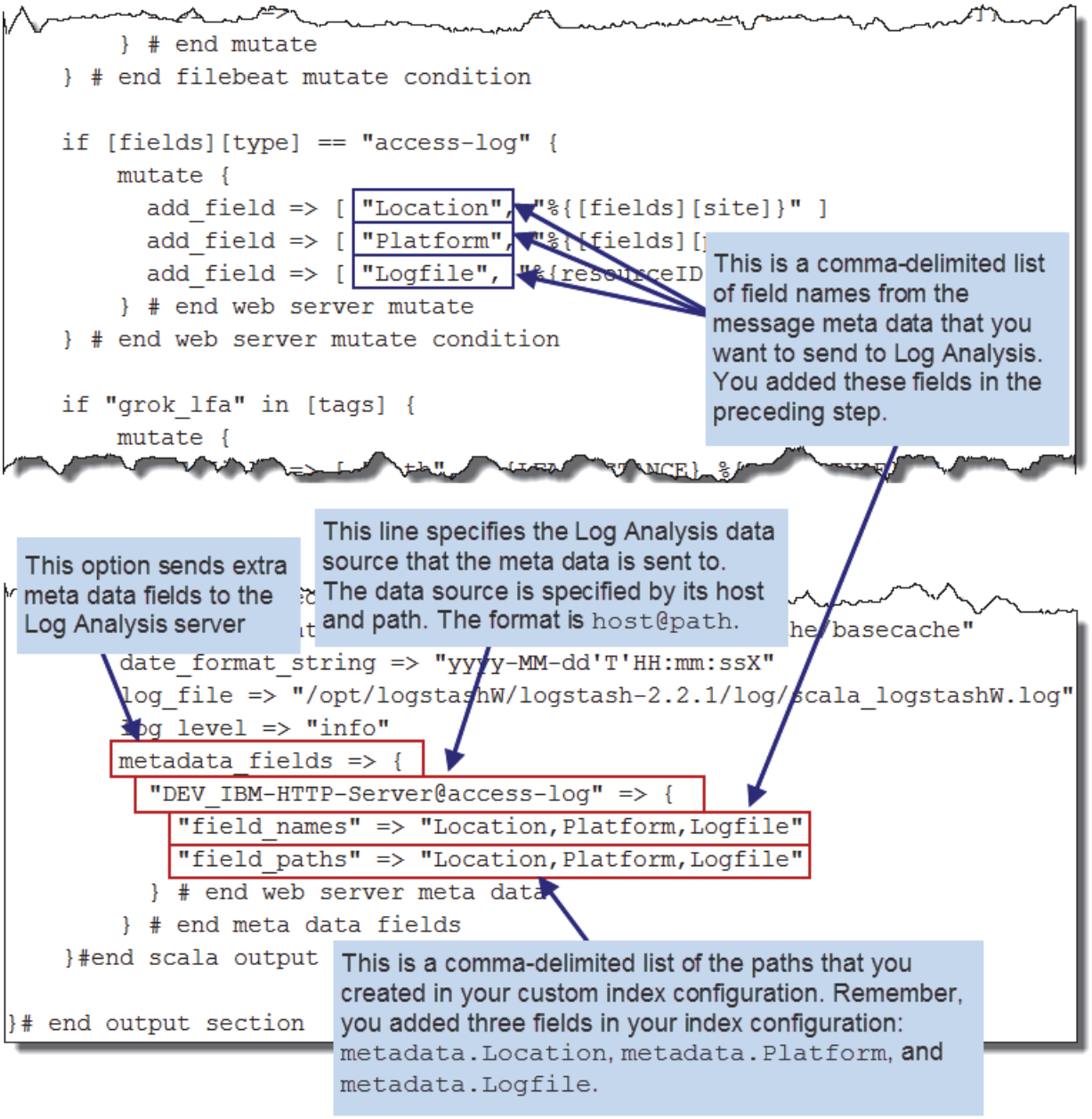


Figure 2 Key fields in logstashW.conf

Note: Remember, you created the paths when you added the three extra metadata fields in your custom index configuration:

Field Name	Data Type	Retrievable	Retrieve by default	Sortable	Filterable	Searchable	Combine	Paths
referrer	TEXT	☑	☑	☑	☐	☑	ALL	annotations.WebAcc
location	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Location
platform	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Platform
logfile	TEXT	☑	☑	☑	☑	☑	ALL	metadata.Logfile

4. Restart the logstashW server.
   a. Run the following command to stop the logstashW server.

      ```
      pkill -f logstashW
      ```

   b. Run the following command to start the logstashW server.

      ```
      startlogstashW
      ```

Exercise 4 Configuring Filebeat

In this exercise, you configure Filebeat to monitor two additional web access logs from an IBM HTTP Server (IHS).

Important: Run all of the steps in this exercise on the host named **collection.csite.ibm.edu** as the **netcool** user.

1. Go to the host named **collection.csite.ibm.edu**.
2. Change your Filebeat configuration file to monitor two additional web access logs.
   a. Open your Filebeat configuration file in a text editor.

      ```
      vi /opt/filebeat-1.1.1-x86_64/filebeat.yml
      ```

b. Add the following lines in bold typeface.

```
    site: DALLAS
    platform: RHEL
  -
   paths:
    - /software/log_samples/IHS_logs/Pune-IHS-access.log
   input_type: log
   fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: PUNE
    platform: SLES
  -
   paths:
    - /software/log_samples/IHS_logs/Cork-IHS-access.log
   input_type: log
   fields:
    collector: filebeats-collection.csite.ibm.edu
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: CORK
    platform: RHEL

output:
 logstash:
  hosts: ["ha-proxy.csite.ibm.edu:20737"]
```

c. Save and close the file when you are finished.

Important: Pay attention to the indentations in the `filebeat.yml` file. This file is indented with spaces, not tabs.

```
    env: DEV
    module: IBM-HTTP-Server
    type: access-log
    site: DALLAS
    platform: RHEL
  -
  paths:
   - /software/log_samples/IHS_logs/Pune-IHS-access.log
  input_type: log
  fields:
   collector: filebeats-collection.csite.ibm.edu
   env: DEV
   module: IBM-HTTP-Server
   type: access-log
   site: PUNE
   platform: SLES
  -
  paths:
   - /software/log_samples/IHS_logs/Cork-IHS-access.log
  input_type: log
  fields:
   collector: filebeats-collection.csite.ibm.edu
   env: DEV
   module: IBM-HTTP-Server
   type: access-log
   site: CORK
   platform: RHEL

output:
```

This is the path to the target log file

These are meta data fields that Filebeat creates and adds to each log message

This is the path to another target log file

These are meta data fields that Filebeat creates and adds to each log message

Figure 3 Key fields in filebeat.yml

3. Restart Filebeat so that it uses the new configuration.
   a. Run the following command to stop Filebeat.

      ```
      pkill -f filebeat
      ```

   b. Run the following command to start Filebeat.

      ```
      /opt/filebeat-1.1.1-x86_64/filebeat &
      ```

   c. Run the following command to verify that Filebeat is running.

      ```
      ps -ef | grep -i filebeat

      netcool   2796  2415  0 19:54 pts/0    00:00:00
      /opt/filebeat-1.1.1-x86_64/filebeat
      ```

Hint: If your Filebeat collection agent cannot start, check the indentation of the lines that you added and try to start Filebeat again.

4. Add more messages to each of the three web server log files.
   a. Run the following command to add more messages to the Dallas web server log file.
      `/software/log_samples/scripts/Dallas_Web_Logs.sh`
   b. Run the following command to add more messages to the Pune web server log file.
      `/software/log_samples/scripts/Pune_Web_Logs.sh`
   c. Run the following command to add more messages to the Cork web server log file.
      `/software/log_samples/scripts/Cork_Web_Logs.sh`

Exercise 5 Verifying log consolidation

In this exercise, you verify that messages from three different log files are consolidated into one Log Analysis data source.

Important: You use two different hosts in this exercise. Pay careful attention to which host you are working on when you complete each step.

1. Go to the host named **s-logstash.csite.ibm.edu**.
2. Run the following command to view the most recent messages in the logstashW debug log file.
   ```
   tail -50 /opt/logstashW/logstash-2.2.1/log/logstashW-debug.log
   ```

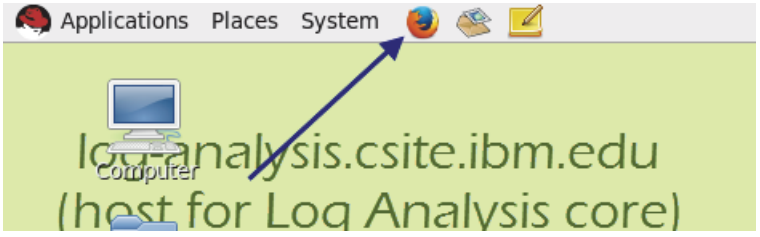
3. Look for messages like the following example. Verify that the logstashW server created the **Location**, **Platform**, and **Logfile** fields.

```
...
         "kafka" => {
              "msg_size" => 886,
                 "topic" => "DEV_IBM-HTTP-Server_access-log",
        "consumer_group" => "G-DEV_IBM-HTTP-Server_access-log",
             "partition" => 0,
                  "key" => byte[99, 111, 108, 108, 101, 99, 116, 105, 111, 110,
46, 99, 115, 105, 116, 101, 46, 105, 98, 109, 46, 101, 100, 117, 95, 47, 115,
111, 102, 116, 119, 97, 114, 101, 47, 108, 111, 103, 95, 115, 97, 109, 112, 108,
101, 115, 47, 73, 72, 83, 95, 108, 111, 103, 115, 47, 67, 111, 114, 107, 45, 73,
72, 83, 45, 97, 99, 99, 101, 115, 115, 46, 108, 111, 103, 95, 49]@60808bbb
    },
          "path" => "access-log",
      "Location" => "CORK",
      "Platform" => "RHEL",
       "Logfile" =>
"collection.csite.ibm.edu_/software/log_samples/IHS_logs/Cork-IHS-access.log_1"
```

4. Go to the host named **log-analysis.csite.ibm.edu**.

5. Verify that messages from the three target log files are present in the Log Analysis Web_Servers data source.

   a. Open a Firefox browser.



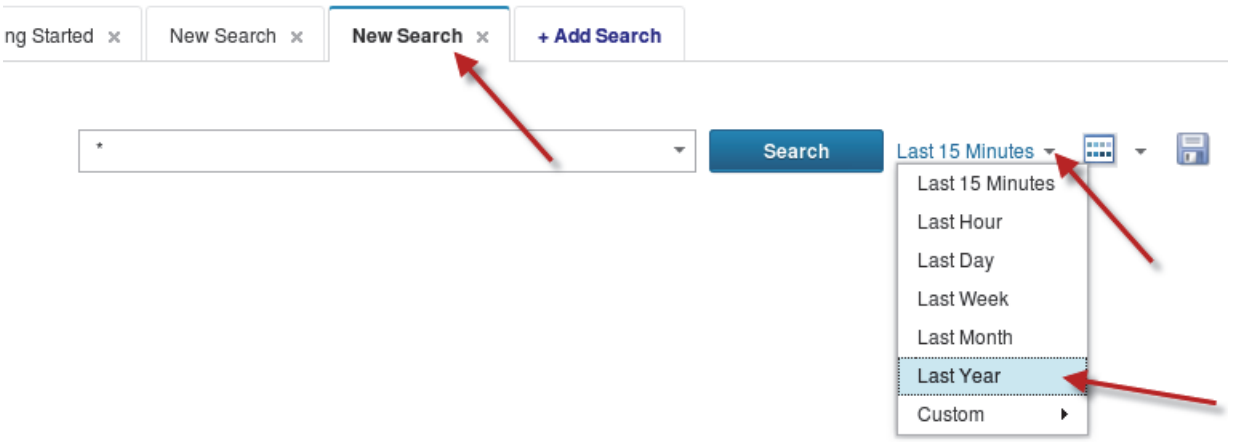
   b. Enter the following address:

      https://log-analysis.csite.ibm.edu:9987/Unity

c. Log in to the user interface with the user name **unityadmin** and the password **object00**.

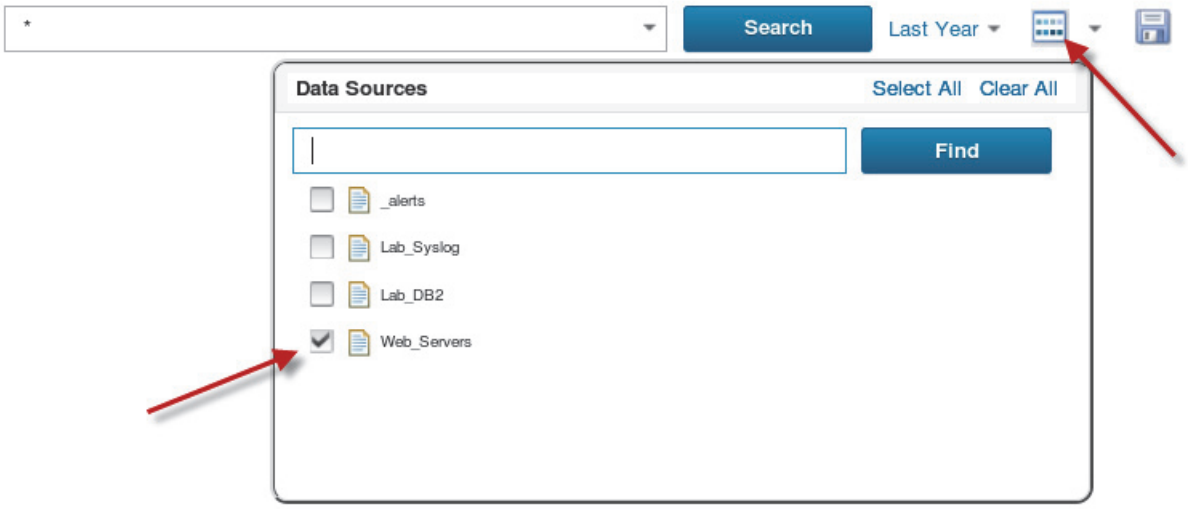
d. Click the **Add Search** or the **New Search** tab.

e. Select **Last Year** as the time filter.

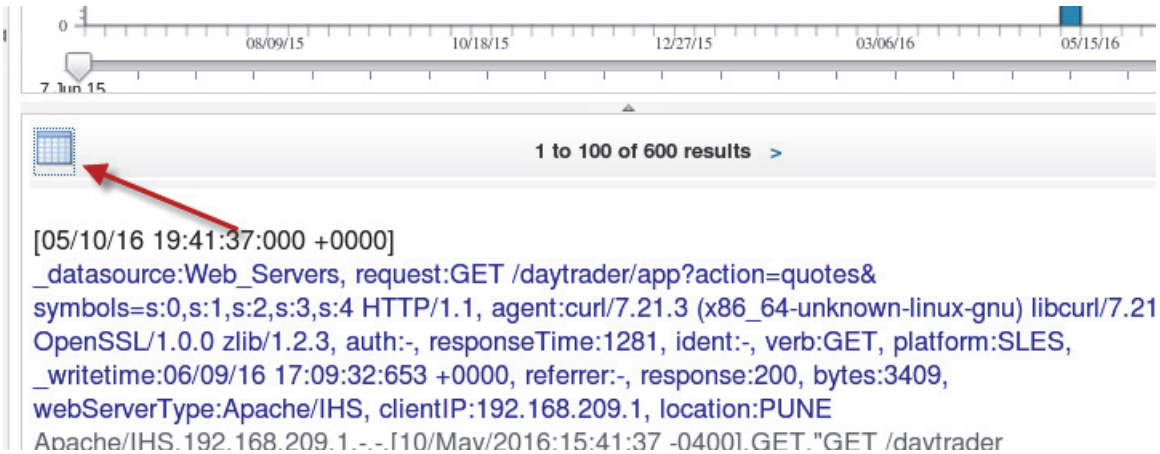


f. Select **Web_Servers** as the only data source.

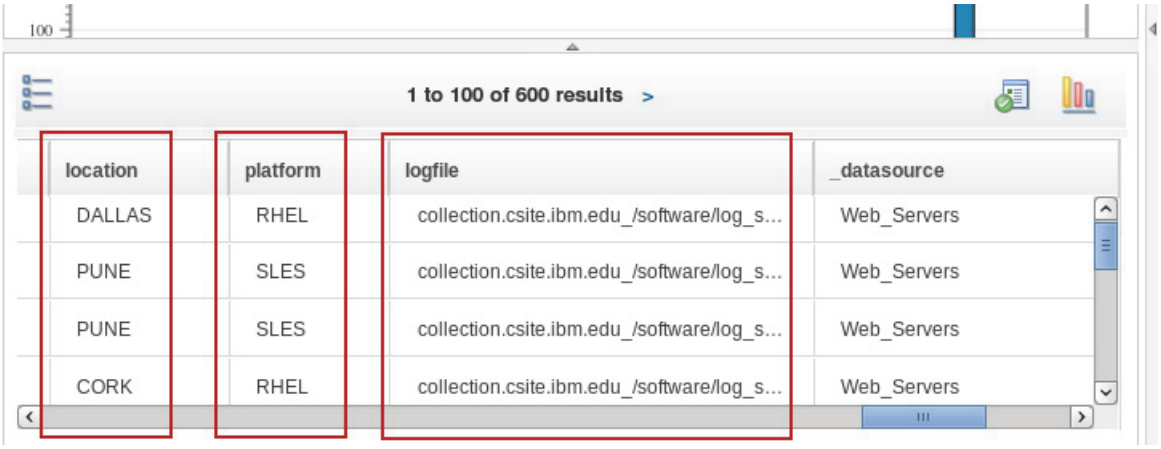
g. Click **Search**.



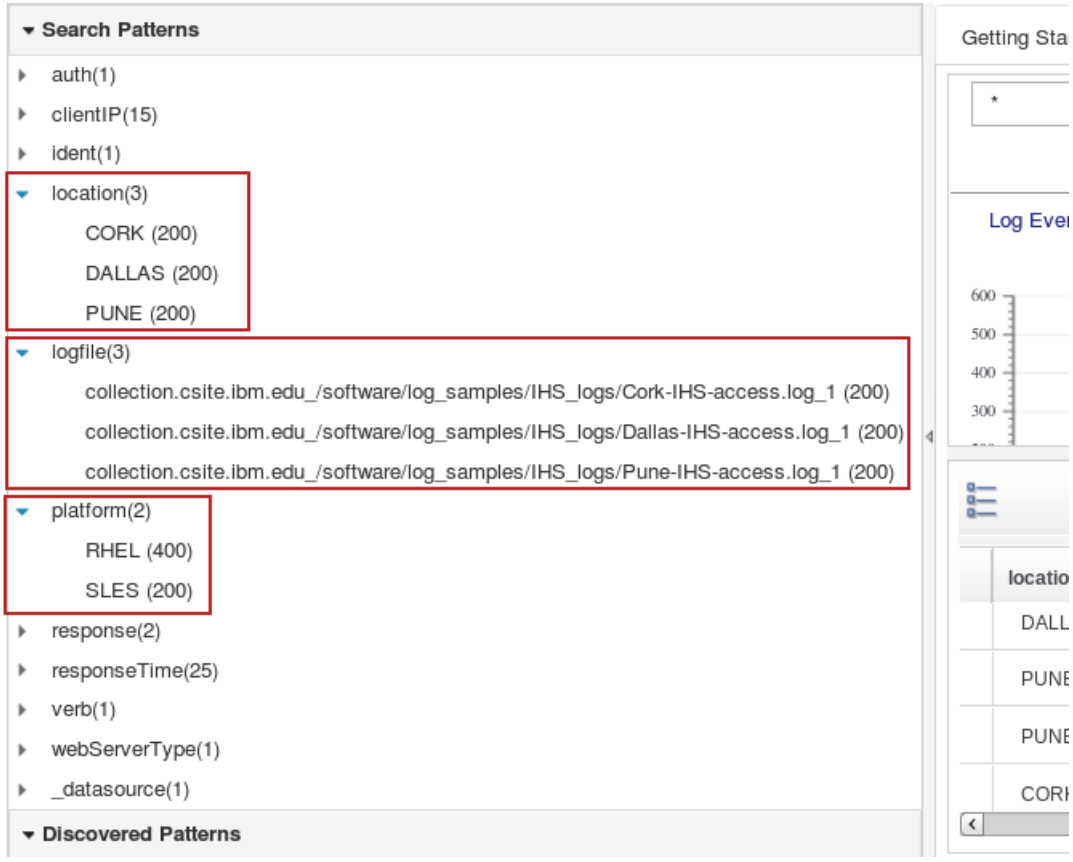
h. Log messages load in to the search interface. Click the **Grid View** button.



i. Scroll right and verify that the **location**, **platform,** and **logfile** columns are present and they have data in them.



j. Look at the Search Patterns at the left of the search interface. Notice the facet counts for **location**, **platform**, and **logfile**.



Log Analysis users can drill down to messages in each target log file by clicking the location, platform, and logfile values, such as PUNE or RHEL.

Important: These exercises illustrate the difference between physical data sources and logical data sources:

- Physical data sources are the actual target log files, such as Dallas-IHS-access.log, Pune-IHS-access.log, or Cork-IHS-access.log.
- Logical data sources are administrative objects that you create with the Log Analysis administrator interface, such as Web_Servers.

Unit 8 Troubleshooting exercises

This unit has no student exercises.

IBM Training