



IBM Training

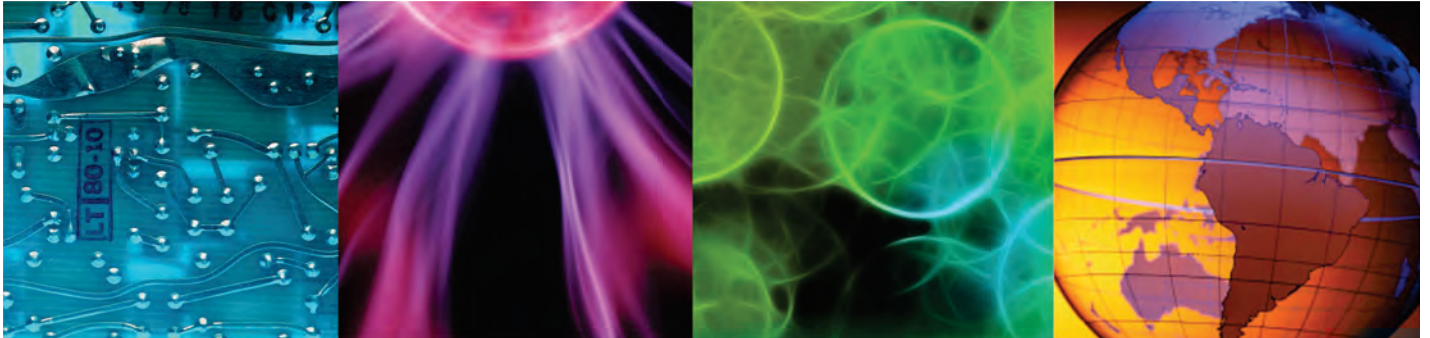
IBM Operations Analytics Log Analysis

1.3 Administration

Course Guide

Course code TN630G ERC 1.0

April 2016



All files and material for this course are IBM copyright property covered by the following copyright notice.

© Copyright IBM Corp. 2016. All Rights Reserved.

US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.



Contents

About this course	vii
About the student	viii
Learning objectives	viii
Course agenda	ix
1 Overview and basic administration	1
Objectives	2
Lesson 1 Overview	3
IBM Operations Analytics Log Analysis defined	4
User interface: Searching through logs	5
User interface: Creating charts and dashboards	7
User interface: Dashboard auto-refresh	8
User interface: Compare messages from dissimilar log sources	10
User interface: Create dashboards from dissimilar log sources	11
User interface: Expert advice	12
Insight Packs provide content	13
Lesson 2 Basic administration tasks	14
Starting and stopping the application	15
Adding users: File-based authentication	16
Encoding passwords	17
Roles	19
Data source definition	21
Adding a data source	22
Data source groups	24
Insight Pack dashboard configuration	26
User searches and chart creation work flow	28
Create new search and plot chart	29
Drill down from chart	30
Interactive dashboards	31
Viewing product usage	33
Server Statistics page	34
The export_statistics tool	35
Deleting data	37
How to delete data	38
Summary	40
Student exercises	41
2 Common configuration tasks	42
Objectives	43

Lesson 1 Generic Annotation Insight Pack	44
Generic Annotation Insight Pack overview	45
How to use the Generic Annotation Insight Pack	46
Creating the index configuration	47
Creating a source type	51
Creating a collection	56
Creating a data source	57
Discovered patterns	58
Multiple word keys	59
Stop words	61
Lesson 2 Delimiter-separated value toolkit	62
Overview of the delimiter-separated value (DSV) toolkit	63
Log file requirements	64
How to use the DSV toolkit	65
Creating a properties file	66
Editing the properties file	67
Creating and deploying the Insight Pack	69
Excluding and combining fields	70
Summary	71
Student exercises	72
3 Troubleshooting	73
Objectives	74
Log Analysis components	75
Log Analysis data ingestion components	76
Log configuration	77
Example troubleshooting workflow	78
Common problems and resolution, 1 of 2	79
Common problems and resolution, 2 of 2	82
Solr troubleshooting	84
Architecture	85
Interfaces and content	86
Log Analysis architecture	87
Ingestion pipeline	88
Example deployment architecture	89
Summary	91
4 Alerts	92
Objectives	93
Lesson 1 Overview	94
Alerts overview	95
Data flow	97
Condition and action templates	98
Lesson 2 Included alert actions	99
Index alert action	100
Email alert action	102
Log alert action	103
Script alert action	104
Working with alert actions	106

Lesson 3 Included alert conditions	107
Search query base condition	108
Working with base conditions	110
Single condition count composite condition	111
Single condition deduplication composite condition	112
Multiple base condition	113
Working with composite conditions	114
Summary	115
Student exercises	116
5 Hadoop Distributed File System (HDFS) integration	117
Objectives	118
Integration overview	119
Log Analysis data tiers	120
Data flow	122
User search experience	123
Typical data housekeeping	124
Integration prerequisites	125
Configuring BigInsights and Hadoop	126
Configuring Log Analysis	128
Verifying the integration	130
Disabling the integration	134
Summary	135
Student exercises	136
6 Performance tuning	137
Objectives	138
Tuning the server and operating system	139
Reducing the TIME_WAIT parameter for socket connections	141
Tuning Java virtual machine (JVM) options	142
Tuning the indexing engine (Solr)	143
Log File Agent (LFA) tuning	144
Configuring EIF receiver buffer size and timeout	145
Solr data tiers	147
Index data organization	150
Cold tier storage	151
Configuring data tiers	152
Number of shards	154
Concurrent searches	155
Configuring concurrent cold tier queries	156
Hardware sizing considerations	157
Summary	158
Student exercises	159
7 Backing up and restoring IBM Operations Analytics Log Analysis	160
Objectives	161
What is backed up	162
Back up and restore prerequisites	163
Limitations	164

Contents

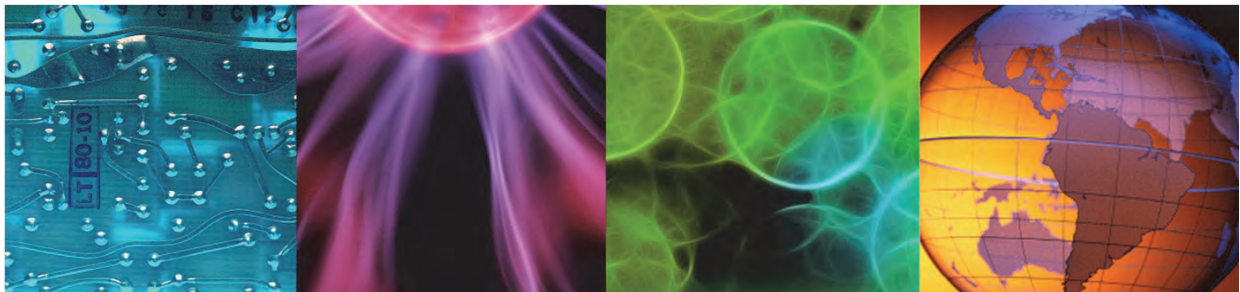
Backing up Log Analysis	165
Restoring Log Analysis	167
Troubleshooting	169
Summary	170



About this course



IBM Operations Analytics Log Analysis Administration



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this course, you learn how to administer, configure, and maintain IBM Operations Analytics Log Analysis. This course includes extensive lab exercises to give you hands-on practice with administration and configuration tasks.

The lab environment for this course uses the VMware virtual machine platform.

For information about other related courses, visit the Cloud & Smarter Infrastructure education training paths website:

ibm.com/software/software/tivoli/education/

Details	
Delivery method	Instructor-led
Course level	ERC 1.0
	This course is an update of TOD26 IBM SmartCloud Analytics Log Analysis 1.2.0.3 Administration ERC1.0

Details	
Product and version	IBM SmartCloud Analytics Log Analysis v1.3
Duration	2 days
Skill level	Intermediate

About the student

This course is designed for implementers, administrators, technical sales persons, and any others who need IBM Operations Analytics Log Analysis skills.

Before taking this course, make sure that you have Linux administration skills.

Learning objectives

Objectives

In this course, you learn to perform the following tasks:

- Explore IBM Operations Analytics Log Analysis and define the function of an Insight Pack
- Start and stop the application
- Add data sources and delete historical data
- Navigate the user interface
- Use the Generic Annotation Insight Pack
- Create an Insight Pack with the DSV tool kit
- Troubleshoot common problems with log files
- Configure alerts
- Configure IBM Operations Analytics Log Analysis to use Hadoop for long-term storage
- Tune host, operating system, and application settings
- Back up and restore IBM Operations Analytics Log Analysis

Course agenda

The course contains the following units:

1. [Overview and basic administration](#)

This unit briefly describes IBM Operations Analytics Log Analysis. It also shows you how to complete basic administrative tasks such as starting and stopping the application and user management.

In these exercises, you perform basic administration tasks, including application management, user management, and data storage housekeeping.

2. [Common configuration tasks](#)

This unit teaches you how to process log files that are not currently supported by Insight Packs. You learn about two techniques to process these log files: the Generic Annotation Insight Pack and the Delimiter Separated Value toolkit.

3. [Troubleshooting](#)

This unit explains the application log files that you should inspect to troubleshoot problems with IBM Operations Analytics Log Analysis.

4. [Alerts](#)

This unit teaches you how to use the alerts feature of IBM Operations Analytics Log Analysis. You learn how to create conditions that detect text patterns and actions that generate notifications when those conditions are met.

In this exercise, you configure the product to generate alerts from conditions in log files.

5. [Hadoop Distributed File System \(HDFS\) integration](#)

In this unit, you learn how to configure Log Analysis to store indexed data in Hadoop Distributed File System (HDFS) for long-term storage.

In these exercises, you configure IBM Operations Analytics Log Analysis to use Hadoop Distributed File System (HDFS) for long-term data storage.

6. [Performance tuning](#)

In this unit, you learn how to change host, operating system, and application settings to tune the performance of IBM Operations Analytics Log Analysis.

7. [Backing up and restoring IBM Operations Analytics Log Analysis](#)

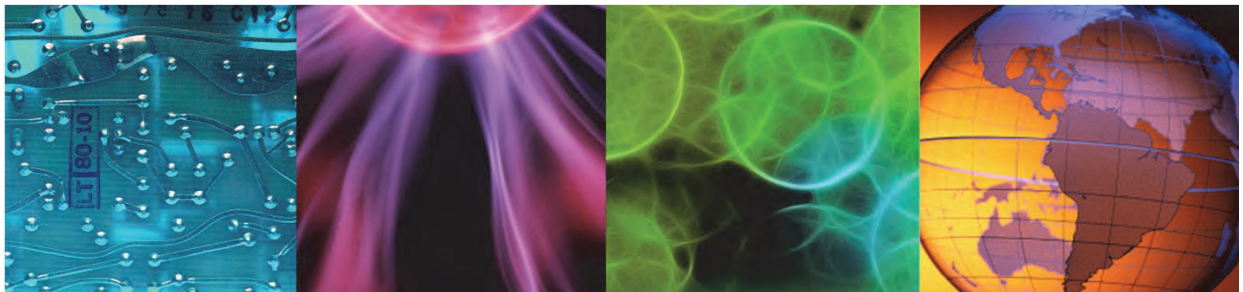
In this unit, you learn how to back up IBM Operations Analytics Log Analysis data and restore it on another system.



1 Overview and basic administration



1 Overview and basic administration



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit briefly describes IBM Operations Analytics Log Analysis. It also shows you how to complete basic administrative tasks such as starting and stopping the application and user management.

Objectives

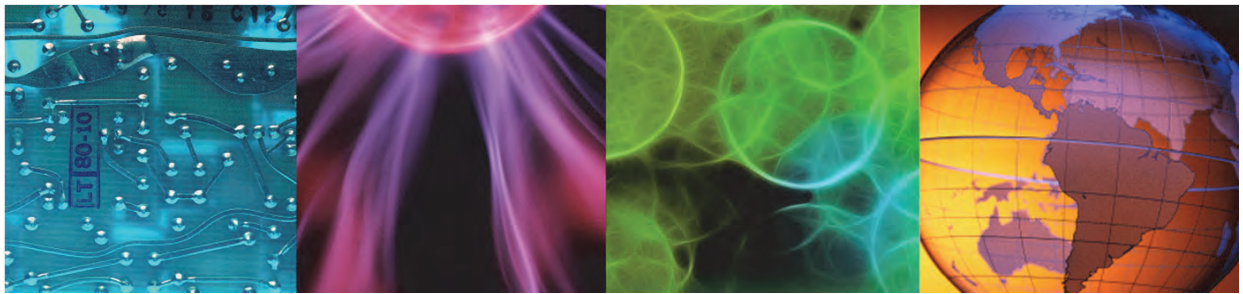
In this unit, you learn to perform the following tasks:

- Explore IBM Operations Analytics Log Analysis
- Define the function of an Insight Pack
- Start and stop the application
- Add data sources
- Delete historical data
- Navigate the user interface

Lesson 1 Overview



Lesson 1 Overview



© Copyright IBM Corporation 2016

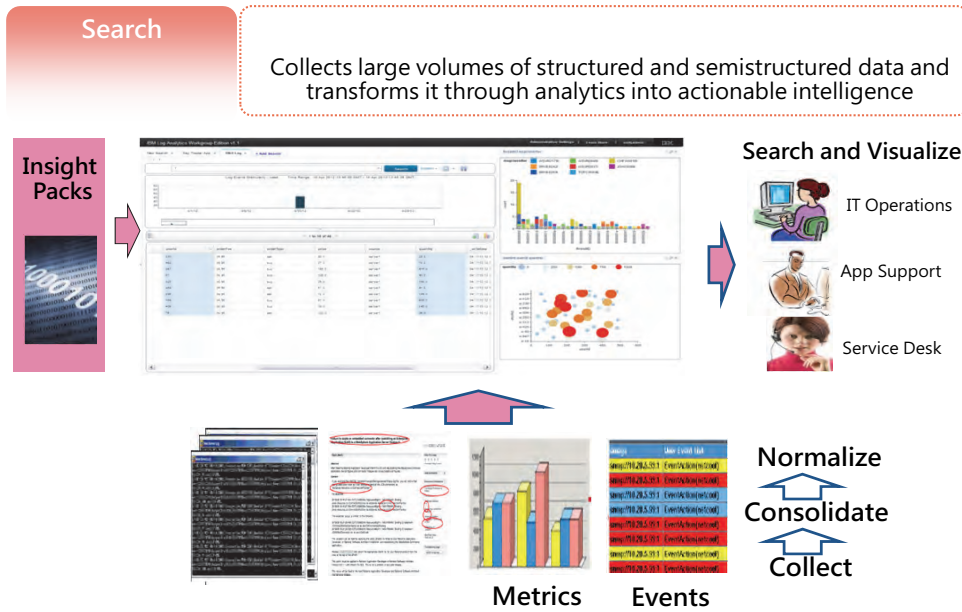
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This lesson teaches the overall function of IBM Operations Analytics Log Analysis and presents a tour of the user interface.

In this lesson, you learn how to perform the following tasks:

- Explore IBM Operations Analytics Log Analysis
- Define the function of an Insight Pack

IBM Operations Analytics Log Analysis defined



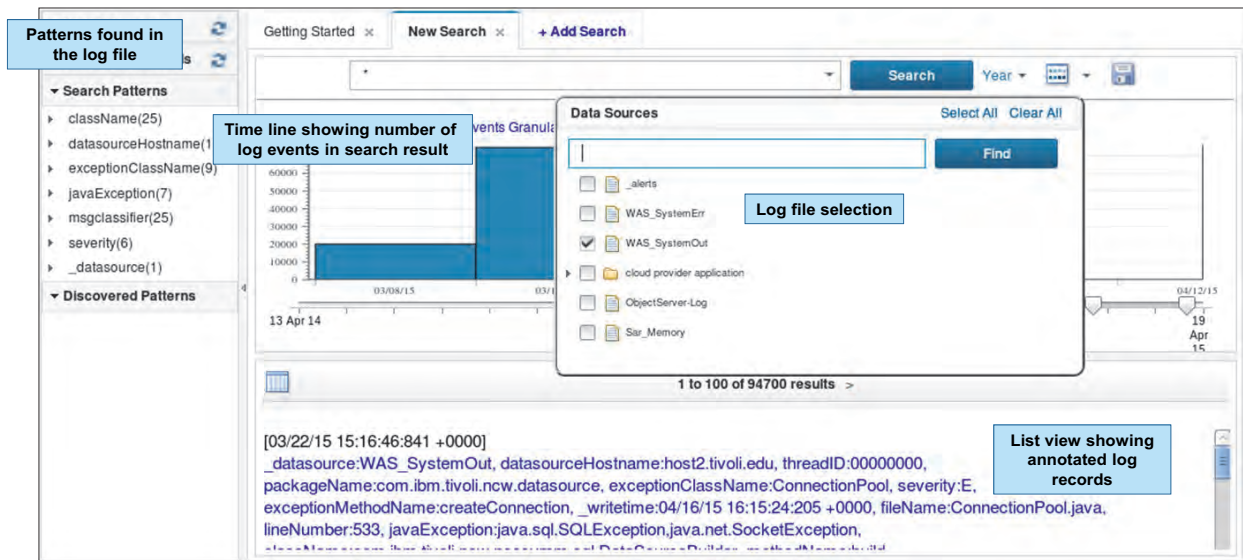
IBM Operations Analytics Log Analysis defined

With IBM Operations Analytics Log Analysis, you can analyze unstructured data to help identify, isolate, and resolve problems. The software integrates data from multiple sources including logs, events, metrics, support documents, and trouble tickets.

The user interface simplifies troubleshooting by presenting data from log files in a unified view. From the user interface, you can quickly find the cause of IT operational problems. You can also easily create charts and dashboards to show trends and summarizations of valuable log data.

Content for IBM Operations Analytics Log Analysis is provided by Insight Packs, which are product plug-ins that you can use to work with a particular technology or vendor.

User interface: Searching through logs



© Copyright IBM Corporation 2015

5

User interface: Searching through logs

You typically start using IBM Operations Analytics Log Analysis by searching through a log. You can select a single log file or multiple log files.

Use the search field to enter key words and operators to narrow your search. The following table defines some common operators.

AND	As an alternative to the + operator, you can use the AND operator. For example, to search for a specific severity and message classifier, enter <code>severity:W AND msgclassifier:"WLTC0032W"</code>
" " (Double quotations)	You can group individual terms into phrases that it searches for as a unit, for example, <code>"document clustering"</code> .
() (Parenthesis)	You can group expressions to guarantee precedence, for example, <code>document AND (cluster OR clustering)</code> .
*	Wildcard operator that can be replaced in the returned value with a number of characters.
?	Wildcard operator that is replaced in the returned value with a single character. This character might either be passed as an operator to the sources or expanded. For example, <code>bo?t</code> might return <code>boat</code> or <code>boot</code> .
\$	Unstemming operators. For example, <code>boat\$</code> might return <code>boat</code> or <code>boats</code> or <code>boating</code> .

+	To get AND like functions, use the plus (+) operator. You must add + as a prefix to these queries. For example, to search for a specific severity and message classifier, enter <code>+severity:W</code> <code>+msgclassifier:"WLTC0032W"</code> .
BEFORE	The specified term or expression must be before another term or expression in the search results, for example, <code>BEFORE clustering</code> . Variations to this keyword are <code>FOLLOWEDBY</code> and <code>THRU</code> .
field:	Use to restrict your query to a specific field. For example, <code>author:smith</code> or <code>title:"war and peace"</code> . These operators are activated for every field your syntax defines.
NEAR	Terms or expressions are matched in the results and contained within a specified proximity. You can apply it to any number of subexpressions. For example, <code>war AND peace NEAR (novel OR book)</code> . A variation to this keyword is <code>WITHIN X WORDS</code> .
NOT	The term or expression is not matched in the search results.
OR	Either term or expression is matched in the results. A variation to this keyword is <code>or</code> .

After you search for results, an interactive timeline is shown near the top of the window. The search results timeline shows a graph of the distribution of log events over a time period. You can use the timeline slider to view the logs for a specific duration. You can zoom in and out to change the range of the data visible.

The search results are listed below the timeline. This list shows records that match your search. You can switch between the default list view and a grid view, which shows log records in a tabular form.

On the left of the window, the Configured Patterns area shows text patterns that the application automatically finds in the log file.

User interface: Creating charts and dashboards

Dashboards show visualizations of text data

Charts and dashboard pages are quick and easy to edit

The screenshot displays the IBM Watson Analytics interface. On the left, a sidebar lists various dashboard templates under 'Dashboards' and 'Discovered Patterns'. The main area shows three dashboards: 'Diag level by component', 'Diag level by process', and 'Diag level by database'. Each dashboard features a stacked bar chart with a legend for diagnostic levels: Event (blue), Warning (green), Severe (yellow), Error (orange), and Info (red). The 'Diag level by database' chart is currently selected, and its configuration panel on the right shows the title 'Diag level by database', chart type 'Stacked Bar Chart', and parameters for x-axis (databaseName), y-axis (count), and categories (diagnosticLevel). A 'Render' button is visible at the bottom of the configuration panel.

© Copyright IBM Corporation 2015

6

User interface: Creating charts and dashboards

You can create charts from search results to reveal trends within your log data. These charts are easy to create and edit. Save these charts in dashboards, which you can reuse to provide a fast visual summary of logs, events, metrics, and other troubleshooting data.

User interface: Dashboard auto-refresh



© Copyright IBM Corporation 2016

7

User interface: Dashboard auto-refresh

Users can configure the charts and graphs on a dashboard page to automatically update as new data is processed. Use the **Actions** button at the upper right of a dashboard to set the refresh interval, or to disable auto refresh.



Important: You can configure a dashboard to automatically refresh only if all the charts in the dashboard use a relative time filter, for example last 10 minutes or last week.

By default, the maximum number of charts that can be refreshed simultaneously across all dashboards is 20. Edit the following property in the

`<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file:

```
#Maximum no. of charts allowed to auto refresh across dashboards  
MAX_AUTO_REFRESH_CHARTS=20
```

Adding the automatic refresh feature to existing dashboards

Some existing dashboards, such as those created with earlier versions of the product, do not have the automatic refresh feature enabled. To add the automatic refresh feature to an existing dashboard, add the following to lines to the .app file of the dashboard:

```
"searchType": "relative",  
"autoRefreshInterval": 1,
```

Add these lines after the name of dashboard in the .app file. In the following example, the automatic refresh feature has been added to the **Web Health Check Custom App** dashboard, which is configured by the `Web Health Check.app` file.

```
{
  "name": "Web Health Check Custom App",
  "type": "DynamicDashboard",
  "description": "Displays visualizations of data from the Web Access Log
Insight Pack",
  "searchType": "relative",
  "autoRefreshInterval": 1,
  "customLogic": {
    "script": "DynamicDashboard.sh",
    "description": "Total Daily Requests (by Hour)",
    "parameters": [
...

```

User interface: Compare messages from dissimilar log sources

message	timestamp	msgclassifier	_datasource
ADM5502W The escalation of "397" locks on table "ADMI...	04/16/13 12:04:14:000 +0000	ADM5502W	db2diag.log
ADM5502W The escalation of "396" locks on table "ADMI...	04/16/13 12:04:32:000 +0000	ADM5502W	db2diag.log
ADM5502W The escalation of "398" locks on table "ADMI...	04/16/13 12:04:52:000 +0000	ADM5502W	db2diag.log
ADM5502W The escalation of "410" locks on table "ADMI...	04/16/13 12:05:13:000 +0000	ADM5502W	db2diag.log
Connection not available while invoking method createOr...	04/16/13 12:05:13:537 +0000	J2CA0045E	SystemOut1.log
Connection not available while invoking method createOr...	04/16/13 12:05:13:537 +0000	J2CA0045E	SystemOut2.log

© Copyright IBM Corporation 2016

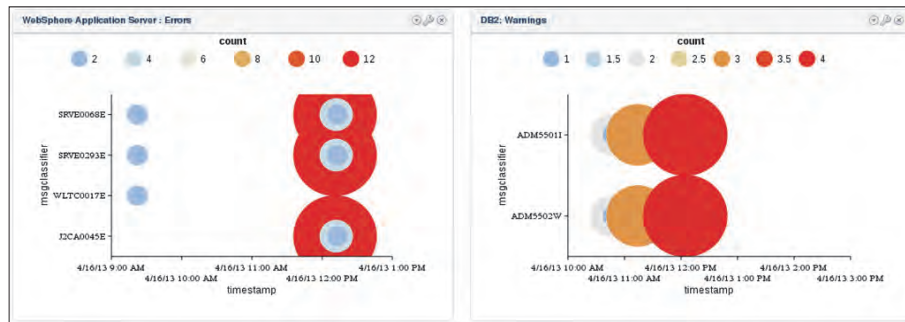
8

User interface: Compare messages from dissimilar log sources

A useful feature of IBM Operations Analytics Log Analysis is that you can load log data from different sources in to the user interface. You can then compare the health of different applications that make up a service.

In this example, a service depends on a web server and a database. You can combine and sort the log files of each underlying application by time stamp to determine which application is in a fault condition, and if that fault has an effect on the other application.

User interface: Create dashboards from dissimilar log sources



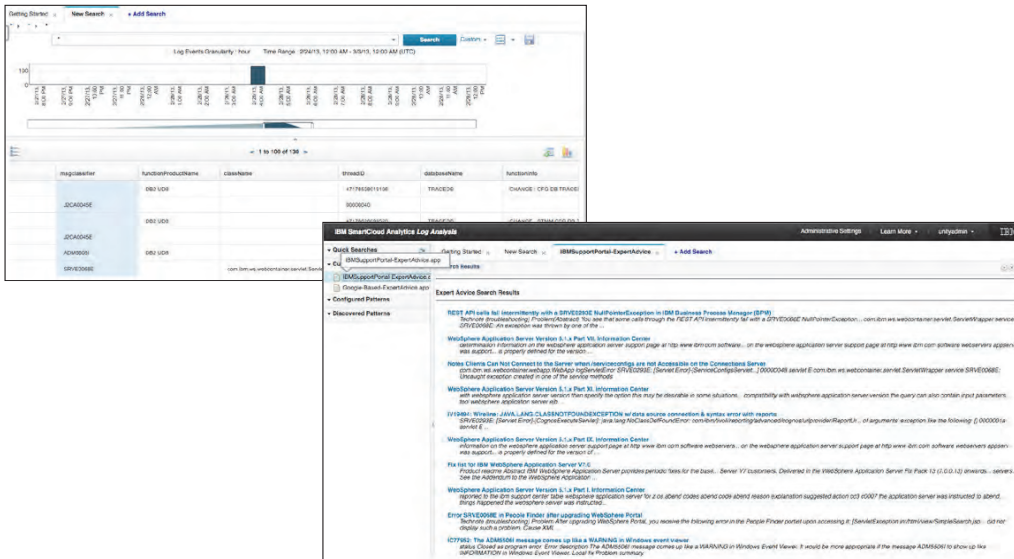
© Copyright IBM Corporation 2016

9

User interface: Create dashboards from dissimilar log sources

You can also create charts and dashboards from different types of log data. In this example, a service depends on a web server and a database. You can plot the log data from each application in common dashboards to show if there is any relation among the log messages.

User interface: Expert advice



© Copyright IBM Corporation 2016

10

User interface: Expert advice

Another key feature of IBM Operations Analytics Log Analysis is the expert advice search interface. You can select a field in a log record, such as an error code, and navigate to domain-specific information that matches your selection.

This expert advice search feature can link to an Internet search, a set of support documents, internal play books, or other information repositories that can aid in troubleshooting.

Insight Packs provide content

- Insight Packs install instructions on how to annotate and split log files
- Insight Packs provide dashboards
- Insight Packs are for a particular technology or application
- Examples of Insight Packs:
 - DB2, for `db2diag.log` files
 - WebSphere, for SystemOut, SystemErr, and trace
 - Web access, for Apache/IHS, Tomcat, and JBoss log files
 - Syslog, for syslog message files

© Copyright IBM Corporation 2016

11

Insight Packs provide content

Insight Packs are product add-ons that allow the product to annotate and interpret unstructured text. They often include dashboards that show visual representations of log data.

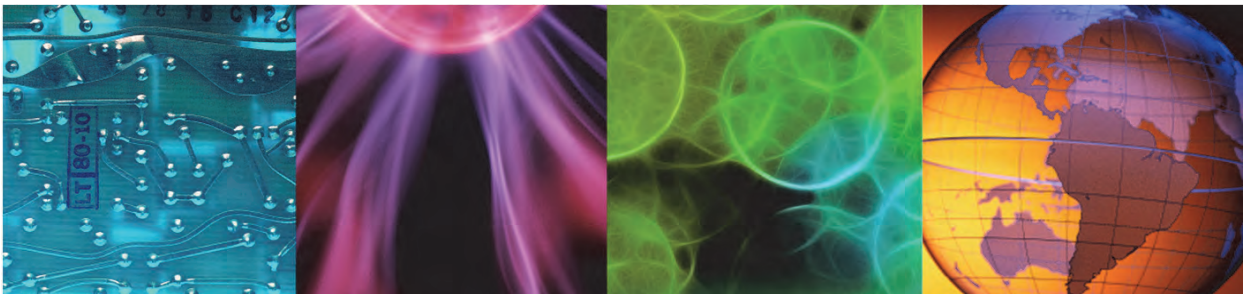
Install these Insight Packs as you need them. They are specific to an application, such as DB2®, WebSphere®, or Windows operating system events.

Insight Packs are released on a schedule that is independent of the core product release cycle. You can also create your own Insight Packs.

Lesson 2 Basic administration tasks



Lesson 2 Basic administration tasks



© Copyright IBM Corporation 2016

12

This lesson teaches you how to complete basic administration tasks, such as managing the application, managing users, adding data sources, and data storage housekeeping. This unit also demonstrates an example of a typical user work flow.

In this lesson, you learn how to perform the following tasks:

- Start and stop the application
- Add data sources
- Delete historical data
- Navigate the user interface

Starting and stopping the application

- Use the `unity.sh` tool to start and stop the IBM Operations Analytics Log Analysis applications
- The tool is in the `<LA_HOME>/utilities` directory, for example: `/opt/IBM/LogAnalysis/utilities`

- Examples:

```
unity.sh -start
unity.sh -stop
unity.sh -status
unity.sh -version
```

© Copyright IBM Corporation 2016

13

Starting and stopping the application

IBM Operations Analytics Log Analysis is made up of several applications:

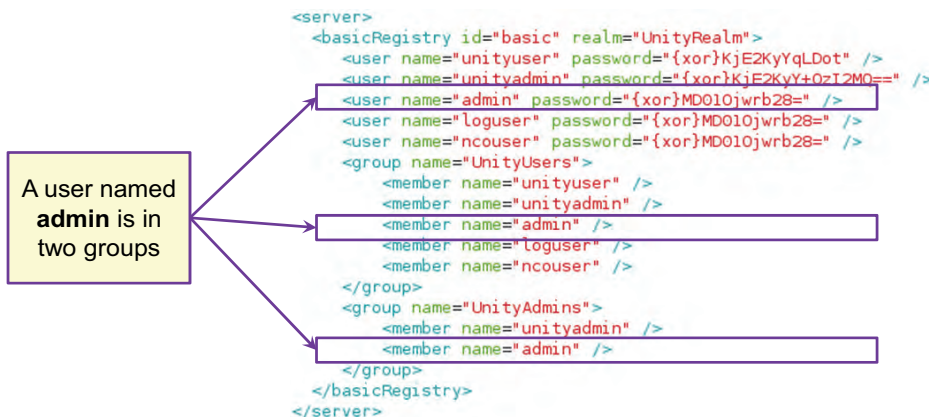
- Derby Network Server
- IBM WebSphere Liberty Profile
- Apache ZooKeeper
- IBM Tivoli Event Integration Facility Receiver
- IBM Tivoli Log File Agent
- Apache Solr

Use the `unity.sh` tool to start and stop all of these applications. In the following examples, IBM Operations Analytics Log Analysis is installed in the `/opt/IBM/LogAnalysis/` directory.

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

Adding users: File-based authentication

- User names and passwords are saved in the basic user registry file:
`<LA_HOME>/wlp/usr/servers/Unity/unityUserRegistry.xml`
- Add the user name and encoded password; then add the user to a group



© Copyright IBM Corporation 2016

14

Adding users: File-based authentication

To add users, edit the `unityUserRegistry.xml` file. This file is in the `<LA_HOME>/wlp/usr/servers/Unity/` directory.

Add individual user names and passwords inside opening and closing `<user>` tags. Use double quotation marks around the user name and password. Passwords must be encoded inside of this file.

Groups are defined with `<group>` tags. Use the `<member>` tag nested within the `<group>` tag to add a user to a group.



Important: If you create custom groups, you must assign them a role in the `unityConfig.xml` file, which is in the same directory. This file is explained in an upcoming slide.

Encoding passwords

- Use the **securityUtility** command located in the `<LA_HOME>/wlp/bin` directory to encode the password that you chose.
- Add the encoded password to the `unityUserRegistry.xml` file
- Examples:

```
./securityUtility encode object00  
{xor}MD010jwrb28=
```

```
./securityUtility encode p@ssw0rd  
{xor}Lx8sLChvLTs=
```

© Copyright IBM Corporation 2016

15

Encoding passwords

You must encode passwords in the `unityUserRegistry.xml` file. Use the `securityUtility` tool to encode plain text passwords.

In the following examples, IBM Operations Analytics Log Analysis is installed in the `/opt/IBM/LogAnalysis/` directory:

```
/opt/IBM/LogAnalysis/wlp/bin/securityUtility encode mypassword  
{xor}MiYvPiwsKDAtoW==
```

```
/opt/IBM/LogAnalysis/wlp/bin/securityUtility encode ibm4you  
{xor}Nj0yayYwKg==
```

Add the encoded version of the user password to the `unityUserRegistry.xml` file. You must include all characters in the user registry file.

Changing the password for unityuser

IBM Operations Analytics Log Analysis uses the **unityuser** account for some internal transactions. If you change the password of the **unityuser** account, you must edit the password in the following files:

- <LA_HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties
- <LA_HOME>/IBM/LogAnalysis/eif_remote_install_tool/config/rest-api.properties
- <LA_HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf
- <LA_HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance.sh
- <LA_HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh



Important: Some of these files require that the password of **unityuser** is encoded with XOR. Other files require that the password is encoded with AES.

XOR passwords

Use the following tool to encrypt XOR passwords:

```
<LA_HOME>/IBM/LogAnalysis/wlp/bin/securityUtility
```

In the following example, the plain text password is **object00**.

```
/opt/IBM/LogAnalysis/wlp/bin/securityUtility encode object00  
{xor}MD010jwrb28=
```

AES passwords

Use the following tool to encrypt AES passwords:

```
<LA_HOME>/IBM/LogAnalysis/utilities/unity_securityUtility.sh
```

In the following example, the plain text password is **object00**.

```
cd /opt/IBM/LogAnalysis/utilities/  
./unity_securityUtility.sh encode object00  
Using keystore file unity.ks.  
/opt/IBM/LogAnalysis/utilities/../../wlp/usr/servers/Unity/keystore/unity.ks  
{aes}79BC5AD667AB2957B9861BEA9026D9DA
```

Roles

Roles are assigned to groups and users in this file:

```
<LA_HOME>/wlp/usr/servers/Unity/unityConfig.xml
```

© Copyright IBM Corporation 2016

16

Roles

There are two default groups: UnityUsers and UnityAdmins. If you create custom groups, you must assign roles to the groups in the `unityConfig.xml` file.

In this example, a new group named MyNewGroup was added to the `unityUserRegistry.xml` file. You must add the new group to the `unityConfig.xml` file.

...

```
<application-bnd>
  <security-role name="UnityUser">
    <group name="UnityUsers" />
    <group name="UnityAdmins" />
    <group name="MyNewGroup" />
  </security-role>
  <security-role name="UnityAdmin">
    <group name="UnityAdmins" />
  </security-role>
</application-bnd>
</application>

<oauth-roles>
  <authenticated>
    <group name="UnityUsers" />
    <group name="MyNewGroup" />
  </authenticated>
</oauth-roles>
```

```
</authenticated>  
</oauth-roles>  
...
```

Data source definition

- A **data source** is a reference to a log file or other source of text, more or less
- Data sources configure the product to start processing a log file or other source of text, more or less
- Adding new data sources is a common administrative task:
 - You use a wizard to create them
 - You can also add data sources with an API

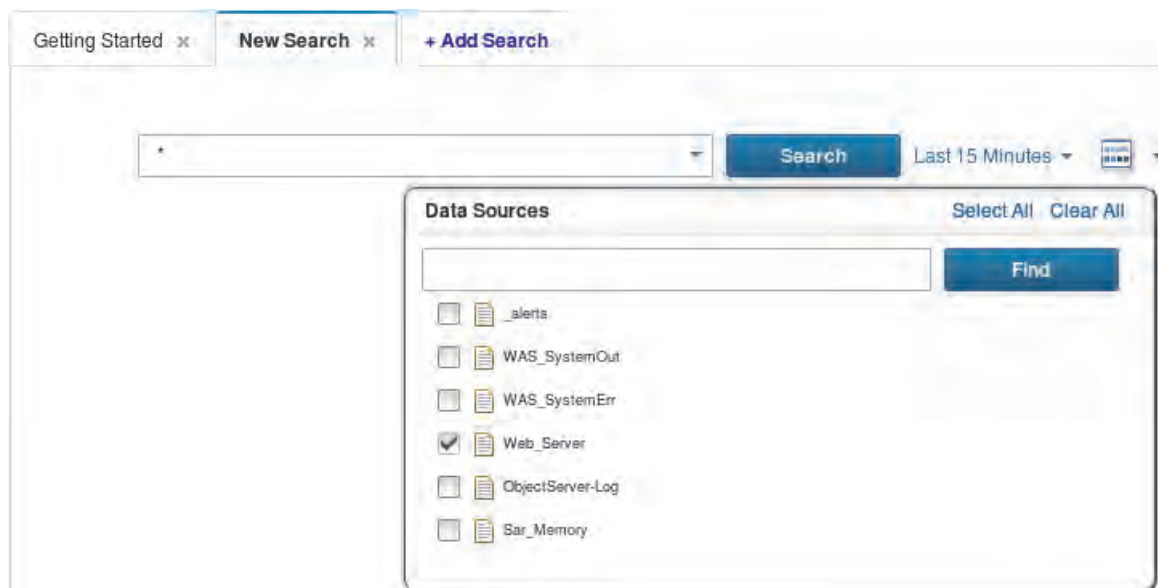
© Copyright IBM Corporation 2016

17

Data source definition

A data source is a configuration object that you use to process the contents of a log file. You create data sources to start processing a specific log source. Each log file that you want to monitor requires a unique data source, so adding data sources is a task that you perform often.

After you create a data source, it is available in the user interface for users to select.



Adding a data source

- A wizard interface guides you through data source configuration
- Validations occur at each step to avoid an invalid configuration
- Use the **Data Sources** tab in the administration user interface to add data sources



© Copyright IBM Corporation 2016

18

Adding a data source

You use a wizard in the administration user interface to add data sources. To add a data source, log in to the administration user interface and click the **Data Sources** tab. Click **New > Data Source** to start the wizard.

For streaming log files, the wizard automatically configures the IBM Tivoli® Log File Agent to monitor the data source and send any new messages to the log analysis server.

The information that you must enter in the wizard depends on the nature and location of the log file. The following table describes the fields in the data source wizard.

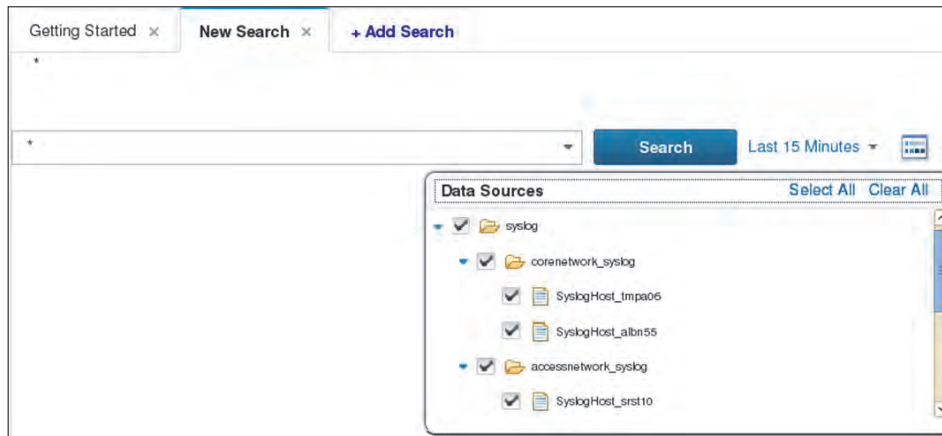
Field	Description
Select Location (Local file, Remote file, Custom)	The host where the log file is located. <ul style="list-style-type: none">• If you select <code>Remote file</code>, you must enter the host name and the operating system user name and password• If you select <code>Custom</code>, you must enter the host name
File path	The absolute path to the log file, including the file name
Type	The type of log file, such as syslog, Apache web access, or DB2
Collection	Use collections to group log data from different data sources that have the same source type. Select a collection to add the data source.

Field	Description
Rolling file pattern	This option is for log files that change their name according to a pattern, such as appending a date to the end of the file name. For example, a WebSphere Application Server log base line file name is <code>SystemOut.log</code> . The rolling file pattern is <code>SystemOut_*.log</code> .
Name	The name of the data source. This name is shown in the user interface.
Description	A description of the data source
Group	The group that you want to associate this data source with. You must configure a group before it is available to select in this field.

You use the same administration user interface to delete data sources.

Data source groups

Group data sources to represent the hierarchy of your services and applications



© Copyright IBM Corporation 2016

19

Data source groups

Groups are useful to users who want to search through multiple log files at the same time. Groups look like folders in the user interface. Data from dissimilar log files might be in the same group.

Groups are defined in the `unityServiceTopology.json` file. This file is in the `<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/com/ibm/tivoli/loganalytics/framework` directory.

In this example, the default `unityServiceTopology.json` file is edited to add three groups.

```
[
  {
    "type": "Service",
    "name": "Syslog",
    "value": [
      {
        "type": "Segment",
        "name": "CoreNetwork_Syslog",
        "value": []
      },
      {
        "type": "Segment",
        "name": "AccessNetwork_Syslog",
        "value": []
      }
    ]
  }
]
```

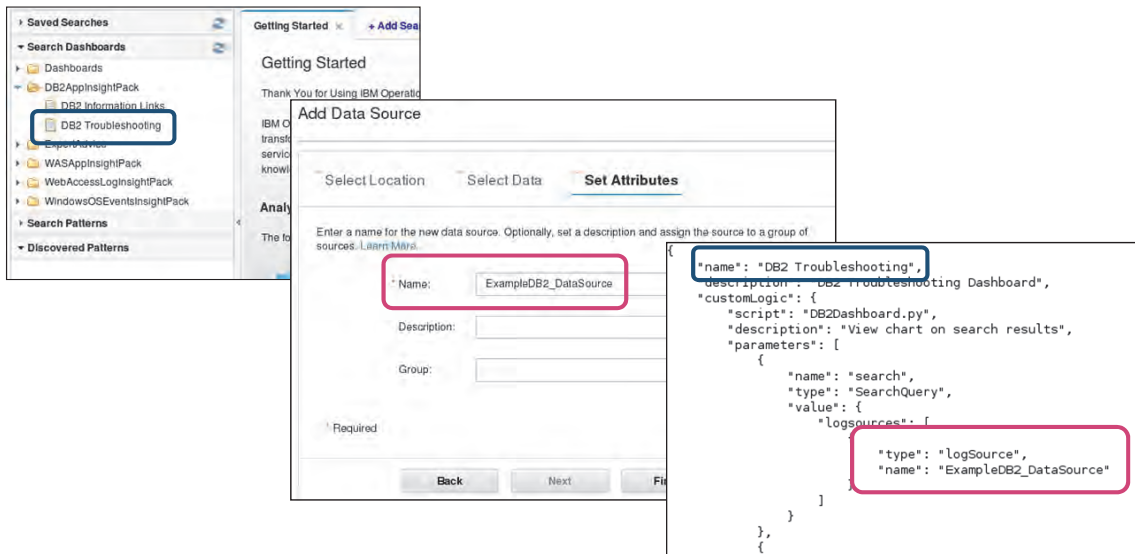
```
    ]  
  },  
  {  
    "type": "Service",  
    "name": "Day Trader",  
    "value": [  
      {  
        "type": "Application",  
        "name": "Trading Application",  
        "value": [  
          {  
            ...  
          }  
        ]  
      }  
    ]  
  }  
  ...  
}
```

After you create new groups in the `unityServiceTopology.json` file, you can use them in the data source wizard.



Important: Save a back-up copy of the default `unityServiceTopology.json` file before you make any changes.

Insight Pack dashboard configuration



© Copyright IBM Corporation 2016

20

Insight Pack dashboard configuration

Dashboards show visual representations of data in log files. Most Insight Packs install dashboards that show interesting facets of your log data. Before you can use these dashboards, you must edit them for your environment.

Configure dashboards by editing their .app files. These files are in the `<LA_HOME>/AppFramework/Apps` directory. In this directory are subdirectories for each of the installed Insight Packs.

At the minimum, you must edit each dashboard with the name of your data source. You can also change other settings in the .app file.

In the following example, the `Web Health Check.app` file is edited with the name of a data source.

```
pwd
/opt/IBM/LogAnalysis/AppFramework/Apps/WebAccessLogInsightPack_v1.1.0.2
```

```
...
vi Web\ Health\ Check.app
"name": "WebRequestsPerHour",
    "type": "FacetedSearchQuery",
    "value": {
        "start": 0,
        "results": 0,
        "filter": {
```

```
        "timefilters": {  
            "granularity" : "day",  
            "lastnum" : 1,  
            "type": "relative"  
        }  
    },  
    "logsources": [  
        {  
            "type": "logSource",  
            "name": "WebServer1"  
        }  
    ],  
    ...
```



Important: Some .app files define the data source more than once. Check the entire dashboard .app file to determine whether you must change one line or multiple lines.

User searches and chart creation work flow

- The user creates a new search
- The user switches to grid view to plot some charts
- The user clicks different data points on the charts to view the corresponding search results in a new search tab

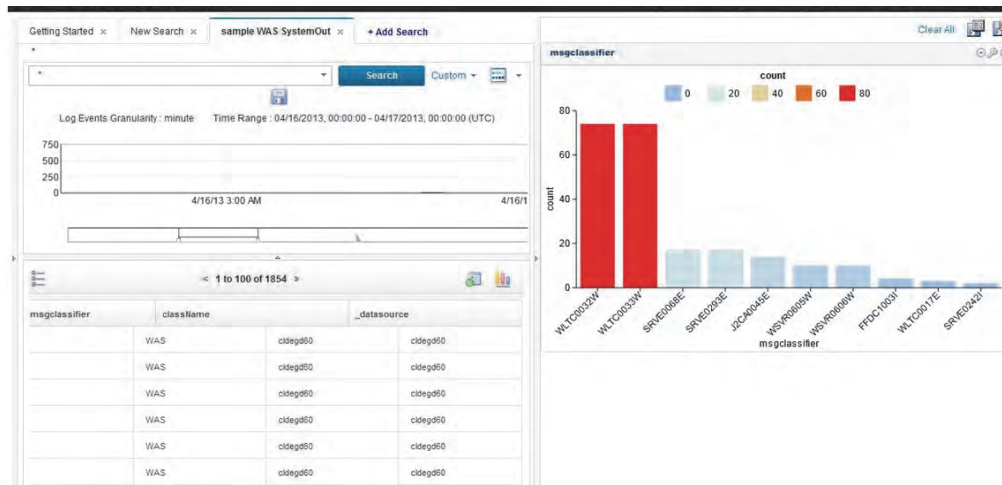
User searches and chart creation work flow

Users switch between log messages and charts when they analyze log files in the user interface. The following work flow is an example of how a user might interact with IBM Operations Analytics Log Analysis:

1. The user searches for messages in log files.
2. The user creates ad hoc charts and dashboards to help find trends and patterns in log data.
3. The user drills down from interactive charts to further refine the search.

The next few slides illustrate these actions.

Create new search and plot chart



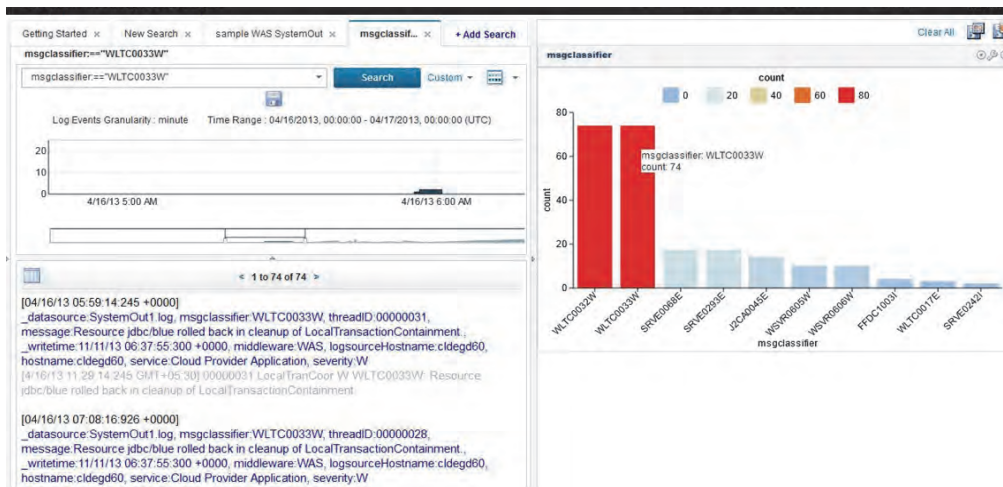
© Copyright IBM Corporation 2015

22

Create new search and plot chart

After you search for data, you can click the **Grid View** button to switch to a tabular view of the log messages. To plot log data in a graph, select a column heading and click the **Plot Data** button. You can also select multiple columns. After the chart is rendered, you can edit it to change the layout, data query, or other visual attributes.

Drill down from chart



Drill down from chart

Charts are interactive. You can hover the mouse pointer over a series in the chart to see the raw data points in hover help. You can also drill down from charts to narrow your search results.

In this example, you see a frequent occurrence of a particular error message classifier. When you click the series that represents the number of times this message classifier occurs in the log, you see a filtered list of log messages that correspond to the data series.

Interactive dashboards

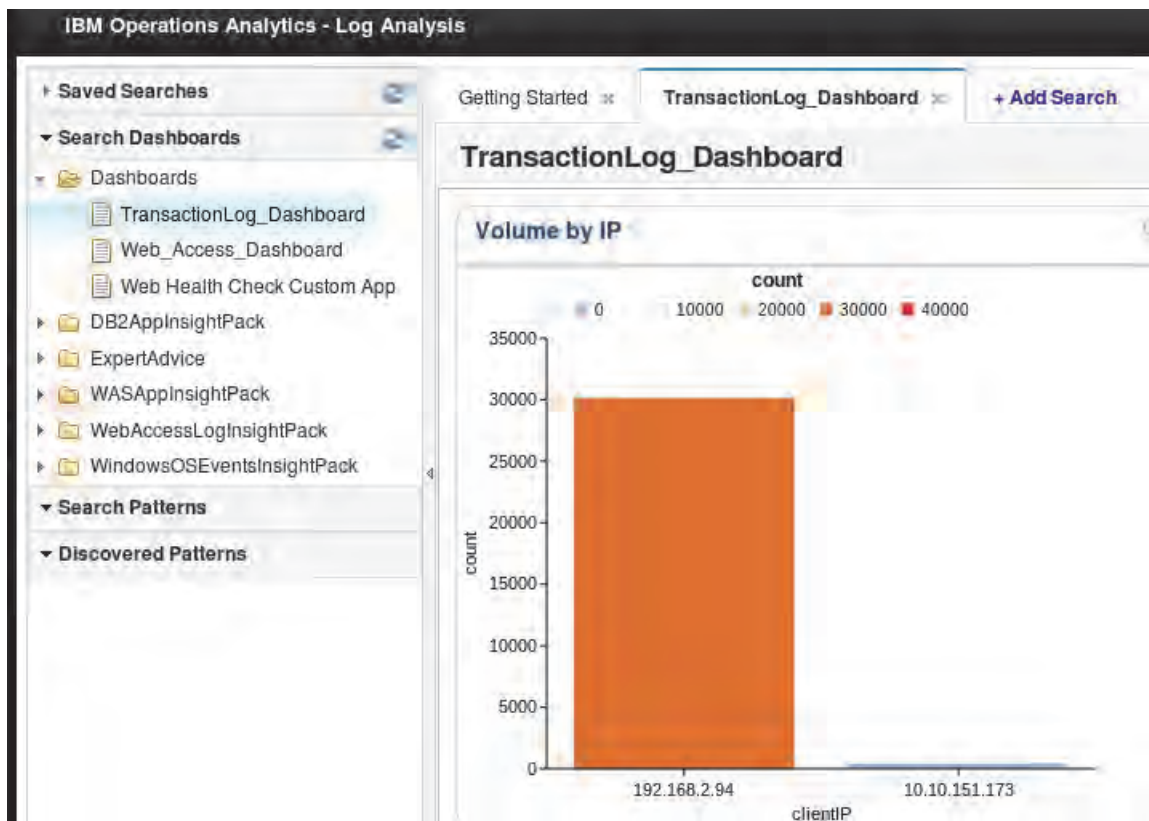


© Copyright IBM Corporation 2015

24

Interactive dashboards

You can create dashboards by saving charts. These reusable dashboards are also interactive. After you create a dashboard, it is available in the Search Dashboards area of the user interface.



An .app file is created for every dashboard. You can edit the dashboard by editing the corresponding .app file.

Viewing product usage

There are two ways to view how much log data has been processed:

1. Statistics page in the administrative user interface
2. Command line with the `export_statistics` tool

Viewing product usage

You can view the volume of data that IBM Operations Analytics Log Analysis processes in the administrator user interface or with the command-line interface. These usage statistics are maintained in internal database tables.

The recording of these statistics is also logged in the `<LA_HOME>/logs/UnityApplication.log` file. This log must be set to DEBUG to enable messages about product usage. Use the following command to find statistics messages.

```
grep -i statistics UnityApplication.log
```



Note: You learn more about IBM Operations Analytics Log Analysis logs in a later unit.

Server Statistics page



© Copyright IBM Corporation 2016

26

Server Statistics page

Click the **Server Statistics** tab in the Administrator interface to view usage information. The page shows the past 30 days of usage and a 30-day average.

The export_statistics tool

- The tool is in the `<LA_HOME>/utilities` directory
- The tool shows daily, thirtydayavg, and summary usage, the default
- Examples:

```
export_statistics -u unityadmin -p unityadmin -t daily
```

```
export_statistics -u unityadmin -p unityadmin -t summary
```

© Copyright IBM Corporation 2016

27

The export_statistics tool

The `export_statistics` tool also shows product usage. The options that are available with this tool show more information than the Server Statistics page. The following examples show the additional information that you can see with the `export_statistics` tool.

This example shows output from the `export_statistics` tool with the `-t daily` option. This shows the volume of data that was processed by the data source. The output is wrapped to fit the page.

```
/opt/IBM/LogAnalysis/utilities/export_statistics -u unityadmin -p unityadmin -t daily
```

Data Source Hostname	Collection	Date	Ingested Bytes	Billable	Log Path
DB2_Log /home/netcool/logs/db2diag.log	DB2_Log	2015-05-14	755578 analysishost	True	
IBM_HTTP_Log /home/netcool/logs/IHS-access.log	IBM_HTTP_Log	2015-05-14	1187312 analysishost	True	
NetworkDeviceMessages /home/netcool/logs/networkDeviceMessages.log	NetworkDeviceMessages	2015-05-14	695686 analysishost	True	
WAS_Trace /home/netcool/logs/WAStracelog	WAS_Trace	2015-05-14	9246223 analysishost	True	
omnibus omnihost.tivoli.edu	OMNIBus1100-Collection	2015-03-11	152599	True	NCOMS

This example shows output from the `export_statistics` tool with the `-t summary` option. This example shows the date of the highest usage within the 30-day average and the volume of data on that day. The output is wrapped to fit the page.

```
/opt/IBM/LogAnalysis/utilities/export_statistics -u unityadmin -p unityadmin -t  
summary
```

```
Current Thirty Day Rolling Avg | Thirty Day Rolling Avg High-Water Mark | Date  
of Thirty Day Rolling Avg High-Water Mark
```

```
-----+-----+-----  
-----  
396159 | 396159 |  
2015-05-14 15:09:52 +0000
```

Deleting data

- You delete data with a command-line tool: `deleteUtility.py`
- You configure what data to delete with the `delete.properties` file
- There are four use cases in the `delete.properties` file:
 - **useCase_1**: This option deletes all data from a single data source
 - **useCase_2**: This option deletes all data from a single collection
 - **useCase_3**: This option deletes all data from a start time to an end time
 - **useCase_4**: This option deletes all data that is older than a set retention period

© Copyright IBM Corporation 2016

28

Deleting data

Delete historical data with the `deleteUtility.py` tool. You run this tool from the command line, and you can schedule it in cron.

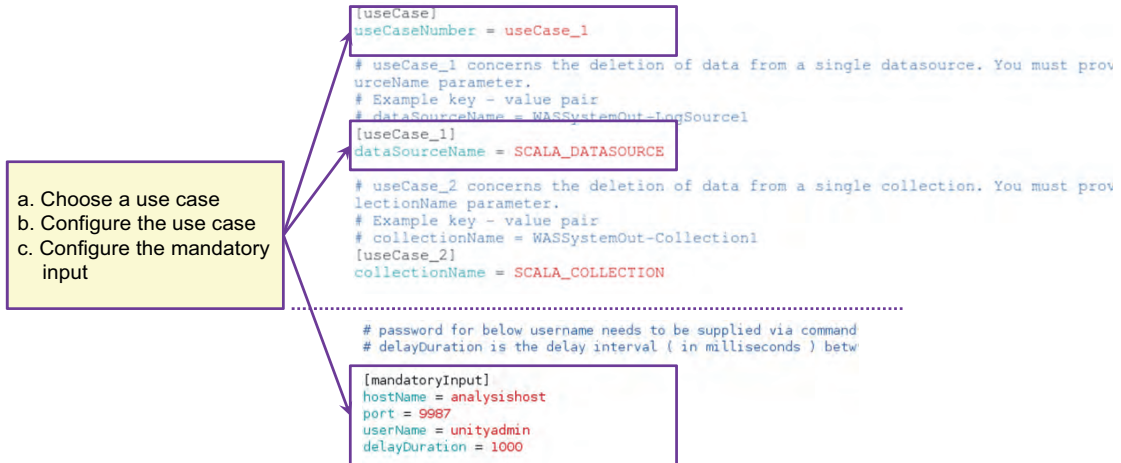
The `deleteUtility.py` tool logs the result of deletion to the `<LA_HOME>/logs/DeleteApplication.log` file.



Important: Before you can delete a data source, you must delete all historical data that is associated with it.

How to delete data

1. Edit the `delete.properties` file
2. Run the `deleteUtility.py` tool



© Copyright IBM Corporation 2015

29

How to delete data

To delete data, you must select the historical data that you want to delete with the `delete.properties` file. After you edit the `delete.properties` file, you run the `deleteUtility.py` tool.

Editing the `delete.properties` file

The `delete.properties` file is in the `<LA_HOME>/utilities/deleteUtility` directory. You must edit three sections of this file:

1. Find the **useCaseNumber** field close to the top of the file. Enter the use case that you want to use. The following values are valid for this field:
 - useCase_1
 - useCase_2
 - useCase_3
 - useCase_4
2. Find the use case configuration options for the use case you want to use. Edit the settings in the use case to select the data you want to delete. The fields that you must edit vary depending on the use case you choose.
3. Enter values for the **mandatoryInput** fields at the bottom of the file.

Running the deleteUtility.py tool

Include the path to Python and an administrative user's password when you run the deleteUtility.py tool, for example:

```
/usr/bin/python2.6 deleteUtility.py unityadmin
```

Add the `-cron` option to the end of the command to run the deleteUtility.py tool in cron mode. In cron mode, the tool automatically updates the startTime parameter that use case three requires, as in this example:

```
/usr/bin/python2.6 deleteUtility.py unityadmin -cron
```



Note: After you delete data, the measurements of how much data that was processed *do not* change. The Server Statistics page and the export_statistics tool both show how much data is processed, regardless of any data that is deleted.

Deleting data from cron

You can also run the deleteUtility.py tool from cron. To delete data from cron, follow these steps:

1. Edit the `<LA_HOME>/utilities/deleteUtility/callDeleteUtility.sh` script and update it with the password of an administrative user.
2. Run the following command to create the cron entry. This script is in the `<LA_HOME>/utilities/deleteUtility/` directory.

```
sh ./createCron.sh
```

Summary

You now should be able to perform the following tasks:

- Explore IBM Operations Analytics Log Analysis
- Define the function of an Insight Pack
- Start and stop the application
- Add data sources
- Delete historical data
- Navigate the user interface

Student exercises



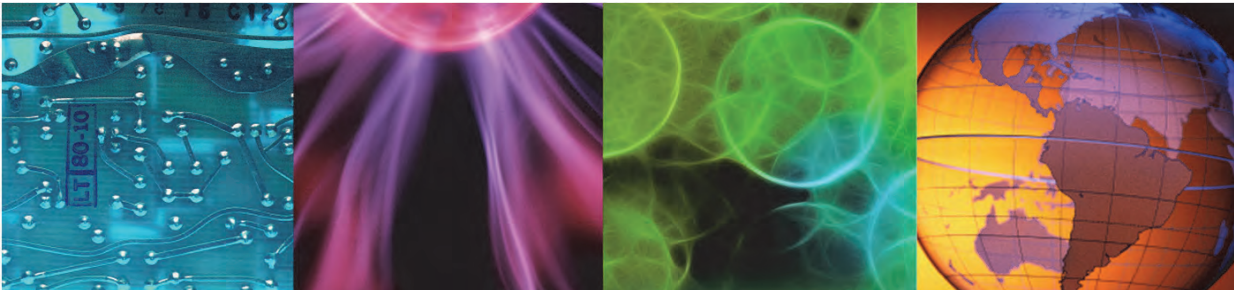
Student exercises



2 Common configuration tasks



2 Common configuration tasks



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches you how to process log files that are not currently supported by Insight Packs. You learn about two techniques to process these log files: the Generic Annotation Insight Pack and the Delimiter Separated Value toolkit.

Objectives

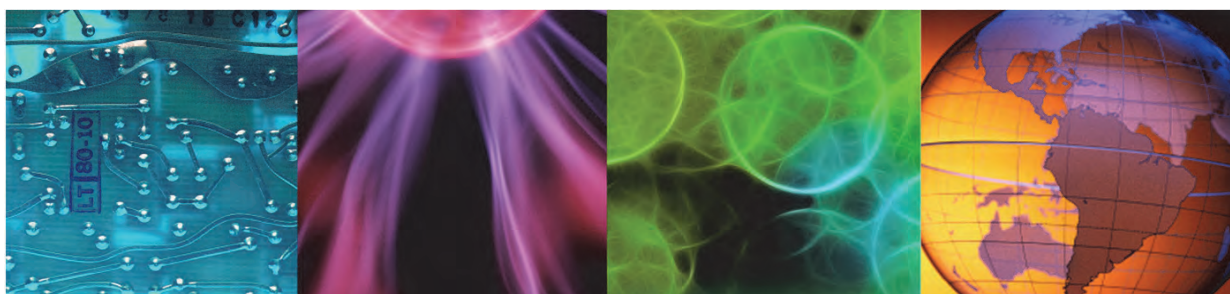
In this unit, you learn to perform the following tasks:

- Use the Generic Annotation Insight Pack
- Create an Insight Pack with the DSV toolkit

Lesson 1 Generic Annotation Insight Pack



Lesson 1 Generic Annotation Insight Pack



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to use the Generic Annotation Insight Pack.

Generic Annotation Insight Pack overview

- Apply the Generic Annotation Insight Pack to log files where a date and time stamp or only a time stamp can be identified within the records (lines) of the log file
- The pack identifies and outputs two kinds of annotations: Concepts and key-value pairs

Generic Annotation Insight Pack overview

You use the Generic Annotation Insight Pack to process log files that are not fully supported by an Insight Pack. The Generic Annotation pack inspects the log file for two kinds of patterns:

- **Concepts:** These are known patterns that are common in many log files, such as host name, IP address, and severity level, such as DEBUG, INFO, WARN. Concepts can also be any patterns, not known, that are found to repeat in a log sample.
- **Key-value pairs:** These are text strings that are separated by the equals (=) character. The generic annotator extracts text that is found in the pattern <key> = <value>. For example, `IVM` is extracted as the key and `56B` is extracted as the value in the string `IVM = '56B'`.

How to use the Generic Annotation Insight Pack

1. Create an index configuration
 - a. Inspect the date format of the log file that you want to use
 - b. Copy the Generic index configuration
 - c. Modify the dateFormats field in your index configuration
2. Create a source type, and use your new index configuration
3. Create a collection
4. Create a data source
5. Search through the log

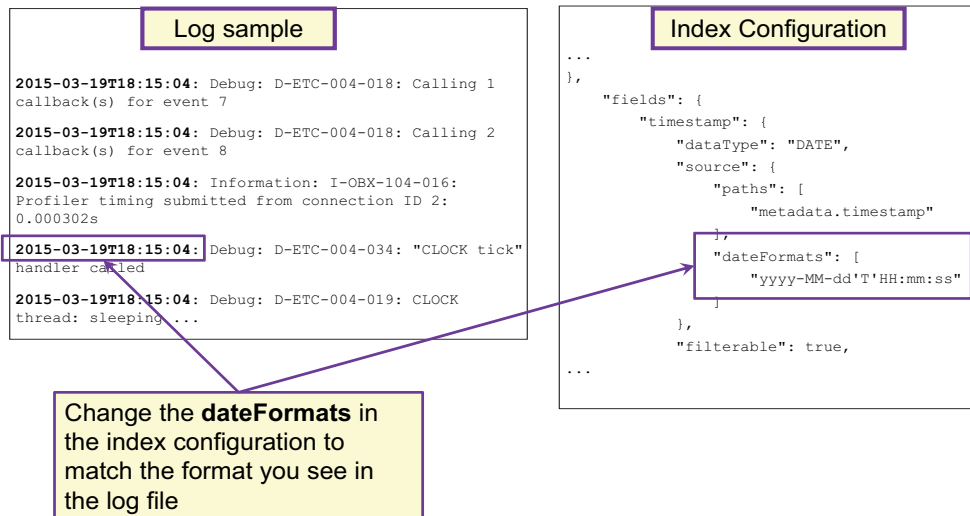
How to use the Generic Annotation Insight Pack

You must create the following configuration objects to process a log file with the Generic Annotation Insight Pack:

- An index configuration with the date format of the log you want to use. You can copy an existing index configuration and edit the copy.
- A source type that includes the index configuration. This source type is a set of instructions of how to index the fields in the log file.
- A collection. This object is a container that you can use to group log data from different logs with the same source type.
- A data source that identifies the host and directory location of the log file.

Use the administrator user interface to create these objects. These configuration objects are explained in the following slides.

Creating the index configuration



© Copyright IBM Corporation 2016

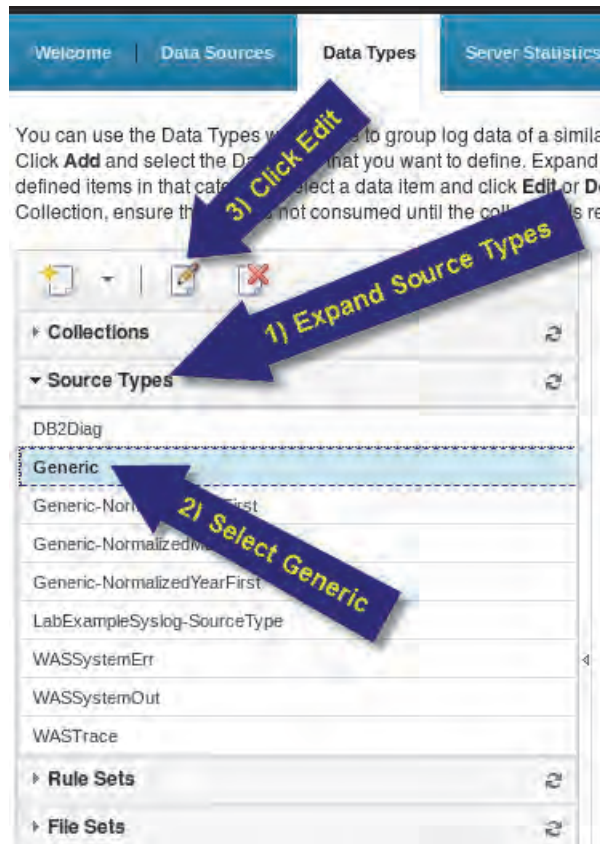
6

Creating the index configuration

An index configuration is part of a source type. There is a source type named `Generic` that contains an index configuration you can copy and modify. Your index configuration must be modified to match the date format in your log file.

Use the following steps to copy the `Generic` index configuration.

Use the **Data Types** tab to find the Generic source type and copy the index configuration.



Click the **View Index Configuration** button.

Edit Generic x

Edit Source Type

A Source Type defines how a particular kind of data is split, annotated, and ind

* Name:

* Input type:

Enable splitter

Rule set

File set

Enable annotator

Rule set

File set

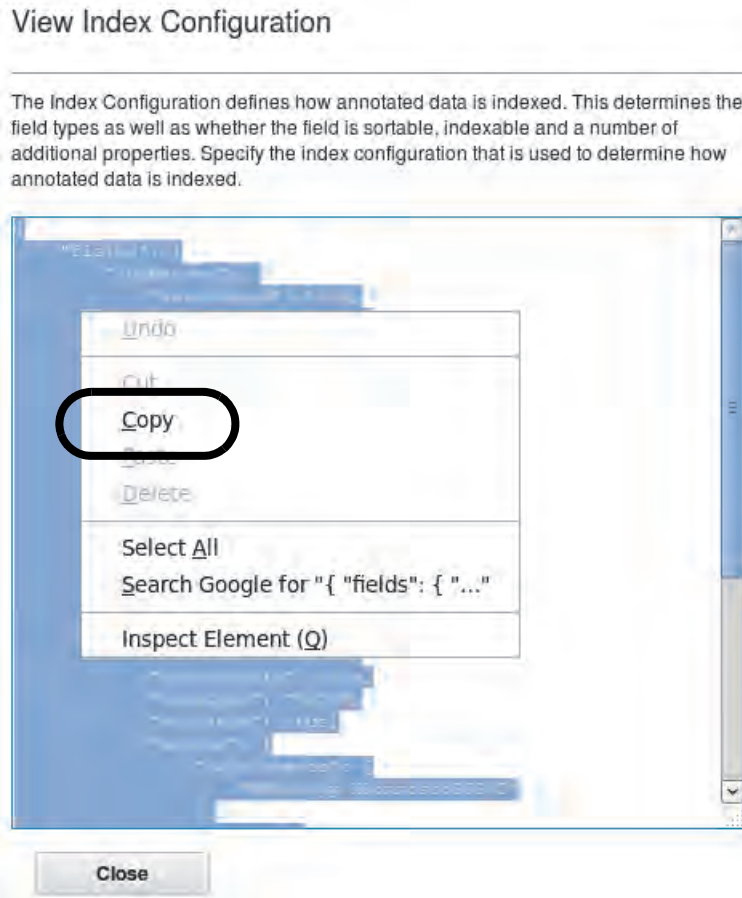
Deliver data on annotator execution failure

View Index Configuration

* Required

Close

Copy all of the text in the index configuration. Close the `Generic` source type when you are done.



Creating a source type

To create a source type:

1. Copy the index configuration from the *Generic* source type
2. Add a source type
3. Add the *Generic* index configuration to the new source type
4. Change the **dateFormats** value
5. Give the new source type a name, select the AQL rules, and save

Change the **dateFormats** in the index configuration to match the format you see in the log file

```
timestamp: {  
  "dataType": "DATE",  
  "source": {  
    "paths": [  
      "metadata.timestamp"  
    ]  
  },  
  "dateFormats": [  
    "yyyy-MM-dd'T'HH:mm:ss"  
  ],  
  "filterable": true,  
  "retrieveByDefault": true,  
  "sortable": true,  
  "searchable": true,  
}
```

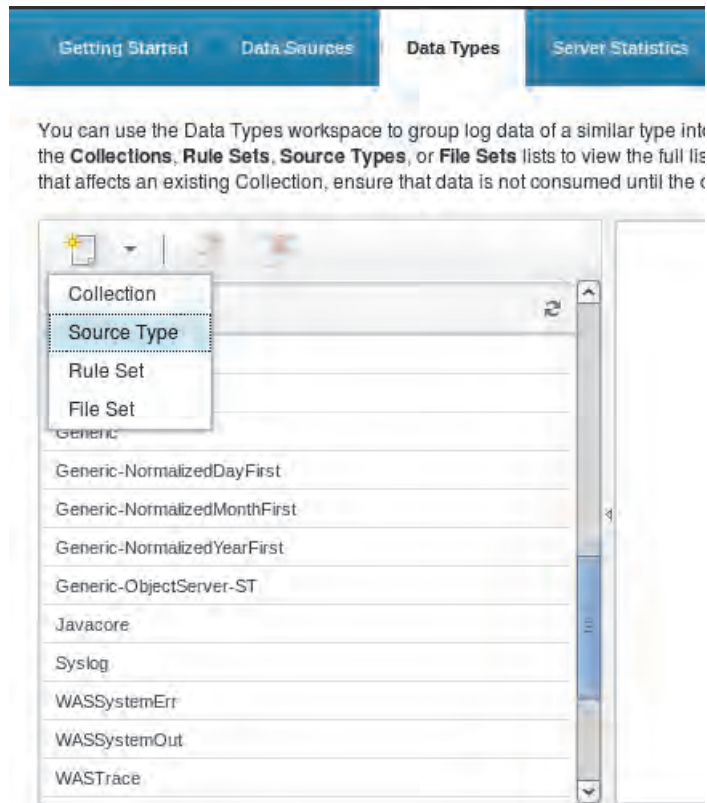
© Copyright IBM Corporation 2015

7

Creating a source type

After you copy the *Generic* index configuration, use it in a new source type.

In the **Data Types** tab, add a source type.



Click the **Edit Index Configuration** button.

Add Source Type x

Add Source Type

A Source Type defines how a particular kind of data is split, annotated, and indexed for searching. [Learn More...](#)

* Name:

* Input type:

Enable splitter

Rule set

File set

Enable annotator

Rule set

File set

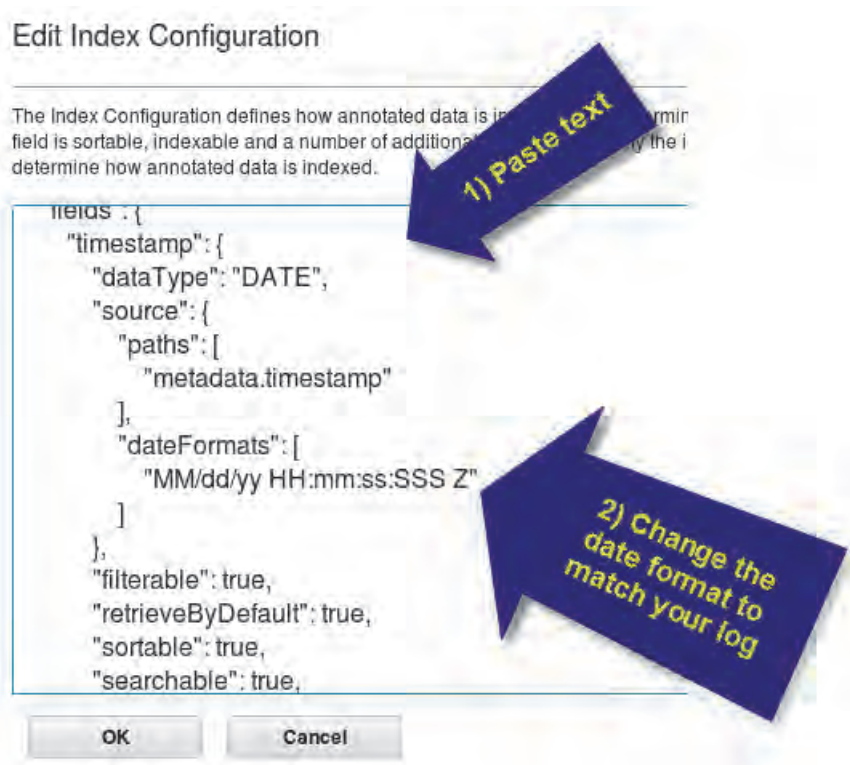
Deliver data on annotator execution failure

Edit Index Configuration

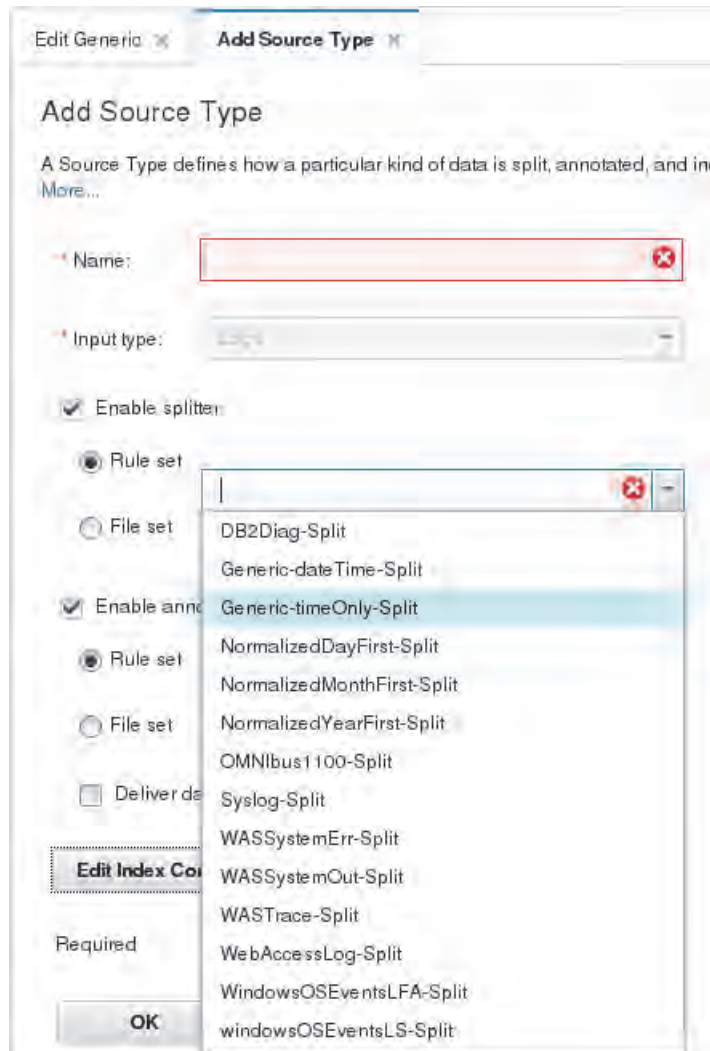
* Required

OK Cancel

Paste the index configuration that you copied from the Generic source type. Change the `dateFormats` field to match the date format in your log file. Click **OK**.



Enter a name for the source type. Select `Enable splitter` and choose the rule set that is appropriate for your log file. Select `Enable annotator` and choose `Generic-Annotate`. Click **OK**.



Creating a collection

Add Collection ✕

Add Collection

You can use Collections to group together log data from different data sources the Type. Before you complete the fields listed, ensure that a Source Type is available requirements. [Learn More](#).

* Name:

* Source Type:

* Required

OK **Cancel**

© Copyright IBM Corporation 2016

8

Creating a collection

In the **Data Types** tab, add a collection.

Welcome | Data Sources | **Data Types** | Server Statistics

You can use the Data Types workspace to group log data of a similar type into File Sets lists to view the full list of previously defined items in that category. refreshed. [Learn More](#).

- Collection
- Source Type
- Rule Set
- File Set

Collection

- Normalized-DayFirst
- Normalized-MonthFirst
- Normalized-YearFirst
- WASSystemErr-Collection1
- WASSystemOut-Collection1
- WASTrace-Collection1

Source Types

Rule Sets

Add Co

Add

You ca
your re

* Na

* So

Req

Enter a name for the collection. Choose your new source type and click **OK**.

Creating a data source

Use the new source type and collection in the data source

The screenshot shows a wizard window titled "Add Data Source". It has three tabs: "Select Location", "Select Data" (which is active), and "Set Attributes". Below the tabs, there is a text instruction: "Enter the location and type of data for this data source. The file path is not validated when you select the custom option. [Learn More...](#)". There are three input fields: "File path:" (a text box), "Type:" (a dropdown menu), and "Collection:" (a dropdown menu). A legend at the bottom left indicates that a red asterisk (*) denotes a required field. At the bottom of the window, there are four buttons: "Back", "Next", "Finish", and "Cancel".

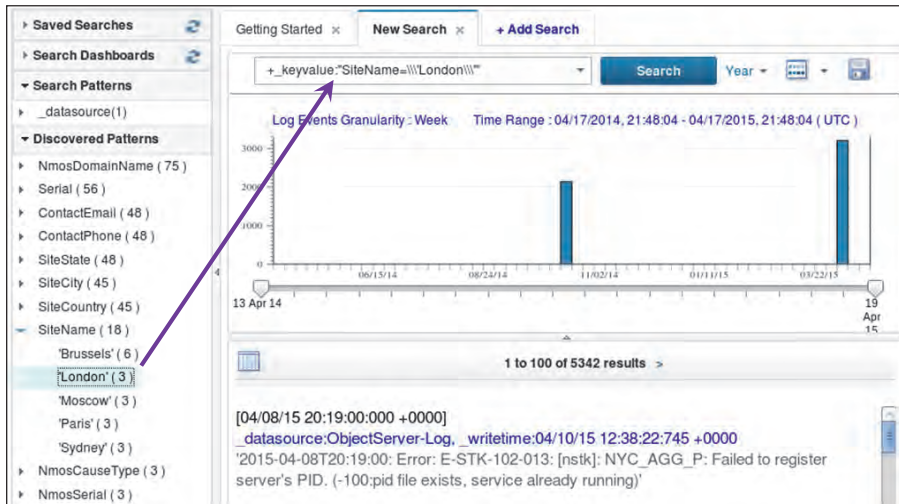
© Copyright IBM Corporation 2016

9

Creating a data source

Use the data source wizard to create a new data source. Use the new source type and collection when you configure the data source.

Discovered patterns



© Copyright IBM Corporation 2016

10

Discovered patterns

You see patterns that are found with the Generic Annotation Insight Pack in the **Discovered Patterns** pane on the left of the user interface.

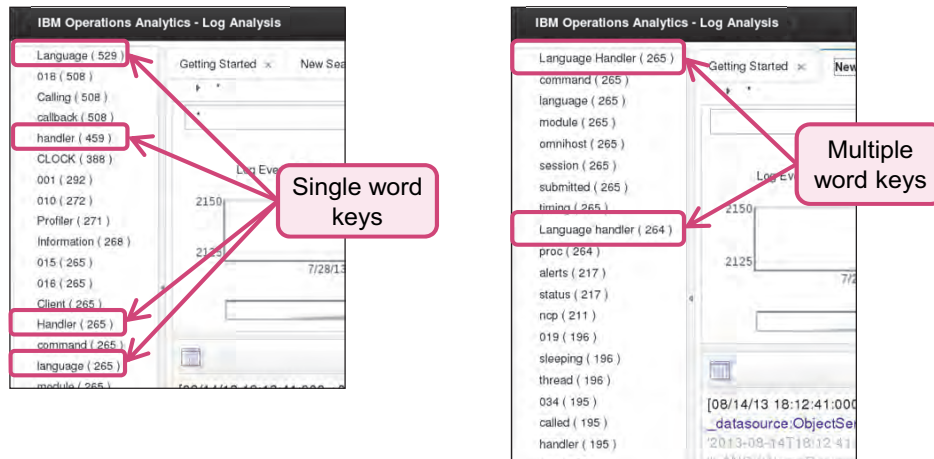
You can expand key-value pairs to show the values that are found in the log file. In this example, `SiteName` is the key and it has several values, such as `London` or `Paris`.

You can see concepts that are found in the log file listed below the key-value pairs. You can click any discovered pattern to use it as a search filter.

Multiple word keys

Log sample

```
2013-08-14T18:12:41: Debug: D-IVM-003-015: Language Handler (module session = 0xf9b950)
2013-08-14T18:12:41: Debug: D-OBX-105-010: Client language command on connection ID 6: [gateway]
2013-08-14T18:12:41: Debug: D-IVM-003-001: Language handler (proc = 0x7f95fc01cf18)
2013-08-14T18:12:41: Debug: D-IVM-003-015: Language Handler (module session = 0xfe2bc0)
2013-08-14T18:12:42: Debug: D-ETC-004-034: "CLOCK tick" handler called
```



© Copyright IBM Corporation 2015

11

Multiple word keys

The generic annotator is only able to identify single word keys or concepts. If you want to find multiple word keys in your log file, you must configure a custom dictionary file.

In this example, you want to find the pattern `language handler`, which is a two-word pattern. The log contains these words in several forms:

- `Language Handler`: This is a two-word pattern you want to find.
- `Language handler`: This is a two-word pattern you want to find.
- `language`: This word is in the log file many times without the word `handler`.
- `handler`: This word is in the log file many times without the word `language`.

The screen capture on the left shows discovered patterns without any customization. Each form of the single words is identified as a pattern: `Language`, `language`, `Handler`, and `handler`.

The screen capture on the right shows discovered patterns after the custom dictionary file is modified. The variations of the two-word key are found: `Language Handler` and `Language handler`. The instances of `language` and `handler` that are in the log file alone were also discovered.

To configure multiple word patterns, you must edit the following file:

```
<LA_HOME>/unity_content/GAInsightPack_v1.1.1.1/extractors/ruleset/GA_common/dicts/
userSpecifiedStrings.dict
```

In this example, the two-word pattern `Language handler` is added to this file.

```
more userSpecifiedStrings.dict
# this file can be used to list strings for both KVP & new concepts
Language handler
```

Add each multiple-word pattern on a single line in this file.

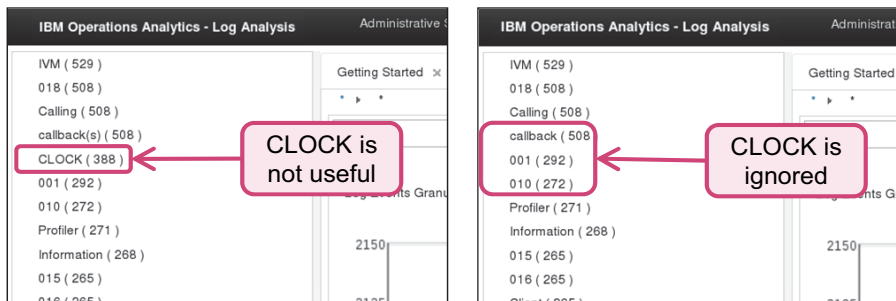


Important: After you modify the `userSpecifiedStrings.dict` file, you must restart IBM Operations Analytics Log Analysis.

Stop words

Log sample

```
2013-08-14T18:12:41: Debug: D-IVM-003-015: Language Handler (module session = 0xf9b950)
2013-08-14T18:12:41: Debug: D-OBX-105-010: Client language command on connection ID 6: [gateway]
2013-08-14T18:12:41: Debug: D-IVM-003-001: Language handler (proc = 0x7f95fc01cf18)
2013-08-14T18:12:41: Debug: D-IVM-003-015: Language Handler (module session = 0xfe2bc0)
2013-08-14T18:12:42: Debug: D-ETC-004-034: "CLOCK tick" handler called
```



© Copyright IBM Corporation 2015

12

Stop words

You can also configure the generic annotator to ignore words in your log file. To omit a word from discovery, edit this file:

```
<LA_HOME>/unity_content/GAInsightPack_v1.1.1.1/extractors/ruleset/GA_common/dicts/stopwords.dict
```

Add each stop word on a single line in this file.

In this example, you are not interested in the word `CLOCK`. The word `CLOCK` is added to this file. Add each stop word on a single line in the `stopwords.dict` file:

```
tail -5 stopwords.dict
yourselves
youve
z
zero
CLOCK
```

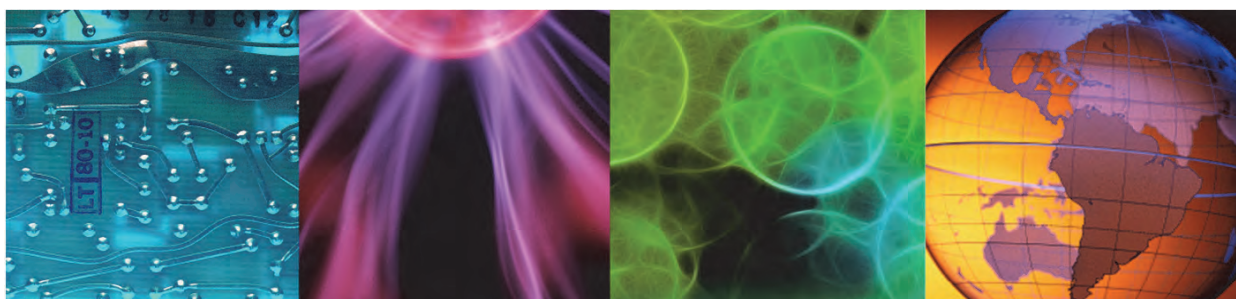


Important: After you modify the `stopwords.dict` file, you must restart IBM Operations Analytics Log Analysis.

Lesson 2 Delimiter-separated value toolkit



Lesson 2 Delimiter-separated value toolkit



© Copyright IBM Corporation 2016

13

In this lesson, you learn how to create a new Insight Pack with the delimiter-separated value toolkit.

Overview of the delimiter-separated value (DSV) toolkit

- Use it to create Insight Packs for logs file that are separated by a delimiting character
- It is based on Python scripts
- It uses a properties file to describe the fields in a log file

Overview of the delimiter-separated value (DSV) toolkit

You use the delimiter-separated value toolkit to quickly create an Insight Pack for log files that are not supported. You can create a full Insight Pack in only four steps. You do not have to know how to create an Insight Pack to use the toolkit. The toolkit scripts that create an Insight Pack from a properties file are installed with IBM Operations Analytics Log Analysis.

Log file requirements

- Each log record must be on a single line
- The last field in a log record must not end with the delimiting character
- The following delimiters are supported
 - comma (,)
 - colon (:)
 - semicolon (;)
 - pipe (|)
 - dash (-)
 - slash (/)
 - backslash (\)
 - tab (\t)

Log file requirements

Only log files that meet these requirements are supported by the delimiter separated value toolkit.

The format of the log file must adhere to the following specifications:

- The log must generate only one message per line.
- The delimiting character in the log file must be one of the supported characters.
- The last field in the log message cannot end with the character that is used as the delimiter.

How to use the DSV toolkit

1. Identify the fields in the log file you want to see as columns in the search workspace
2. Create a properties file with the `primeProps.py` script
3. Edit the properties file to match your log file
4. Generate the Insight Pack and deploy it with the `devGen.py` script

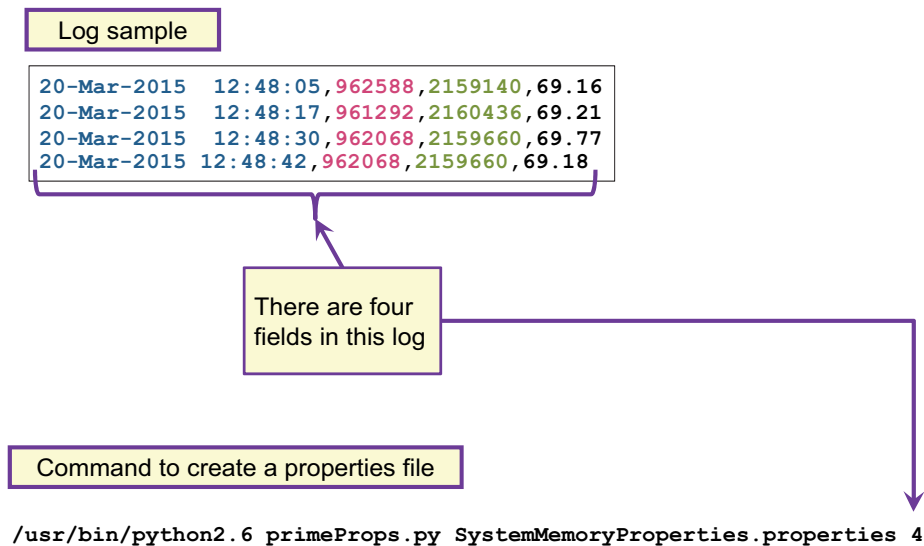
How to use the DSV toolkit

To use the toolkit, inspect the log file that you want to use. Identify the fields in the log messages that you want to use in the IBM Operations Analytics Log Analysis user interface. Each of the fields that show interesting data require configuration with a text editor.

Use the `primeProps.py` script to create a properties file. You modify this properties file to configure each interesting field in the log file.

After you modify the properties file, use the `devGen.py` script to create the Insight Pack. These tasks are described in the following slides.

Creating a properties file



© Copyright IBM Corporation 2015

17

Creating a properties file

In this example, there are four fields in the log file. The command to create the properties file for this log sample contains the following options.

- The path to Python. In this example, the path is `/usr/bin/python2.6`.
- The script `primeProps.py`. This script creates the properties file.
- The name of the properties file. Choose a name that describes the content of the log file. In this example, the name of the file is `SystemMemoryProperties.properties`.

The number of columns in the log file that you want to process. In this log sample, there are four fields, including the time stamp field.

Editing the properties file

```
[SCALA_server]
scalaHome: /opt/IBM/LogAnalysis

[DSV_file]
delimiter: ,
totalColumns: 4
aqlModuleName: sarmemDSV4Column
version: 1.0.0.0

[field0_indexConfig]
name: logRecord
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
path_1: content.text
combine: FIRST

[field1_indexConfig]
name: timestamp
dataType: DATE
retrievable: true
retrieveByDefault: true
sortable: true
filterable: true
searchable: true
dateFormat: dd-MMM-yyyy HH:mm:ss

[field2_indexConfig]
name: memoryFreeInKb
dataType: LONG
retrievable: true
retrieveByDefault: true
sortable: true
filterable: true
searchable: true

[field3_indexConfig]
name: memoryUsedInKb
dataType: LONG
retrievable: true
retrieveByDefault: true
sortable: true
filterable: true
searchable: true

[field4_indexConfig]
name: memoryUsedPercent
dataType: DOUBLE
retrievable: true
retrieveByDefault: true
sortable: true
filterable: true
searchable: true
```

© Copyright IBM Corporation 2016

18

Editing the properties file

After you run then `primeProps.py` file, a `.properties` file is generated. You must edit this file to describe the fields in the log messages you want to see in the user interface.

Each field is assigned an identifier: `[fieldN_indexConfig]`, followed by a paragraph for each of the fields you specified when you ran the `primeProps.py` script. In this example, there are four properties file paragraphs to describe the interesting fields in the log file:

`[field1_indexConfig]` to `[field4_indexConfig]`. The number in the field identifier corresponds to the position of the interesting field in the log file.

There are two mandatory values that you must set in the heading of this file:

- `scalaHome`: Set this value to the directory where IBM Operations Analytics Log Analysis is installed.
- `aqlModuleName`: The value in this field becomes the name of the Insight Pack.

The first paragraph, `[field0_indexConfig]`, sets the mapping for the rest of the file. Do not edit this field.

One of the paragraphs must describe the time stamp field of your log file. Set the name of this paragraph to `timestamp` and change the `dataType` to `DATE`. Add a line to this paragraph for the property `dateFormat` and enter the time stamp format in your log file. Set all of the other properties in the `timestamp` paragraph to `true`.



Important: The date format must conform to the Java 7 `SimpleDateFormat` class specification.

Edit the other paragraphs to describe the remaining fields in your log file. The following list explains the properties in these paragraphs:

- Enter a `name` for the rest of the paragraphs. The name that you enter becomes the name of the corresponding column in the search interface.
- Enter a `dataType` for the rest of the paragraphs. Valid values for this property are `TEXT`, `DATE`, `LONG`, or `DOUBLE`.
- Enter `true` or `false` for the value of the index configuration properties. The following table describes these properties.

Property	Description
<code>retrievable</code>	Determines whether the contents of this field are stored for retrieval. When set to <code>false</code> , the content is not stored in the index. When set to <code>true</code> , the content is stored and available for retrieval. The <code>retrieveByDefault</code> value controls how and when the content of this field is included in search results.
<code>retrieveByDefault</code>	When set to <code>true</code> , the contents of the field are always returned as part of any search response. When set to <code>false</code> , the field is not part of the default response. However, when required, the content of the field can be explicitly requested using the parameters that are supported by the search run time. The <code>retrievable</code> property must be set to <code>true</code> for this attribute to work.
<code>sortable</code>	Enable or disable the field for sorting and range queries.
<code>filterable</code>	Enable or disable facet counting and filtering on this field.
<code>searchable</code>	Controls whether the field is enabled for searching and matching.

Save the properties file after you finish editing it.

Creating and deploying the Insight Pack

- Running the `devGen.py` script creates the Insight Pack
- Command options can deploy the pack

Example of a command to create the Insight Pack and deploy it

```
/usr/bin/python2.6 dsvGen.py sarMemProperties.properties -o  
-d -u unityadmin -p unityadmin
```

© Copyright IBM Corporation 2015

19

Creating and deploying the Insight Pack

Use the `dsvGen.py` tool to create the Insight Pack. Include the path to Python and the name of your properties file when you run the `dsvGen.py` tool. If you want to deploy the Insight Pack, include the `-d` option and an administrative user name and password.

In this example, the name of the properties file is `sarMemProperties.properties`. The `-o` option overwrites the previous version of the Insight Pack. The `-d` option deploys the pack. The user name and password is `unityadmin`.

After you create the Insight Pack, you create a data source to process the log file. This task is no different from creating a data source for any other type of log file.

Excluding and combining fields

- Exclude a field in the log by omitting `fieldN_indexConfig` sections from the properties file
- Use paths to combine the values in multiple fields, for example:

```
[field2_indexConfig]
name: shortMessage
dataType: TEXT
retrievable: false
retrieveByDefault: false
sortable: false
filterable: false
searchable: false

[field4_indexConfig]
name: longMessage
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
path_1: annotations.csv5Col_shortMessageFinal.shortMessage
combine: ALL
```

© Copyright IBM Corporation 2016

20

Excluding and combining fields

You can configure the properties file to exclude and combine columns from your log file when they are shown in the search interface.

If you do not want to show a field in your log file, omit the corresponding paragraph in the properties file. In this example, only Fields 2 and 4 are included.

You can combine multiple fields from the log file in to one column in the search interface. Use the `path_n` property to specify multiple paths in one paragraph. The paragraph with multiple paths must have the highest column number.

Summary

You now should be able to perform the following tasks:

- Use the Generic Annotation Insight Pack
- Use the DSV toolkit to create an Insight Pack

Summary

Student exercises



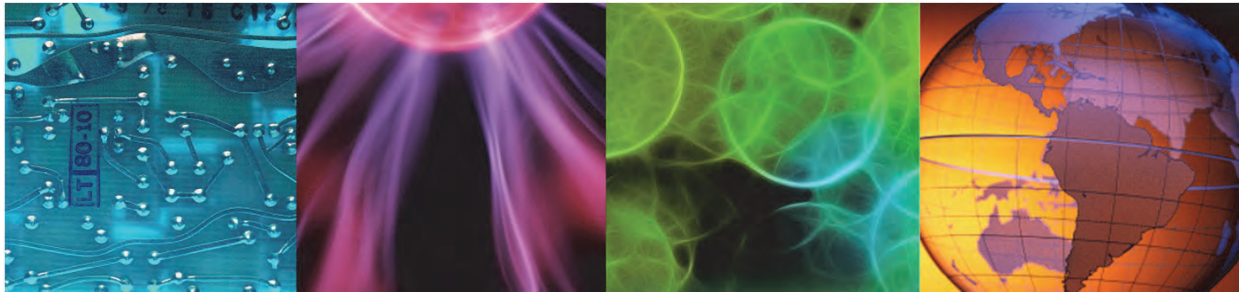
Student exercises



3 Troubleshooting



3 Troubleshooting



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

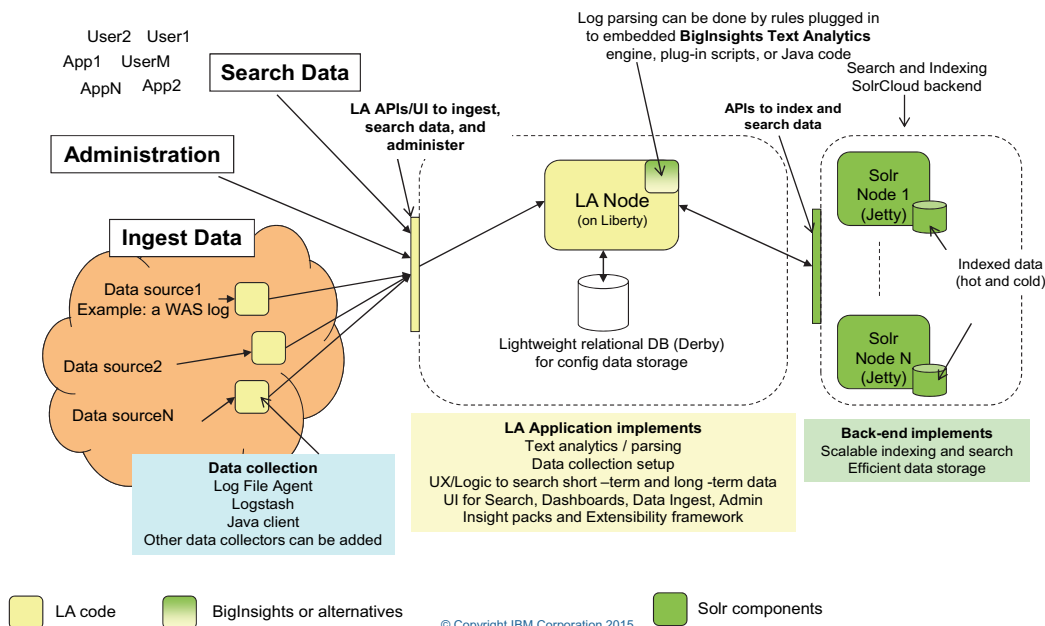
This unit explains the application log files that you should inspect to troubleshoot problems with IBM Operations Analytics Log Analysis.

Objectives

In this unit, you learn to perform the following tasks:

- Find key log files
- Use log files to troubleshoot common problems

Log Analysis components



3

Log Analysis components

This diagram shows the main IBM Operations Analytics Log Analysis applications. These applications are grouped by their general function: Data ingestion, user search, and so on. The following list is a map of each function and the log file where you find detailed information about the operation of the associated application:

- Administration and search: `UnityApplication.log`
- Data ingestion: `GenericReceiver.log`
- Streaming data sources that are managed by the Log File Agent: `UnityEifReceiver.log`
- Batch uploading of logs with Java client: `DataCollectorClient.log`
- Dashboards: `UnityApplication.log`

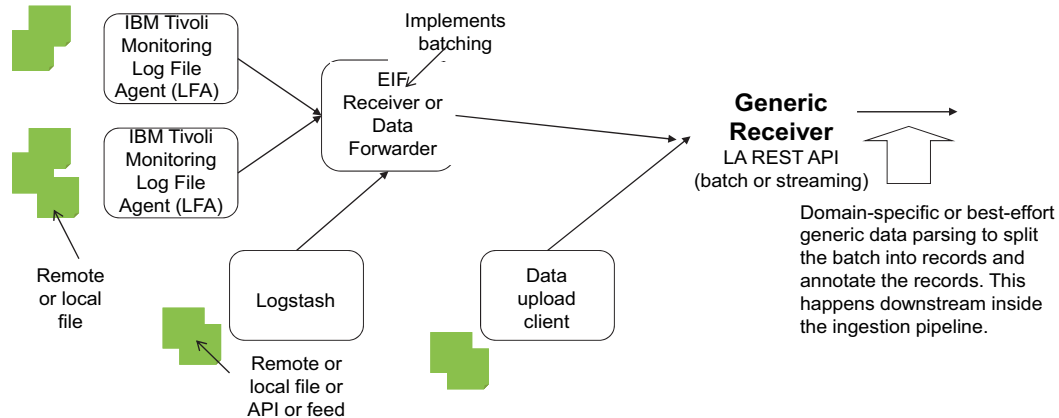
These log files are in the `<LA_HOME>/logs` directory:

- `UnityApplication.log`
- `GenericReceiver.log`
- `UnityEifReceiver.log`

The `DataCollectorClient.log` file is in the

`<LA_HOME>/utilities/datacollector-client/logs` directory.

Log Analysis data ingestion components



© Copyright IBM Corporation 2015

4

Log Analysis data ingestion components

All incoming data is sent to IBM Operations Analytics Log Analysis through the Generic Receiver process. The Generic Receiver receives incoming data through a REST API.

For data collection, there are several IBM Operations Analytics Log Analysis components that receive log data and send it to the Generic Receiver:

- **IBM Tivoli Monitoring Log File Agent (LFA):** This component captures streaming messages from a live log and sends them to IBM Operations Analytics Log Analysis. The LFA sends messages directly to the EIF Receiver.
- **Event Integration Facility (EIF) Receiver:** This component accepts messages from the LFA and sends them to the Generic Receiver in JSON POST requests.
- **Java data collector client:** This component is a Java data collector client that installs on the same host as IBM Operations Analytics Log Analysis. This component uploads historical log files.
- **Logstash:** This component is included with IBM Operations Analytics Log Analysis, but it is not installed by default. You can use this component to transport, parse, and annotate log messages.

You can also create and use custom data collectors in your environment.

Log configuration

- `UnityApplication.log` and `GenericReceiver.log` files are configured in the file
`<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/log4j.properties`
- The `EIFReceiver` log file is configured in this file:
`<LA_HOME>/UnityEIFReceiver/jars/log4j.properties`

© Copyright IBM Corporation 2016

5

Log configuration

Changing the UnityApplication.log and GenericReceiver.log log level

The `UnityApplication.log` and `GenericReceiver.log` logs are both configured in the same file. To increase the logging level of these logs, edit the

`<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/log4j.properties` file.

To increase the logging level of the `UnityApplication.log` file, find the following line and change the value of `INFO` to `DEBUG`.

```
log4j.logger.Unity=INFO,UNITY_FILE
```

To increase the logging level of the `GenericReceiver.log` file, find the following line and change the value of `INFO` to `DEBUG`.

```
log4j.logger.UnityGenericReceiver=INFO,UNITY_GR_FILE
```

Changing the UnityEifReceiver.log log level

The `UnityEifReceiver.log` is configured in the `<LA_HOME>/UnityEIFReceiver/jars/log4j.properties` file.

To increase the logging level of the `UnityEifReceiver.log` file, find the following line and change the value of `INFO` to `DEBUG`:

```
log4j.logger.Unity=INFO,UNITY_FILE
```

Example troubleshooting workflow

No log records are found in the user interface:

- Check the `UnityEifReceiver.log` file to verify that the Log File Agent is receiving data
- Check the `GenericReceiver.log` file to verify that the Generic Receiver has accepted data
- If there is no indication that data is received, check the `unity_data_collection_setup_<date>.log` file for configuration errors
- If you find no configuration problems, look for errors in `scloganalytics_install.log`
- If all of these logs are without errors, check the `UnityApplication.log` file

Example troubleshooting workflow

This example of a troubleshooting scenario shows how you can use the application log files to find the root cause of the problem.

1. Open the `UnityEifReceiver.log` file. If no relevant records are shown in this log, the IBM Tivoli Monitoring Log File Agent did not detect the log file in the log source directory, or no Event Integration Facility (EIF) events are being generated. The connection from the IBM Tivoli Monitoring Log File Agent to the EIF Receiver might not be correctly configured.
2. If the EIF Receiver has recorded EIF events and posted them to the Generic Receiver, review the `GenericReceiver.log` file.
3. The `GenericReceiver.log` file indicates an ERROR if the log source configuration is incorrect. Verify that you have configured the host name and log path to match the host name and log path in the EIF event.
4. If there are no records that indicate an EIF event reported to EIF Receiver, review the `unity_itm_logagent_setup_TIMESTAMP.log` file to confirm that the IBM Tivoli Monitoring Log File Agent setup is correct. This file is an installation log.
5. If the `unity_itm_logagent_setup_TIMESTAMP.log` file contains no issues or errors, review the `scloganalytics_install.log` to determine whether any issues were recorded during installation of any of the other applications.
6. If no issues are recorded in the logs mentioned, review the `UnityApplication.log` file, determine whether there is any issue while querying for indexed data, or with user interface requests, database queries, and so on.

Common problems and resolution, 1 of 2

Installation

- No logs are available if installation is incomplete or fails
- Installation fails if the Log File Agent is already installed
- Silent installation fails after repository location is updated

Errors during data ingestion

- Host name and log path combination is not found
- Time stamp formats are not compatible
- Log File Agent fails to post events
- Exceptions occur during log splitting
- Exceptions occur during annotation; exceptions are in `GenericReceiver.log`
- Some of the log records are not ingested

© Copyright IBM Corporation 2016

7

Common problems and resolution, 1 of 2

Installation problems

Problem: No logs are available after a failed installation

An installation fails with an error message. No logs are available in the `<LA_HOME>/logs` directory. The installer performs a silent installation of other embedded products. If the installation of any component fails, IBM Operations Analytics Log Analysis does not complete the installation. When an installation fails, it provides log files in the directory where the IBM Operations Analytics Log Analysis installation media is saved and not in `<LA_HOME>/logs` directory or in the installation directory. Check this location for more information.

Problem: The installation fails if the IBM Tivoli Monitoring Log File Agent is already installed

The existing IBM Tivoli Monitoring Log File Agent file from a previous installation conflicts with the files that are installed as part of the overall installation process. This condition causes the installer to fail. Use one of the following options to fix this problem:

- Uninstall the existing IBM Tivoli Monitoring Log File Agent.
- Rename the existing IBM Tivoli Monitoring Log File Agent folder.

Problem: The silent installer fails after the repository location is changed

A silent installation fails after the IBM Installation Manager repository is changed. The problem occurs even after you update the response file with the correct repository location. This is because the old repository is still open and connected in IBM Installation Manager. To resolve the problem, remove the old repository from IBM Installation Manager with either the user interface or the console. Then update the response file with the new repository location and repeat the silent installation.

Errors during data ingestion**Problem: The combination of host name and log path is not found**

Change the host name in the Log File Agent configuration to use the short name format. If you are using the Java client, check the `javaDatacollector.properties` file for the format of the host name.

Problem: Time stamp formats are not compatible

Verify that the time stamp format in the index configuration of the data type matches the actual log files.

Problem: The Log File Agent does not post events

The log events that are listed in the log file do not show in the IBM Tivoli Monitoring Log File Agent. To resolve this problem, reconfigure the Log File Agent, select `NO TEMS`, and then restart the agent.

Problem: Exceptions during splitting of logs

If you see log exceptions regarding log message splitting, use the following troubleshooting steps:

1. Check the Insight Pack that you are using for the logs. There might be a mismatch in the log and the type you selected.
2. Test the splitter logic with the Insight Pack tools and a log sample.
3. Verify that the python package required to run the script-based splitter is installed on the IBM Operations Analytics Log Analysis host.

Problem: Exceptions during log annotation in the `GenericReceiver.log` file

If you are using the script-based annotator, make sure that the correct python package is installed on the IBM Operations Analytics Log Analysis host. If you are using the generic annotator, verify that the source type you create has the correct time stamp in the index configuration.

Problem: Some log records are not ingested

You see the following error message in the `GenericReceiver.log` file:

```
CTGLA5133E : Multiple values for DATE field 'timestamp'
```

Verify that the Insight Pack you are using records only one value for the time stamp field. You can test this with the Insight Pack tools and a log sample.

Common problems and resolution, 2 of 2

Administration: There are blank administrative settings in the user interface

No results are returned by a search query

Dashboards fail

- Python libraries
- OAUTH failures

Common problems and resolution, 2 of 2

Administration page problems

If you see a blank administrative settings page, IBM Operations Analytics Log Analysis failed to connect. Restart the product.

Missing results in a search query

If you cannot find data in a search, use the following troubleshooting steps:

1. Check whether the correct log source is selected in the user interface.
2. Check the time filter that you specified in the user interface.
3. Check the query that is specified in the search bar. The search string must conform to the query syntax of Apache Lucene, which is part of Apache Solr.
4. Check the data retention period.

Problems with custom applications and dashboards

If an application or dashboard fails, it generates an error message in the `UnityApplication.log` file. Python `simplejson` libraries are required to run the post processing script custom applications use. Download and install the Python `simplejson` package and install it using its RPM file.

If you see an OAUTH error message in the user interface, check the `/etc/hosts` file of the IBM Operations Analytics Log Analysis host. Verify that the format of the host name is correct.

Solr troubleshooting

Apache Solr writes to the log file:

```
<LA_HOME>/solr-4.7.1/scala_instance1/logs/solr.log
```

The `solr.log` file is useful to find these items:

- Query failures
- Invalid data types (for example, LONG versus TEXT)
- Out-of-memory failures
- Connection failures
- Connectivity failures between Zookeeper and a remote Solr node

Solr troubleshooting

The Apache Solr component indexes new data and processes user queries.

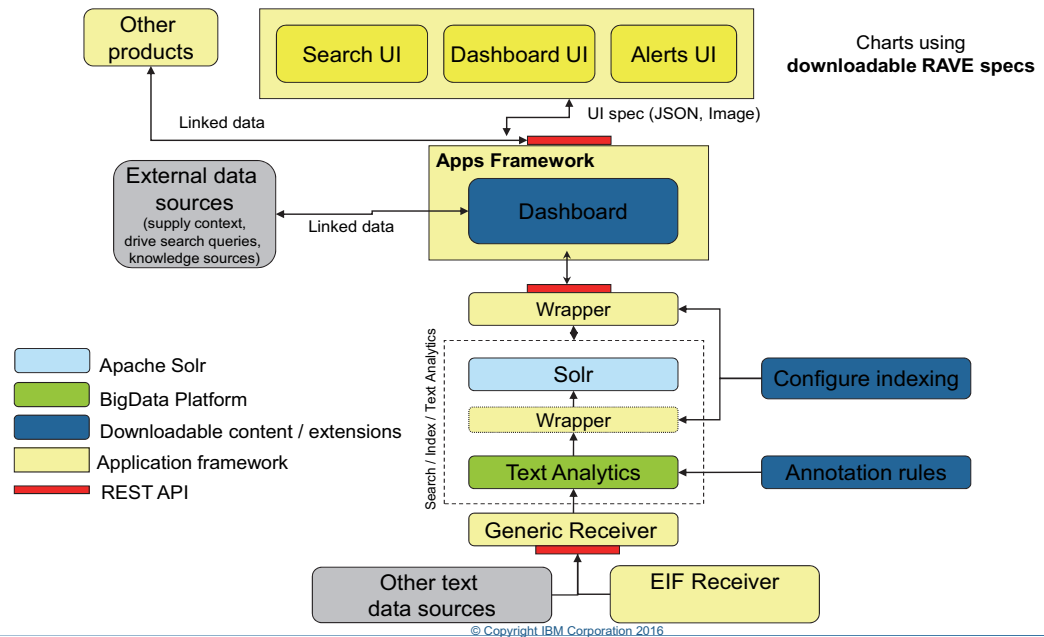
By default, the log level is set to `WARN`. Only warning, error, and fatal messages are in the log, such as data commit errors, socket exceptions, out of memory exceptions.

You can change the level of this log by editing the `<LA_HOME>/solr-4.7.1/scala_instance1/resources/log4j.properties` file. In this example, the log level is set to `WARN`.

```
# Logging level
solr.log=logs/
log4j.rootLogger=WARN, file, CONSOLE
```

When you set the logging level to `INFO`, you see detailed information about every batch that is written and every query that is run.

Interfaces and content



11

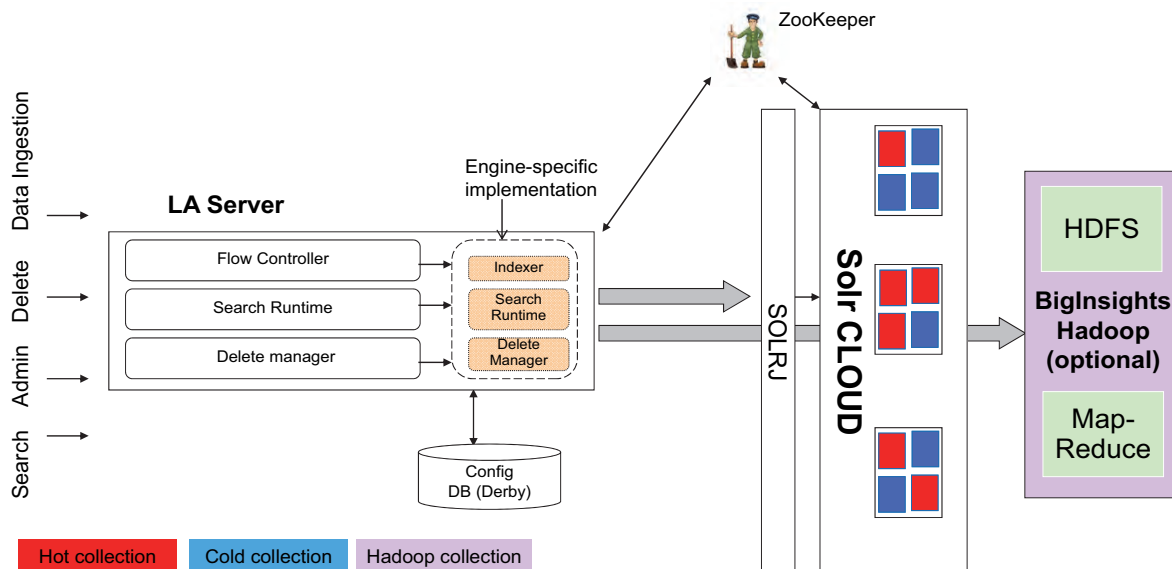
Interfaces and content

IBM Operations Analytics Log Analysis has several API interfaces. One of these is the REST interface of the Generic Receiver. Other interfaces include the query/search API, and the API for dashboards and the custom application framework.

Content is provided by Insight Packs. Insight Packs add parsing and annotation rules to support different log file formats, for example syslog or Apache HTTP access logs. Insight Packs also typically add custom applications and dashboards to the user interface.

If you want to create your own content, IBM Operations Analytics Log Analysis includes tools that you can use to create custom Insight Packs.

Log Analysis architecture



© Copyright IBM Corporation 2016

12

Log Analysis architecture

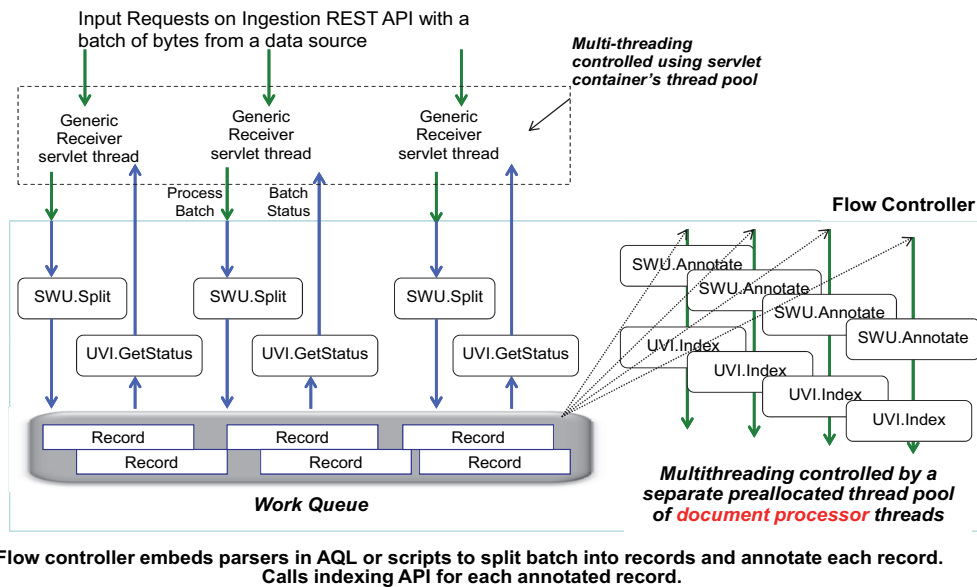
Indexed and raw log data is saved in the file system of the Solr server. Several Solr nodes can be installed in a cluster configuration for faster indexing and query processing.

Solr stores data in two ways: hot tier and cold tier data collections.

- Hot tier collections hold the most recent data in memory, for fast searching.
- Cold tier collections store older data on disk.

Optionally, you can integrate Log Analysis with Hadoop Distributed File System (HDFS) for long-term data storage. This integration is described later in this course.

Ingestion pipeline



13

Ingestion pipeline

This diagram shows two components:

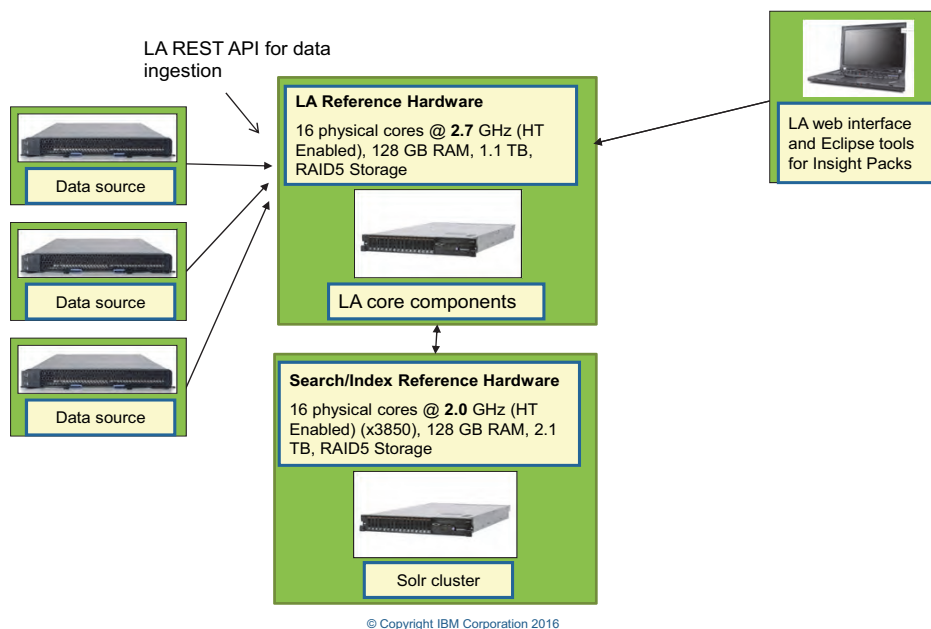
- The Generic Receiver is at the top of the slide.
- The BigInsights text analytics engine is at the bottom of the slide. This text analytics engine is part of the core IBM Operations Analytics Log Analysis software. The BigInsights text analytics engine parses log messages and annotates the fields in each message.

When new messages arrive, the Generic Receiver sends data from log files to the text analytics engine in batches. Because the Generic Receiver is multi-threaded, data is sent in parallel to the text analytics engine.

The text analytics engine uses Annotation Query Language (AQL) rules or scripts to split the log data in to individual messages. After the log records are split, the text analytics engine uses different AQL rules or scripts to annotate the fields in each message.

The text analytics engine sends annotated log messages to the Solr cluster to be indexed. The Solr cluster is not in this diagram.

Example deployment architecture



14

Example deployment architecture

This diagram is an example of an IBM Operations Analytics Log Analysis environment that is distributed over multiple hosts.

Data sources

The data sources in this slide represent any application or component that sends log data to the core IBM Operations Analytics Log Analysis application. Examples of data sources are Log File Agents (LFA), Logstash sources, and custom data collectors and forwarders. Data sources send data to the core log analysis application through the REST interface of the Generic Receiver. Data source hosts are typically deployed close to the source of the text, such as a centralized syslog server, or in a data center.

IBM Operations Analytics Log Analysis core

The IBM Operations Analytics Log Analysis core components process the following tasks:

- Text analytics and parsing
- Data collection setup
- User interface logic to search short and long-term data
- User interface for search, dashboards, data ingestion, and administration
- Insight packs and an extensibility framework

After the log analysis core components split and annotate text from the data sources, it sends data to the Solr cluster to be indexed and stored for user queries. The log analysis core components are typically hosted in a data center.

Solr cluster

The Solr cluster indexes and stores data for user searches. The data is organized into multiple pieces, or *shards*, in the SolrCloud cluster. These shards can be hosted on multiple computers for scalability and efficient data storage.

Client and user hosts

Users access the search interface with a local web browser. Log analysis administrators also use a local web browser to access the administrator user interface.

Log analysis implementers and administrators might have to create custom Insight Packs. You can download and install Eclipse-based toolkits on your local computer that help you to create Insight Packs.

Summary

You now should be able to perform the following tasks:

- Find key log files
- Use log files to troubleshoot common problems

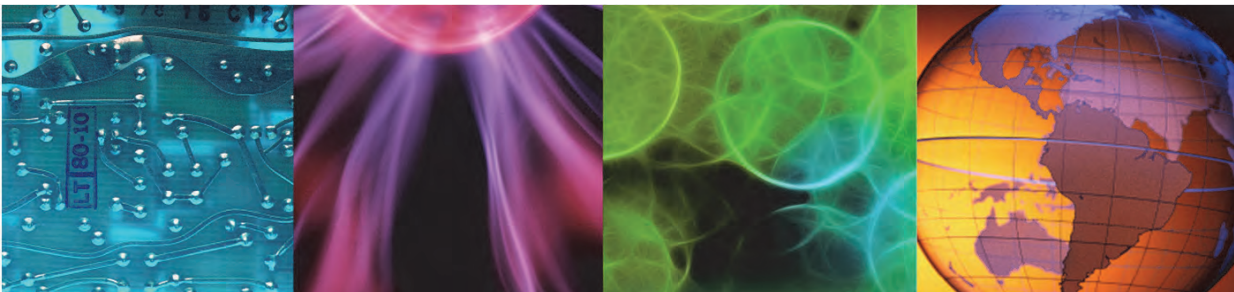
Summary



4 Alerts



4 Alerts



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches you how to use the alerts feature of IBM Operations Analytics Log Analysis. You learn how to create *conditions* that detect text patterns and *actions* that generate notifications when those conditions are met.

Objectives

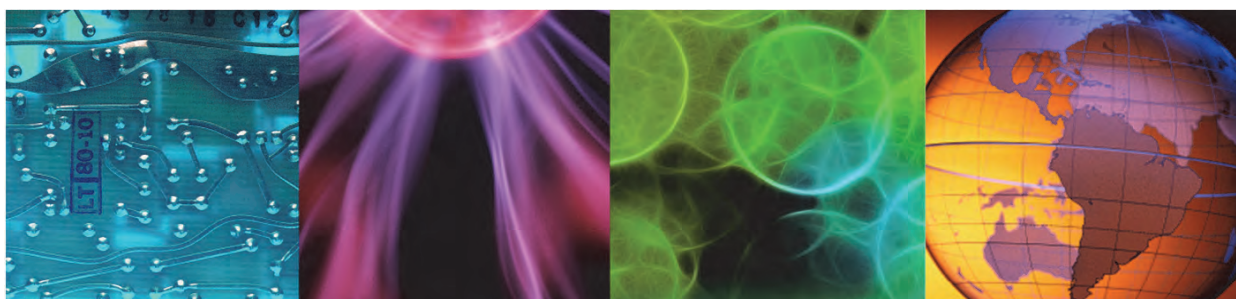
In this unit, you learn to perform the following tasks:

- Create alert actions
- Create base conditions
- Create composite conditions

Lesson 1 Overview



Lesson 1 Overview



© Copyright IBM Corporation 2016

3

This lesson is a brief description of the IBM Operations Analytics Log Analysis alerts feature. In this lesson, you learn about how the alerts feature works.

Alerts overview

- Generate real-time alerts when user-specified conditions are detected
- Conditions are evaluated after the annotation phase in the log processing pipeline
- After a condition is detected, one or more actions can be taken
The following actions are included with the product:
 - Email notifications
 - Running custom scripts
 - Logging alerts to a file
 - Indexing the alert for reporting
- Command-line utilities are available to add, update, show, and delete alert configurations
- REST APIs are available for defining custom conditions, actions, and listing alerts

© Copyright IBM Corporation 2016

4

Alerts overview

You use the alerting features of IBM Operations Analytics Log Analysis to monitor real-time data processing and trigger events based on specified conditions.

Alerts are based on *conditions* and *actions*. Conditions detect text patterns in incoming data. Actions send notifications when a pattern is detected by a condition. You configure conditions that trigger actions such as sending an email notification, running a custom script, or logging an alert.

There are two types of conditions: base conditions and composite conditions.

Base conditions operate on a single log record for a single data source. You configure a search pattern in a base condition. The base condition then inspects every record that arrives and queries the text in the records for the text pattern. If the text pattern is found, the condition triggers an action. For example, a base condition can trigger an action if the word `ERROR` is found in the severity field of log record.

Composite conditions aggregate the result of base conditions over a time window. Composite conditions use base conditions as inputs. For example, if multiple log records with the severity of `Error` (detected by a base condition) arrive five times in the same minute, a composite condition can trigger an action.

You can also correlate patterns across multiple data sources with composite conditions. For example, a composite condition can trigger an action if an error from a WebSphere log arrives close to the same time as an error from a DB2 log.

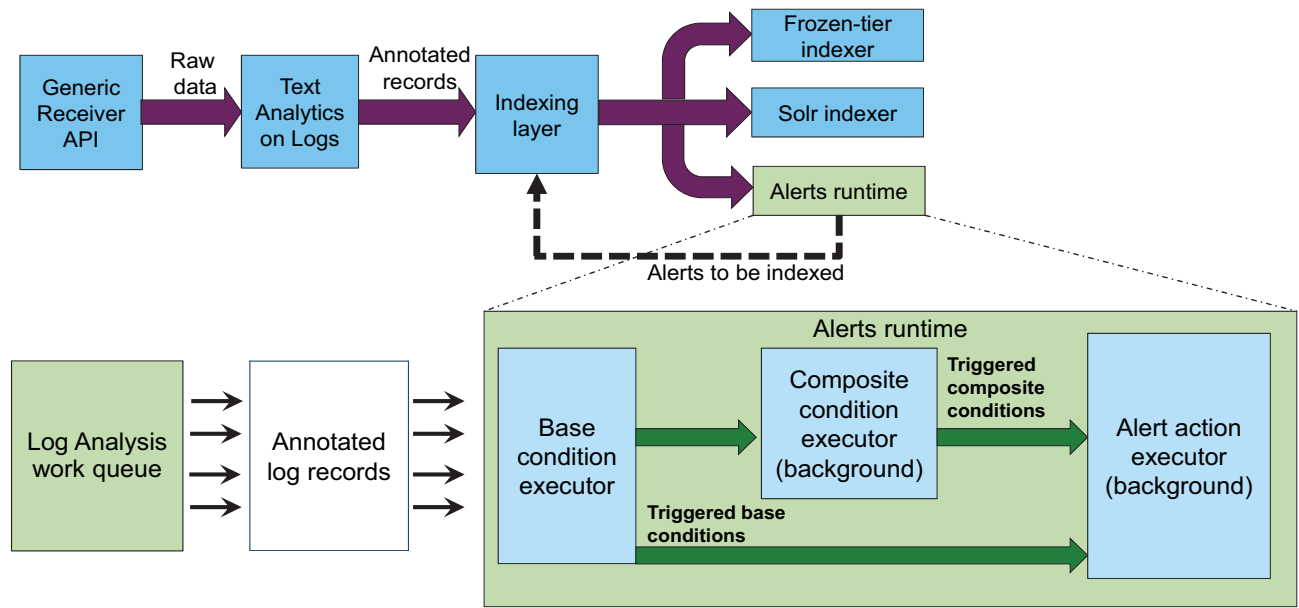
Base conditions look for a pattern in the incoming stream of text; composite conditions look for the trigger counts and frequencies of base conditions.



Note: Currently, you cannot use a composite condition as an input for another composite condition.

There are several actions and conditions that are included with the product. There is also a REST API available to create custom actions and conditions.

Data flow



© Copyright IBM Corporation 2015

5

Data flow

The alerts runtime component of Log Analysis evaluates streaming text after it is split and annotated by the Insight Packs. The indexing layer sends indexed data to Solr for storage and search queries and to the alerts runtime component. The alerts runtime component then detects conditions within the incoming text and runs actions.

The executors within the runtime component evaluates base and composite conditions in real time and runs any triggered actions.

Condition and action templates

- A template provides the actual implementation for a condition or action, for example: the included email action template uses the JavaMail API to send an email alert
- Template implementations are written in Java
- Templates can expose parameters, for example the included email alert action exposes parameters such as SMTP server, from and to addresses, and subject line
- Actual base conditions and actions are instantiated from a template by entering parameter values
For example, you can use the included email action template twice: once to send mail to user_A and again to send mail to user_B

© Copyright IBM Corporation 2016

6

Condition and action templates

A template is the most basic object in the alerts configuration. You use parameters within a template to set the details of the alert, for example, the way an action is run, the text a base condition is searching for, or the time window of a composite condition.

To create an action or condition, edit a set of preset fields in JSON format to suit your environment, then use the template to create an instance of the action or condition. If no template is available for the action or condition that you need, there is a REST API that you can use to create custom actions and conditions.

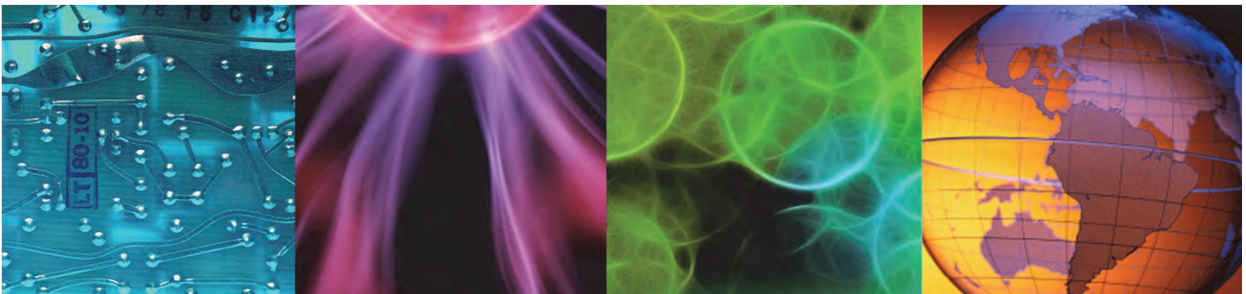


Note: Currently, you must use Java to create all custom templates.

Lesson 2 Included alert actions



Lesson 2 Included alert actions



© Copyright IBM Corporation 2016

7

This lesson explains the three alert actions that are included with the product. You learn how to configure and use these alert actions.

Index alert action

- This action indexes triggered conditions to a fixed data source named `_alerts`
- The index alert action does not accept any parameters and is enabled by default
- The alerts that are saved in the `_alerts` data source contain the following fields:
 - **conditionName**: The name of the condition that was detected in the log
 - **conditionDatasource**: A list of data sources that triggered the condition
 - **conditionRecord**: The log record that triggered the condition (only for base conditions)
 - **Timestamp**: Time stamp of the log record that triggered the alert
- All queries in the user interface can be performed on the `_alerts` data source

Index alert action

The index alert action is configured by default. It does not accept any parameters. You do not have to configure or enable this action.

If a condition is triggered by a log record, the index action writes the details of that log record to a special data source named `_alerts`.

Users can query the `_alerts` data source in the search interface in the same way that they query any other data source. They can search for alerts within a time range, look at facet counts for search results, and create charts and dashboards based on alerts.

The screenshot shows a search interface with a 'Data Sources' dialog box open. The dialog box has a search bar and a 'Find' button. Below the search bar is a list of data sources with checkboxes. The '_alerts' source is checked, and an arrow points to it. Other sources include 'WAS_SystemOut', 'WAS_SystemErr', 'Web_Server', 'ObjectServer-Log', and 'Sar_Memory'. The background shows a search results table with columns 'datasource', 'conditionName', and 'logRecord'.

datasource	conditionName	logRecord
_alerts	WEB-base-condi...	{ "condition
_alerts	WEB-base-condi...	{ "conditionName": "WEB-base-condition", "conditionType": "base", "datasources": ["Web_Server"], "triggeringI
_alerts	WASOUT-base-...	{ "conditionName": "WASOUT-base-condition", "conditionType": "base", "datasources": ["WAS_SystemOut"], "

Email alert action

The name of the template is **emailAlertAction.json**

Make a copy of the template and edit it for your environment

```
"name": "Email action name",
"description": "E-mail action description",
"alertActionTemplateName": "email",
"parameterValues": {
  "smtpMailServer": "mail-server-host-name",
  "secure": false,
  "from": "from@ibm.com",
  "to": ["to@ibm.com"],
  "cc": [],
  "bcc": [],
  "subjectPrefix": "SCALA Alert",
  "header": "Dear User,",
  "footer": "*** This is a system generated e-mail, please do not reply to this e-mail ***\n",
  "attachLogRecordAnnotations": false,
```

© Copyright IBM Corporation 2016

9

Email alert action

The email alert action uses the JavaMail API to send an email when a condition is triggered.

To use the email action, make a copy of the `emailAlertAction.json` template and then edit the copy to suit your environment. The `emailAlertAction.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your action in the `name` field. You use this name later when you create conditions. The `alertActionTemplateName` field must be set to **email**. Use the `parameterValues` fields to specify the details of the email message, such as the SMTP server that sends the email and the mail recipient.

After you edit your copy of the `emailAlertAction.json` file, you must use the `alerts.sh` utility to create the action.

Log alert action

The name of the template is **logAlertAction.json**

Make a copy of the template and edit it for your environment

```
{
  "name": "Log action name",
  "description": "Log action description",
  "alertActionTemplateName": "log",
  "parameterValues": {
    "filePath": "/tmp/alert.log"
  }
}
```

© Copyright IBM Corporation 2016

10

Log alert action

The log alert action buffers alerts that are triggered by conditions and appends the details of the alerts to a log file every 10 seconds.

To use the log action, make a copy of the `logAlertAction.json` template and then edit the copy to suit your environment. The `logAlertAction.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your action in the `name` field. You use this name later when you create conditions. The `alertActionTemplateName` field must be set to **log**. Use the `filePath` field to set the path and file name where you want to log the alert details. The user who runs Log Analysis must have permissions to write to the directory you enter here.

After you edit your copy of the `logAlertAction.json` file, you must use the `alerts.sh` utility to create the action.

Script alert action

The name of the template is **scriptAlertAction.json**

Make a copy of the template and edit it for your environment

```
{
  "name": "Script action name",
  "description": "Script action description",
  "alertActionTemplateName": "script",
  "parameterValues": {
    "scriptPath": "/path/to/script",
    "commandLineParameters": [],
    "workingDirectory":""
  }
}
```

© Copyright IBM Corporation 2016

11

Script alert action

The script alert action runs an external script when a condition is triggered. It passes details of the alert to the standard input (stdin) of the script, which you can then use in your script. Every time the action is invoked, the script runs.

To use the script action, make a copy of the `scriptAlertAction.json` template and then edit the copy to suit your environment. The `scriptAlertAction.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your action in the `name` field. You use this name later when you create conditions. The `alertActionTemplateName` field must be set to **script**. Use the `parameterValues` fields to set the details of your script, such as the location of the script, any required command-line parameters, and the working directory.

After you edit your copy of the `scriptAlertAction.json` file, you must use the `alerts.sh` utility to create the action.

This is the output that is passed to the stdin of your script. It is in JSON format:

```
{
  "conditionName": "<condition_name>",
  "conditionType": "base or composite",
  "conditionDatasources": [<data_sources>],
  "triggeringInput": "<annotated-record-after-index-config-translation>",
  //only for base conditions
  "timestamp": <long_timestamp_value>,
  "date": "<human_readable_ISO-8601_timestamp>"
}
```

```
"alertDetails": <condition_specific_alert_output>  
}
```

Working with alert actions

Use the `alerts.sh` utility to work with alert actions, for example:

`alerts.sh -createAlertAction <FILE_NAME>.json` creates the action that is configured in the file

`alerts.sh -deleteAlertAction <ACTION_NAME>` deletes an alert action

`alerts.sh -getAlertAction` lists all alert actions

The `alerts.sh` utility is in the `<LA_HOME>/utilities/alerts/` directory

Working with alert actions

Use the `alerts.sh` utility to create actions. Use the following command-line options to work with actions.

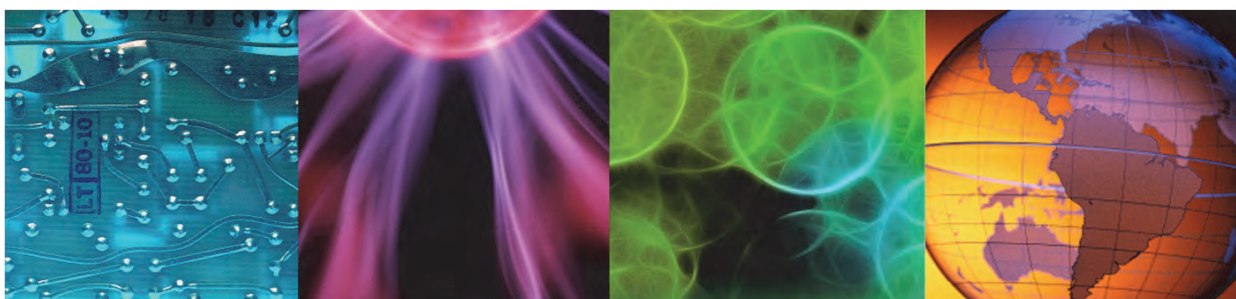
Table 1 Alert action operations

Option	Description
<code>alerts.sh -createAlertAction <file></code>	Create an alert action from a JSON file
<code>alerts.sh -getAlertAction <name></code>	Show all alert actions or a specific action
<code>alerts.sh -deleteAlertAction <name></code>	Delete an alert action by name
<code>alerts.sh -updateAlertAction <file></code>	Update an alert action description and parameters from a JSON file
<code>alerts.sh -enableAlertAction <name></code>	Enable an existing alert action
<code>alerts.sh -disableAlertAction <name></code>	Disable an existing alert action

Lesson 3 Included alert conditions



Lesson 3 Included alert conditions



© Copyright IBM Corporation 2016

13

This lesson explains the four alert conditions that are included with the product. You learn how to configure and use these alert conditions.

Search query base condition

The name of the template is **queryBaseCondition.json**

Make a copy of the template and edit it for your environment

```
{  
  "name": "Base condition name",  
  "description": "Base condition description",  
  "baseConditionTemplateName": "query",  
  "datasourceName": "datasource-name",  
  "parameterValues": { "query" : "query-string"},  
  "actions": ["index","some-other-action"]  
}
```

The query parameter value accepts the Solr query syntax

Search query base condition

The search query base condition runs a query to detect text patterns in every record from a single data source. If text within a record matches the query, the condition is triggered and all actions are run.

To use the query base condition, make a copy of the `queryBaseCondition.json` template and then edit the copy to suit your environment. The `queryBaseCondition.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your condition in the `name` field. The `baseConditionTemplateName` field must be set to **query**. Use the `datasourceName` field to set the data source that you want your condition to apply to.

Replace `query-string` in the `parameterValues` field with the pattern you want to search for. Use the Solr query syntax to format your query. The query string that you enter here is exactly like the query string that you use when searching in the user interface.

Enter the name of the alert actions you want to run if this condition is triggered in the `actions` field.

After you edit your copy of the `queryBaseCondition.json` file, you must use the `alerts.sh` utility to create the condition.

In the following example, the base condition searches for the text `E` or `W` in the `severity` field. The condition queries a data source named `datasource1`. If an `E` or `W` is found in the `severity` field, the actions named `log-triggered-conditions`, `email-user1`, and `index` are run.

```
{  
  "name": "datasource1-severity-base-condition",
```

```
"description": "Base condition for datasource1 severity values",  
"baseConditionTemplateName": "query",  
"datasourceName": "datasource1",  
"parameterValues": { "query" : "severity:E OR severity:W"},  
"actions": ["log-triggered-conditions", "email-user1", "index"]  
}
```

Working with base conditions

Use the `alerts.sh` utility to work with conditions, for example:

`alerts.sh -createBaseCondition <FILE_NAME>.json` creates the condition that is configured in the file

`alerts.sh -deleteBaseCondition <ACTION_NAME>` deletes a base condition

`alerts.sh -getBaseCondition` lists all base conditions

The `alerts.sh` utility is in the `<LA_HOME>/utilities/alerts/` directory

Working with base conditions

Use the `alerts.sh` utility to create base conditions. Use the following command-line options to work with base conditions.

Table 2 Base condition operations

Option	Description
<code>alerts.sh -createBaseCondition <file></code>	Create a base condition from a JSON file
<code>alerts.sh -getBaseCondition <name></code>	Show all base conditions or a specific base condition
<code>alerts.sh -deleteBaseCondition <name></code>	Delete a base condition by name
<code>alerts.sh updateBaseCondition <file></code>	Update the description and parameters of a base condition from a JSON file
<code>alerts.sh -addActionToBaseCondition <condition> <action></code>	Add an action to a base condition
<code>alerts.sh -removeActionFromBaseCondition <condition> <action></code>	Remove an action from a base condition
<code>alerts.sh -enableBaseCondition <name></code>	Enable a base condition
<code>alerts.sh -disableBaseCondition <name></code>	Disable a base condition

Single condition count composite condition

This composite condition counts the number of times a base condition is triggered within a time period

For example, this condition creates an alert when the base condition named **input-condition-example** triggers four times in 60 seconds

The name of the template is **singleConditionCount.json**

Make a copy of the template and edit it for your environment

```
"name": "Condition name",
"description": "Condition description",
"compositeConditionTemplateName": "single-condition-count",
"inputConditions": ["input-condition-example"],
"parameterValues": { "windowDuration" : "60s", "threshold": 4},
"actions": ["index", "some-other-action"]
```

© Copyright IBM Corporation 2016

16

Single condition count composite condition

This composite condition takes input from a single base condition. It counts the number of times the base condition is triggered within a sliding time window. In this example, the composite condition runs all actions if the base condition named `input-condition-example` triggers four or more times in any 60-second window.

To use the single condition count composite condition, make a copy of the `singleConditionCount.json` template and then edit the copy to suit your environment. The `singleConditionCount.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your condition in the `name` field. The `compositeConditionTemplateName` field must be set to **single-condition-count**. Enter the base condition that you want to use in the `inputConditions` field. You can use only one base condition.

Enter the sliding time window in the `windowDuration` field. This field accepts values such as `60s`, `5m`, or `6h`. Enter the base condition trigger count in the `threshold` field. Enter the name of the alert actions you want to run if this condition is triggered in the `actions` field.

After you edit your copy of the `singleConditionCount.json` file, you must use the `alerts.sh` utility to create the condition.

Single condition deduplication composite condition

This composite condition deduplicates multiple alerts that were triggered by a base condition within a time period

For example, this condition creates only **one** alert if the base condition named **input-condition-example** triggers multiple times within a 5-minute interval

The name of the template is **singleConditionDedup.json**

Make a copy of the template and edit it for your environment

```
"name": "Condition name",
"description": "Condition description",
"compositeConditionTemplateName": "single-condition-dedup",
"inputConditions": ["input-condition-example"],
"parameterValues": { "windowDuration" : "5m"},
"actions": ["index", "some-other-action"]
```

© Copyright IBM Corporation 2016

17

Single condition deduplication composite condition

You use this composite condition to suppress multiple alerts from a base condition. In this example, if the base condition named `input-condition-example` triggers any number of times within any 5-minute window, only one alert is created. This composite condition is useful if you want to see only one alert from a base condition, instead of a flood of many alerts.

To use the single condition deduplication composite condition, make a copy of the `singleConditionDedup.json` template and then edit the copy to suit your environment. The `singleConditionDedup.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your condition in the `name` field. The `compositeConditionTemplateName` field must be set to **single-condition-dedup**. Enter the base condition that you want to use in the `inputConditions` field. You can use only one base condition.

Enter the sliding time window in the `windowDuration` field. This field accepts values such as `60s`, `5m`, or `6h`. Enter the name of the alert actions you want to run if this condition is triggered in the `actions` field.

After you edit your copy of the `singleConditionDedup.json` file, you must use the `alerts.sh` utility to create the condition.

Multiple base condition

This composite condition creates an alert when multiple base conditions are triggered over a time period

In this example, the condition creates an alert when base conditions **input-condition-A** and **input-condition-B** are triggered within a 60-second interval

The name of the template is **multiConditionWindow.json**

Make a copy of the template and edit it for your environment

```
"name": "Condition name",
"description": "Condition description",
"compositeConditionTemplateName": "multi-condition-window",
"inputConditions": ["input-condition-A", "input-condition-B"],
"parameterValues": { "windowDuration" : "60s"},
"actions": ["index", "some-other-action"]
```

© Copyright IBM Corporation 2016

18

Multiple base condition

This composite condition takes input from two or more base conditions. In this example, if **all** of the base conditions in the `inputConditions` field trigger within any 60-second window, then the composite condition runs all actions. This composite condition is useful if you want to correlate text patterns and alerts across multiple data sources.

To use the single condition deduplication composite condition, make a copy of the `multiConditionWindow.json` template and then edit the copy to suit your environment. The `multiConditionWindow.json` file is in the `<LA_HOME>/utilities/alerts/` directory.

Enter a unique name for your condition in the `name` field. The `compositeConditionTemplateName` field must be set to **multi-condition-window**. Enter the base conditions that you want to use in the `inputConditions` field. You can use two or more base conditions.

Enter the sliding time window in the `windowDuration` field. This field accepts values such as `60s`, `5m`, or `6h`. Enter the name of the alert actions you want to run if this condition is triggered in the `actions` field.

After you edit your copy of the `multiConditionWindow.json` file, you must use the `alerts.sh` utility to create the condition.

Working with composite conditions

Use the `alerts.sh` utility to work with composite conditions, for example:

```
alerts.sh -createCompositeCondition <FILE_NAME>.json creates the condition that is configured in the file
```

```
alerts.sh -deleteCompositeCondition <ACTION_NAME> deletes a composite condition
```

```
alerts.sh -getCompositeCondition lists all composite conditions
```

The `alerts.sh` utility is in the `<LA_HOME>/utilities/alerts/` directory

© Copyright IBM Corporation 2016

19

Working with composite conditions

Use the `alerts.sh` utility to create composite conditions. Use the following command-line options to work with composite conditions.

Table 3 Composite condition operations

Option	Description
<code>alerts.sh -createCompositeCondition <file></code>	Create a composite condition from a JSON file
<code>alerts.sh -getCompositeCondition <name></code>	Show all composite conditions or a specific composite condition
<code>alerts.sh -deleteCompositeCondition <name></code>	Delete a composite condition by name
<code>alerts.sh -updateCompositeCondition <file></code>	Update the description and parameters of a composite condition from a JSON file
<code>alerts.sh -addActionToCompositeCondition <condition> <action></code>	Add an action to a composite condition
<code>alerts.sh -removeActionFromCompositeCondition <condition> <action></code>	Remove an action from a composite condition
<code>alerts.sh -enableCompositeCondition <name></code>	Enable a composite condition
<code>alerts.sh -disableCompositeCondition <name></code>	Disable a composite condition

Summary

You now should be able to perform the following tasks:

- Create alert actions
- Create base conditions
- Create composite conditions

Summary

Student exercises



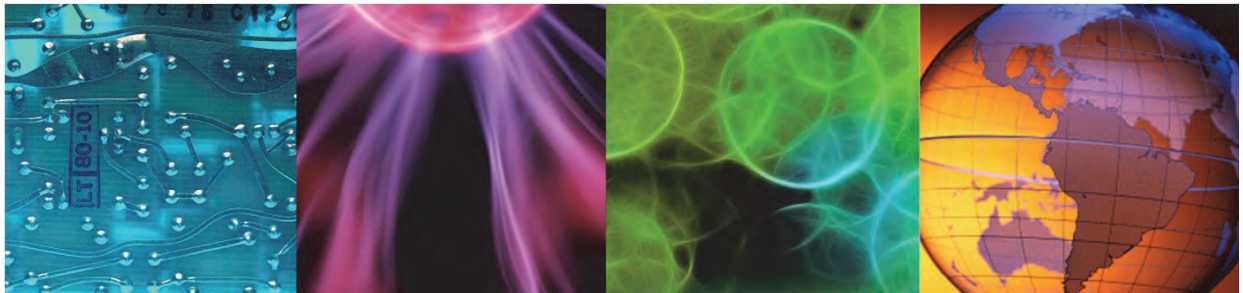
Student exercises



5 Hadoop Distributed File System (HDFS) integration



5 Hadoop Distributed File System (HDFS) integration



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to configure Log Analysis to store indexed data in Hadoop Distributed File System (HDFS) for long-term storage.

Objectives

In this unit, you learn to perform the following tasks:

- Integrate IBM Operations Analytics Log Analysis with Hadoop Distributed File System (HDFS)
- Disable the HDFS integration

Objectives

Integration overview

- Integrating Log Analysis with HDFS provides support for storing compressed, long-term historical data on Hadoop
- Users can then seamlessly search the data in HDFS with the same search interface
- Users can use all features in the search interface, including saved searches and dashboards
These features can combine data from HDFS and non- HDFS data
- There are no changes to existing Insight Packs or tools

Integration overview

Log Analysis gives you the option to store long-term data in HDFS in highly compressed binary format. If you enable the Log Analysis - HDFS integration, users can run queries on the data that is in HDFS from the search interface. There are no changes to the existing Insight Packs with the HDFS integration enabled.

Log Analysis data tiers

Log analysis stores data to three different tiers:

- **Hot tier (Solr search index)**
 - Stores the most recently indexed data
 - A larger portion of data is stored in memory
 - Interactive search: Fastest searches with more memory and processor allocation
- **Cold tier (Solr search index)**
 - Stores a few weeks or months of indexed data
 - Disk-based access, with lower memory utilization than hot tier
 - Data is partitioned by time
 - Incremental search: Fast searches with moderate memory and processor allocation
- **Hadoop tier**
 - Long-term storage of highly compressed data on HDFS
 - Low storage and memory requirements
 - Data is partitioned by type and time
 - Allows you to use historical data for other Hadoop and BigInsights applications, such as these examples:
 - Search/ Reporting
 - Modeling/ Mining
 - Scan-based search: The Hadoop tier searches compressed HDFS files with no indexes; so the search is slower than hot or cold tier

© Copyright IBM Corporation 2016

4

Log Analysis data tiers

Storing and searching data in Solr

By default, Log Analysis stores data in Apache Solr. This setting cannot be changed. Data is organized into two tiers within Solr: hot and cold. New data is saved into the hot tier. As time passes and the data ages, data is moved from the hot tier to the cold tier. The age of data is based on writetime, which is the time a log record was processed by IBM Operations Analytics Log Analysis.

Log Analysis uses more system resources, such as memory and processor time, to search for data in the hot tier. Because cold-tier searches use less memory and less processor time, queries are slower.

In the search results of the user interface, hot-tier data is rendered first, then cold. You learn more about these Solr data tiers later in this course.

Storing and searching data in HDFS

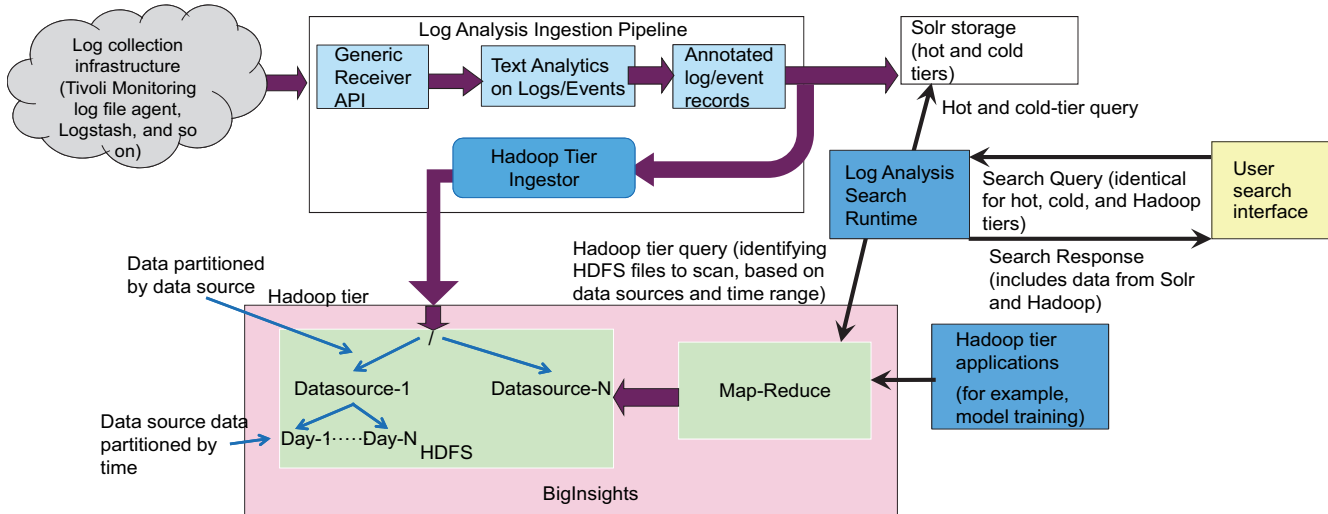
Configuring Log Analysis to store data in HDFS in addition to Solr is optional. Annotated and indexed data that is stored in Solr takes approximately the same amount of space as the volume of data that was processed. For example, if Log Analysis processes 2 GB of log data, it takes roughly 2 GB of disk space in the Solr file system to store it.

To more efficiently save long-term historical data, you might choose to store data in HDFS. Compressed data in HDFS uses far less disk space than data saved in Solr. However, data in

HDFS is not indexed. HDFS uses a scan-based search rather than an indexed search, so search times are slower in HDFS.

The typical use case for the Log Analysis - HDFS integration is to store data that is many months or years old in a highly compressed format. The trade-off for disk space savings is that search times are slower when users query old data sets. In the search results of the user interface, hot-tier data is rendered first (Solr), then cold-tier data (Solr). HDFS data is rendered last.

Data flow



© Copyright IBM Corporation 2015

5

Data flow

At the top of this diagram is the Log Analysis text processing ingestion pipeline. Log Analysis splits the text into individual records and then annotates the meaningful fields within each record. The annotated data is then sent to Solr and BigInsights/HDFS for storage.



Important: If you enable the Log Analysis - HDFS integration, incoming data is stored in both Solr *and* HDFS.

Log Analysis data in HDFS is partitioned into directories based on the data source and the time stamp of the text records.

All search queries from the user interface look for data in both Solr and HDFS, based on the time range in the query. The search runtime component merges the results from Solr and HDFS and renders them in the search results. If identical data exists in both Solr and HDFS, then data is retrieved from Solr.

User search experience

- Users can search data in the Hadoop tier from the Log analysis user interface
- The search experience and workflow is no different from other searches
- Searches are split across Solr and Hadoop, based on the time range in the search query
- Solr queries and Hadoop map-reduce queries run in parallel
- Data from Solr and Hadoop is combined and rendered in the search results
- All Solr results are rendered first; then Hadoop results are incrementally rendered

User search experience

Users do not have to know where data is stored when they use the search interface. Solr and HDFS data are rendered in the same search results. Users can create charts, dashboards, and custom applications with HDFS data, just like with Solr data. It is not apparent to the user where the data is stored.

When a user searches for log messages, Log Analysis queries the Solr and Hadoop file systems. The data in both file systems is combined and displayed in the user search results. If there is identical data in both file systems, Log Analysis shows data only from Solr to the user.

The search work flow does not change, whether the data is retrieved from hot-tier, cold-tier, or HDFS storage. The speed of the search does change. The Interactive (hot-tier) search gets more processor threads and memory than the incremental (cold-tier) search; both of these searches happen in Solr. The Hadoop-tier search happens on highly compressed HDFS files for which there are no indexes; so the search is slower than hot or cold tier.

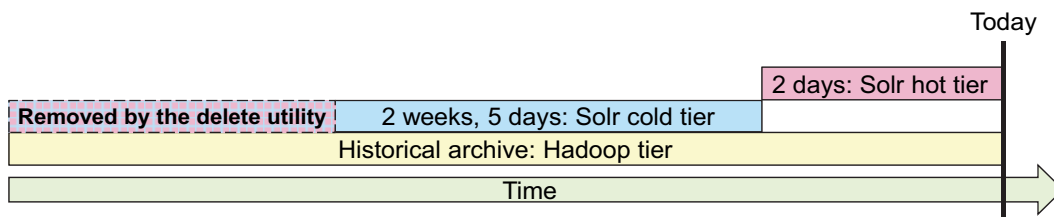
Typical data housekeeping

As data ages, it moves from the Solr hot tier to the Solr cold tier

If you use HDFS to archive older data, you should purge data from the cold tier, for example:

- Data from the most recent 2 days is stored in the hot tier (configurable)
- Data older than 2 days is stored in the cold tier
- The delete utility purges any data from Solr that is older than 3 weeks (configurable)
- Data older than 3 weeks is retrieved from the Hadoop tier if users search for an extended time range

Important: Data automatically ages from the hot tier to the cold tier, but you must configure the delete utility to remove data from the cold tier



© Copyright IBM Corporation 2015

7

Typical data housekeeping

When you integrate Log Analysis with HDFS, all incoming data is written to Solr and HDFS simultaneously. So if you never delete data from Solr, then all historical data is in both Solr and HDFS. Remember when users query for data that is in both Solr and HDFS, only the Solr results are returned. Without regular Solr housekeeping, the data in HDFS is never used.

In this example, Solr is configured to keep data in the hot tier for 2 days. You configure this setting when you install Log Analysis. After 2 days, data is stored in the Solr cold tier. Data stays in the Solr cold tier until you delete it with the `deleteUtility.py` tool. In this example, the `deleteUtility.py` tool is configured to delete data from Solr that is older than 3 weeks.

If a user searches for data from the past 6 months, the first 2 days of search results are retrieved from the Solr hot tier. The next 2 weeks and 5 days of search results are retrieved from the Solr cold tier. Because data older than 3 weeks has been deleted from Solr, the rest of the data in the time range is retrieved from HDFS.

Integration prerequisites

- IBM InfoSphere BigInsights 3.0.x, which includes Hadoop 2.2.
Components required by Log Analysis are HDFS and map-reduce
- Create an operating system user on all of the HDFS data nodes
The user must match the user that owns and runs Log Analysis
- Configure passwordless SSH authentication between each data node server in the cluster
You must set up passwordless SSH in a full mesh configuration for all hosts in the data node cluster, including authentication for the hosts connecting to themselves

© Copyright IBM Corporation 2016

8

Integration prerequisites

Log Analysis only integrates with IBM InfoSphere BigInsights Hadoop as part of IBM InfoSphere BigInsights 3.0.0.1.



Note: You cannot use Apache Hadoop. You must use IBM InfoSphere BigInsights Hadoop.

You must create a user account on all of the HDFS data nodes that matches the user who owns and runs Log Analysis.

You must configure passwordless SSH authentication between hosts for the user that owns Log analysis. You must set up passwordless SSH in a full mesh configuration for all hosts in the data node cluster, including authentication for the hosts connecting to themselves.

While configuring passwordless SSH, use the host names for the data nodes that are specified in the `slaves` file in the Hadoop `config` directory, as in this example:

```
vi /opt/ibm/biginsights/hadoop-conf/slaves
```

```
bivm.ibm.com
```

Configuring BigInsights and Hadoop

- Add the Log Analysis BigInsights ingestion service to each HDFS data node
 - Create a directory for the Log Analysis BigInsights ingestion service on each HDFS data node
 - Decompress the `service.zip` file into the directory that you created.
 - Start the ingestion service with the `server.sh clusterStart` command
- Create directories in HDFS to save Log Analysis data
 - Create a top-level directory in HDFS for, for example, `la-hadoop-tier`
 - Create subdirectories named **data**, **jars**, and **output**, for example:


```
la-hadoop-tier/data
la-hadoop-tier/jars
la-hadoop-tier/output
```
 - Change ownership of these directories to the user who owns Log Analysis
- Copy all `.jar` files to the `la-hadoop-tier/jars` directory
 - These JARs are in the archive files `service.zip` and `search.zip`

© Copyright IBM Corporation 2016

9

Configuring BigInsights and Hadoop

To configure the data ingestion service on each HDFS data node, follow these steps

1. Create a directory on each HDFS data node where the ingestion service runs. The Log Analysis owner must own this directory.
2. Copy the `service.zip` file from the Log Analysis server to the new directory on each HDFS data node and decompress it. You can find the `service.zip` file on the Log Analysis server in the `<LA_HOME>/utilities/hadoop` directory.
3. Start the data ingestion service with the `server.sh clusterStart` command. The `server.sh` utility is included in the `service.zip` file.
4. Create a top-level directory in HDFS to store Log Analysis data. In this example, the top-level directory is `la-hadoop-tier`. Do this on each HDFS data node.
5. Create three subdirectories in HDFS like the following example. Do so on each HDFS data node.


```
la-hadoop-tier/data
la-hadoop-tier/jars
la-hadoop-tier/output
```
6. Grant ownership of the new directories to the Log Analysis user. Do so on each HDFS data node.

7. Copy and decompress the `search.zip` file from the Log Analysis server to a temporary directory on each HDFS data node. You can find the `search.zip` file in the `<LA_HOME>/utilities/hadoop` directory.
8. Copy all JAR files from the `service.zip` and `search.zip` files to the `jars` directory that you created earlier in HDFS, for example `la-hadoop-tier/jars`.

Use the `hadoop fs` command to create the HDFS directories, change ownership, and copy the JAR files. Use commands like the following example:

```
hadoop fs -mkdir /la-hadoop-tier
hadoop fs -mkdir /la-hadoop-tier/data
hadoop fs -mkdir /la-hadoop-tier/jars
hadoop fs -mkdir /la-hadoop-tier/output
hadoop fs -chown -R netcool:ncoadmin /la-hadoop-tier
hadoop fs -copyFromLocal /home/netcool/<SERVICE_JARS_DIR>/lib/*.jar
/la-hadoop-tier/jars/
hadoop fs -copyFromLocal /home/netcool/SEARCH_JARS_DIR/*.jar
/la-hadoop-tier/jars/
```

In this example, the **netcool** user own Log Analysis.

Configuring Log Analysis

- Edit `unitysetup.properties` on the Log Analysis server

Edit the following properties:

```
INDEX_IMPLEMENTATION=SOLR,HADOOP
...
HADOOP_TIER_HDFS_BASE_DIR=hdfs://<NAME_NODE_ADDRESS>:9000/<TOP_LEVEL_LA_DIR>
HADOOP_TIER_HDFS_ADMIN_USER=hdfs
HADOOP_TIER_JOB_TRACKER_URI=<NAME_NODE_ADDRESS>:9001
HADOOP_TIER_SERVER_PORT=9003
```

- Copy the `core-site.xml` and `hdfs-site.xml` configuration files to `<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/`
- Restart Log Analysis

Configuring Log Analysis

To configure the Log Analysis server, follow these steps:

1. Edit the following line in the

`<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file. Append `,HADOOP` to the `INDEX_IMPLEMENTATION` property.

```
INDEX_IMPLEMENTATION=SOLR,HADOOP
```

2. Edit the four `HADOOP_TIER` properties in the `unitysetup.properties` file to suit your environment, for example:

```
HADOOP_TIER_HDFS_BASE_DIR=hdfs://bivm.ibm.com:9000/la-hadoop-tier
HADOOP_TIER_HDFS_ADMIN_USER=hdfs
HADOOP_TIER_JOB_TRACKER_URI=bivm.ibm.com:9001
HADOOP_TIER_SERVER_PORT=9003
```

- a. To find the value of `HADOOP_TIER_HDFS_BASE_DIR`, use the value of the `fs.defaultFS` property in the `core-site.xml` file. This file is in the Hadoop config directory of any HDFS data node, for example:

```
vi /opt/ibm/biginsights/hadoop-conf/core-site.xml
...
<property>
  <!-- The default file system used by Hadoop -->
  <name>fs.defaultFS</name>
```

```
<value>hdfs://bivm.ibm.com:9000</value>
</property>
```

...

Add the top-level directory in HDFS you created earlier to the value of `fs.defaultFS`. In this example, the final value of `HADOOP_TIER_HDFS_BASE_DIR` is `hdfs://bivm.ibm.com:9000/la-hadoop-tier`.



- b. To find the value of `HADOOP_TIER_JOB_TRACKER_URI`, use the value of the `mapreduce.jobtracker.address` property in the `mapred-site.xml` file. This file is in the Hadoop `config` directory of any HDFS data node, for example:

```
vi /opt/ibm/biginsights/hadoop-conf/mapred-site.xml
```

...

```
<property>
  <name>mapreduce.jobtracker.address</name>
  <value>bivm.ibm.com:9001</value>
</property>
```

...

3. Copy the `core-site.xml` and `hdfs-site.xml` configuration files from any HDFS data node to the `<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/` directory on the Log Analysis server. These files are in the Hadoop `config` directory of any HDFS data node, for example:

```
/opt/ibm/biginsights/hadoop-conf/core-site.xml
/opt/ibm/biginsights/hadoop-conf/hdfs-site.xml
```

4. Restart Log Analysis.

Verifying the integration

Use the following methods to verify that the integration is working:

- Process more log messages with Log Analysis
- Check the following log files:
 - <LA_HOME>/logs/Hadooptier.log
 - <LA_HOME>/logs/GenericReceiver.log
 - <LA_HOME>/logs/UnityApplication.log
- Browse to `http://<HDFS_NODE>:50070` and verify that there are compressed binaries files in the `<TOP_LEVEL_LA_DIR>/data` directory
- From the Log Analysis user interface, force a query to the Hadoop tier by searching a data source with the `[_hq]*` filter

Verifying the integration

Use the following log files to verify that Log Analysis is storing data in HDFS.

Hadooptier.log

If Log Analysis is successfully storing data in HDFS, you see messages like the following example in `Hadooptier.log` when data is saved.

```
05/08/15 16:40:37:103 UTC [Thread-46] INFO
com.ibm.tivoli.unity.hadoop.ingestion.client.ServiceNode - Retrieving batches
status from service node [192.168.100.166:9003]
05/08/15 16:40:37:158 UTC [Thread-46] INFO
com.ibm.tivoli.unity.hadoop.ingestion.client.ServiceNode - Received updates for
[649] batches from service node [192.168.100.166:9003]. [0] batches finished with
errors.
05/08/15 16:40:37:158 UTC [Thread-46] INFO
com.ibm.tivoli.unity.hadoop.ingestion.client.ServiceNode - Retrieved batches
status for service node [192.168.100.166:9003]
```

UnityApplication.log

When users search for data in HDFS, you see messages like the following example in UnityApplication.log.

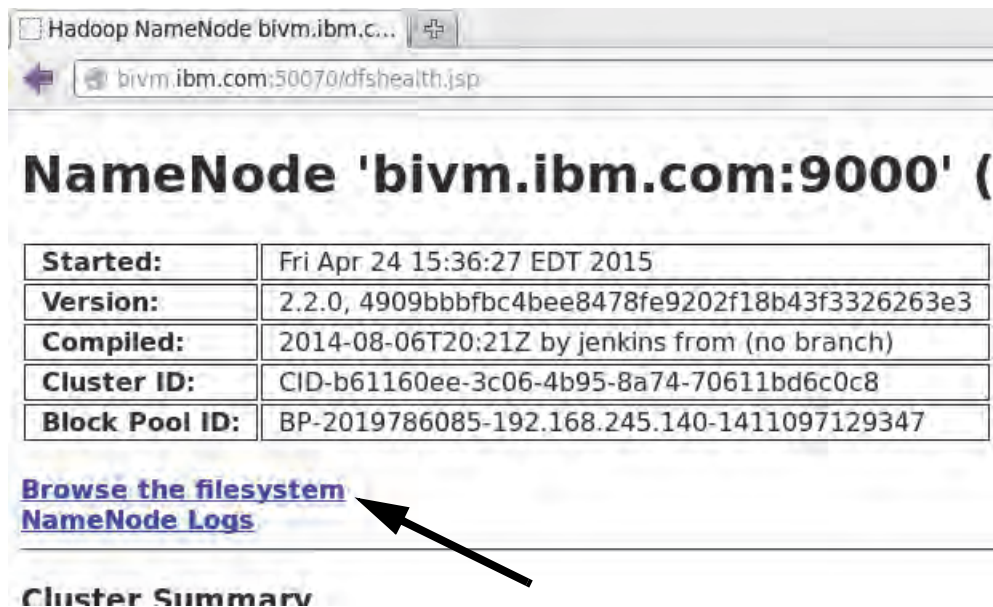
```
04/27/15 15:11:26:718 UTC [Thread-90] INFO - HadoopTierSearchQuery : Hadoop Query
2[1/1], search Time: 35871
04/27/15 15:11:26:718 UTC [Thread-90] INFO - HadoopTierSearchQuery : Hadoop Query
2, total Results: 557
04/27/15 15:11:26:718 UTC [Thread-90] INFO - HadoopTierSearchQuery : Hadoop Query
2, num Results: 90
04/27/15 15:11:26:718 UTC [Thread-90] INFO - SearchQueryRunner : Completed
executing hadoop-tier queries: 2
04/27/15 15:11:26:718 UTC [Thread-90] INFO - SearchQueryRunner : Completed query
execution for hadoop-tier query ID 2, status = COMPLETE
```

GenericReceiver.log

If Log Analysis cannot connect to HDFS, GenericReceiver.log contains error messages.

Viewing HDFS data

You can use a browser to view the HDFS file system and verify that log data is present. Open a browser and enter the host name of any HDFS node, followed by :50700. In this example, the URL is `http://bivm.ibm.com:50070`. Click **Browse the filesystem**.



The screenshot shows a web browser window with the address bar containing `bivm.ibm.com:50070/dfshealth.jsp`. The main content area displays the title **NameNode 'bivm.ibm.com:9000'** followed by a table of status information:

Started:	Fri Apr 24 15:36:27 EDT 2015
Version:	2.2.0, 4909bbbfb4bee8478fe9202f18b43f3326263e3
Compiled:	2014-08-06T20:21Z by jenkins from (no branch)
Cluster ID:	CID-b61160ee-3c06-4b95-8a74-70611bd6c0c8
Block Pool ID:	BP-2019786085-192.168.245.140-1411097129347

Below the table, there are two links: [Browse the filesystem](#) and [NameNode Logs](#). A black arrow points to the [Browse the filesystem](#) link.

At the bottom of the page, the text **Cluster Summary** is visible.

Click the top-level directory that you created for Log Analysis data. In this example, the top-level directory is **/la-hadoop-tier**.

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
biginsights	dir				2014-09-18 23:55	rw-rw-r-x	hdfs	biadmin
hadoop	dir				2014-09-18 23:32	rw-r-xr-x	hdfs	biadmin
hbase	dir				2015-04-24 15:37	rw-r-xr-x	hbase	biadmin
la-hadoop-tier	dir				2015-04-24 16:20	rw-r-xr-x	netcool	ncoadmin
tmp	dir				2014-09-19 00:34	rw-rw-rwt	hdfs	biadmin
user	dir				2014-09-25 20:11	rw-rw-rwx	hdfs	biadmin

Click the **data** directory.

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
data	dir				2015-04-27 09:35	rw-r-xr-x	netcool	ncoadmin
jars	dir				2015-04-24 16:29	rw-r-xr-x	netcool	ncoadmin
output	dir				2015-04-24 16:20	rw-r-xr-x	netcool	ncoadmin

Click the **UnityCollection_<date>_UTC** directory.

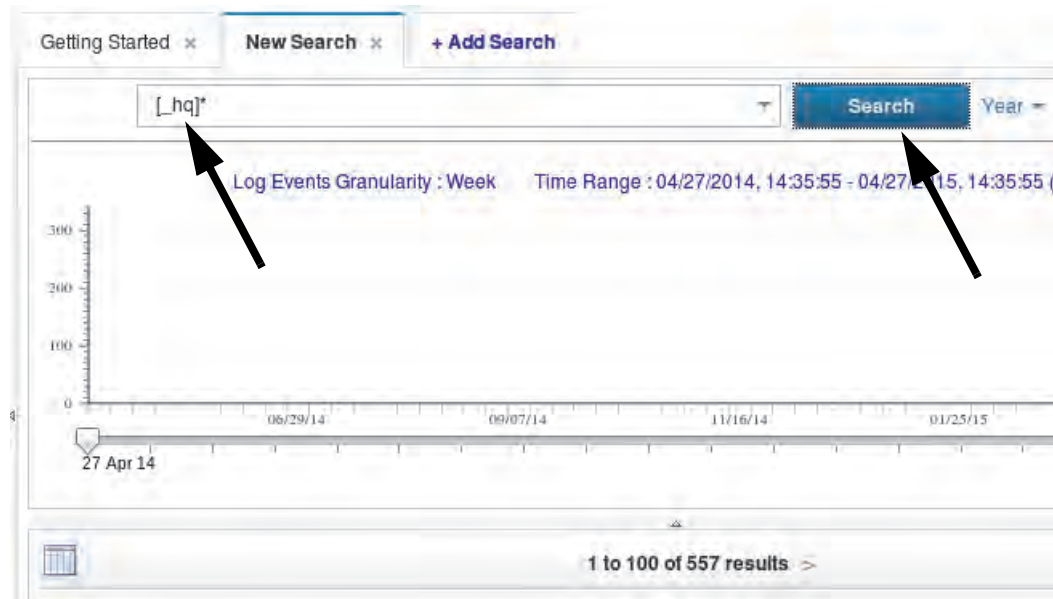
[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Mod
.tmp	dir				2015
UnityCollection_27_04_2015_00_00_00_UTC	dir				2015

From here, you can browse to the subdirectories that store the compressed binary data files.

Querying HDFS data from the user interface

You can force the Log Analysis user interface to search for data only in the HDFS nodes. Use the filter `[_hq]*` in the search interface to force a search for data only in HDFS.



Disabling the integration

- Edit `unitysetup.properties` on the Log Analysis server. Edit the following property so that the only value is `SOLR`

```
INDEX_IMPLEMENTATION=SOLR
```

- Restart Log Analysis

Disabling the integration

To disable the integration with HDFS, follow these steps:

1. Edit the following line in the

`<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file. Remove `,HADOOP` from the `INDEX_IMPLEMENTATION` property, so that the only value is `SOLR`.

```
INDEX_IMPLEMENTATION=SOLR
```

2. Restart Log Analysis.

Summary

You now should be able to perform the following tasks:

- Integrate IBM Operations Analytics Log Analysis with Hadoop File System (HDFS)
- Disable the HDFS integration

Summary

Student exercises

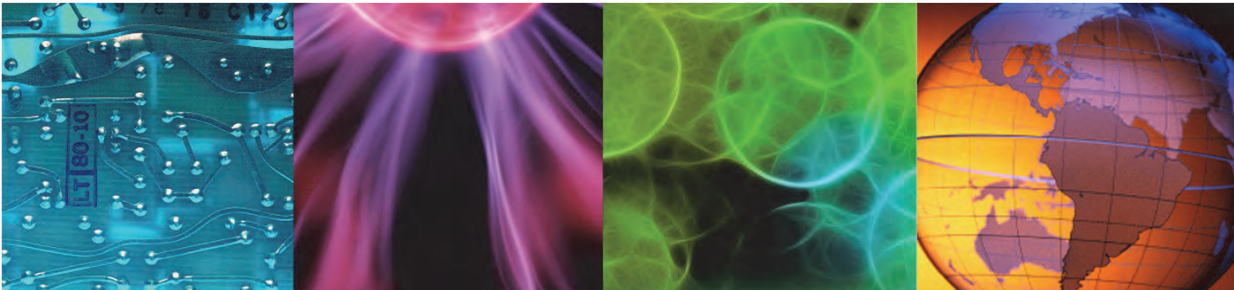




6 Performance tuning



6 Performance tuning



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to change host, operating system, and application settings to tune the performance of IBM Operations Analytics Log Analysis.

Objectives

In this unit, you learn to perform the following tasks:

- Tune host and operating system settings
- Tune application configuration settings

Tuning the server and operating system

- Enable hyperthreading
- Increase the maximum number of processes for the Log Analysis user
 - Increase max users processes in `/etc/security/limits.conf`

```
LAuser hard nproc 4096
LAuser soft nproc 4096
```
 - Increase the max processes for all Log Analysis and Apache Solr hosts

© Copyright IBM Corporation 2016

3

Tuning the server and operating system

IBM Operations Analytics Log Analysis uses multithreading and hyperthreading for much of the processing. Therefore, it is important to enable hyperthreading on your server.

Increase operating system resource limits for the user who owns IBM Operations Analytics Log Analysis. Edit the `/etc/security/limits.conf` file to increase the maximum number of user processes (nproc). Change this setting on all hosts in your IBM Operations Analytics Log Analysis environment.

In the following example, many resource limits for the netcool user have been increased or set, including the maximum number of user processes.

```
...
netcool soft nofile 32768
netcool hard nofile 32768
root soft nofile 32768
root hard nofile 32768
netcool soft memlock unlimited
netcool hard memlock unlimited
root soft memlock unlimited
root hard memlock unlimited
netcool soft as unlimited
netcool hard as unlimited
root soft as unlimited
root hard as unlimited
netcool soft open file 8192
```

```
netcool hard open file 8192  
root soft open file 8192  
root hard open file 8192  
netcool hard nproc 4096  
netcool soft nproc 4096  
root soft nproc 4096  
root hard nproc 4096
```

Reducing the TIME WAIT parameter for socket connections

Decrease the time that must elapse before TCP/IP can release a closed connection and reuse its resources.

- Edit the `/etc/sysctl.conf` file
- Restart the networking service

© Copyright IBM Corporation 2016

4

Reducing the TIME WAIT parameter for socket connections

Open the `/etc/sysctl.conf` file. Add the following line close to the top of the file.

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8) and
# sysctl.conf(5) for more details.
net.ipv4.tcp_fin_timeout=15 ←
# Controls IP packet forwarding
net.ipv4.ip_forward = 0
```

You must restart the networking service after you make the change. Use the following command:

```
service network restart
```

Tuning Java virtual machine (JVM) options

Edit the `jvm.options` file in the `<LA_HOME>/wlp/usr/servers/Unity/` directory

- Ensure that the minimum and maximum heap sizes are the same, for example:

```
-Xms8g
```

```
-Xmx8g
```

- For a Log Analysis WebSphere server with 16 or more physical cores, set the number of garbage collection threads to 6, for example:

```
-Xgcthreads6
```

- Increase the stack size to support long log records, for example:

```
-Xss32M
```

© Copyright IBM Corporation 2016

5

Tuning Java virtual machine (JVM) options

To ensure that the minimum and maximum heap sizes are the same, edit the `<LA_HOME>/wlp/usr/servers/Unity/jvm.options` file. In this example, the minimum and maximum heap size is set to 8 GB.

```
-Xms8g
```

```
-Xmx8g
```

For a WebSphere Liberty Profile server with 16 or more physical cores, set the number of garbage collection threads to 6, for example:

```
-Xgcthreads6
```

To accommodate long log records, increase the size of the frame stack that is used by each Java application thread, for example:

```
-Xss32M
```

After you change the `jvm.options` file, you must restart IBM Operations Analytics Log Analysis.

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

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


Tuning the indexing engine (Solr)

Edit the `unitysetup.properties` file to tune the Solr JVM heap size

This file is in this directory:

```
<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/
```

Edit the `jetty.xml` file to tune Jetty (Java servlet container) maximum idle time size in milliseconds

This file is in this directory:

```
<LA_HOME>/solr-4.7.1/scala_instance1/etc/
```

© Copyright IBM Corporation 2016

6

Tuning the indexing engine (Solr)

Edit the `unitysetup.properties` file to tune the Solr JVM heap size. In this example, the minimum and maximum heap size is set to 8 GB.

```
vi
/opt/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.pr
operties
#Solr Specific properties
#Minimum Java Heap size for running Solr, in MB
MIN_SOLR_HEAP_SIZE=8192
#Maximum Java Heap size for running Solr, in MB
MAX_SOLR_HEAP_SIZE=8192
```

Edit the `jetty.xml` file to tune the Jetty maximum idle time size. In this example, the maximum idle time is 5000 milliseconds.

```
vi /opt/IBM/LogAnalysis/solr-4.7.1/scala_instance1/etc/jetty.xml
<Set name="maxIdleTime">5000</Set>
```

After you change `unitysetup.properties` or `jetty.xml`, you must restart IBM Operations Analytics Log Analysis.

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

Log File Agent (LFA) tuning

For large log files, increase maximum LFA cache file size:

- Edit the appropriate `.conf` file (for example, `WASInsightPack-lfawas.conf`)
These files are in the `<LA_HOME>/IBM-LFA-6.30/config/lo` directory
- Edit the `BufEvtMaxSize` property
- You must delete the appropriate existing `.cache` file
(for example, `<LA_HOME>/logs/lfa-WASInsightPack.cache`)
- You must restart IBM Operations Analytics Log Analysis after you make the change

Log File Agent (LFA) tuning

To increase the increase maximum LFA cache file size, you must edit the LFA configuration file that supports a specific Insight Pack. For example, edit the `WASInsightPack-lfawas.conf` file to change this setting for WebSphere Application Server data sources. These `.conf` files are in the `<LA_HOME>/IBM-LFA-6.30/config/lo` directory.

In the following example, the maximum LFA cache file size is 102400 KB.

```
vi /opt/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf
...
BufEvtMaxSize=102400
```

After you make the change, you must delete the appropriate existing LFA `.cache` file. For example, you would delete the `/opt/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache` file after you change the maximum LFA cache file size for a WebSphere Application Server data source.

After you change the configuration, you must restart IBM Operations Analytics Log Analysis.

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

Configuring EIF receiver buffer size and timeout

Configure the buffer size and timeout period to match the rate of incoming events

Edit:

```
<LA_HOME>/UnityEIFReceiver/config/unity.conf (for local EIF)
```

or

```
<remote_deployment_location>/LogAnalysis/DataForwarders/
EIFReceivers/<eif_inst_#>/config/unity.conf (for remote EIF)
```

```
#Timeout in Seconds
logsource.buffer.wait.timeout=10
#Buffer Size in Bytes
logsource.max.buffer.size=250000
```

© Copyright IBM Corporation 2016

8

Configuring EIF receiver buffer size and timeout

You can configure the EIF receiver buffer size and timeout period to match the rate of incoming events. When the event rate increases, increase the buffer size and decrease the timeout period. The timeout period is the rate at which events are flushed to the generic receiver for indexing. These settings are in the `unity.conf` file. The properties to configure are as follows:

- `logsource.buffer.wait.timeout`
- `logsource.max.buffer.size`

To configure these settings, edit the `unity.conf` file:

- For local EIF Receivers, edit the `<LA_HOME>/UnityEIFReceiver/config/unity.conf` file.
- For remote EIF Receivers, edit the `<remote_deployment_location>/LogAnalysis/DataForwarders/EIFReceivers/<eif_inst_#>/config/unity.conf` file.

In this example, the timeout period is 10 seconds. The maximum buffer size is 250000 Bytes.

```
vi /opt/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf
#Timeout in Seconds
logsource.buffer.wait.timeout=10
#Buffer Size in Bytes
logsource.max.buffer.size=250000
```

After you change the configuration, you must restart IBM Operations Analytics Log Analysis.

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

Use the `eifutil.sh` utility to start and stop remote EIF Receivers.

Solr data tiers

- Hot tier (Solr search index)
 - Holds most recently indexed data (for example, the last 2 days)
 - Higher memory utilization
 - Interactive search
- Cold tier (Solr search index)
 - Holds a few weeks of indexed data
 - Disk-based access for facet queries, lower memory utilization than hot tier
 - Incremental search

© Copyright IBM Corporation 2016

9

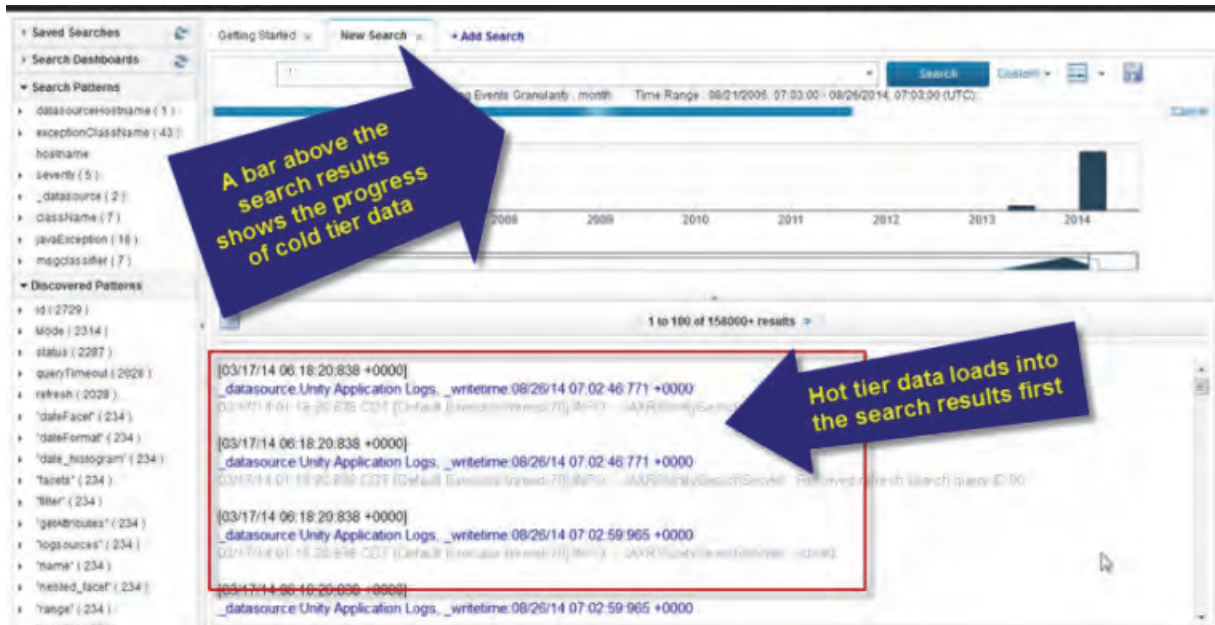
Solr data tiers

Log data is saved to two different tiers within the Solr system. The purpose of this two-tier design is to support retention and search on large data sets that are organized by time.

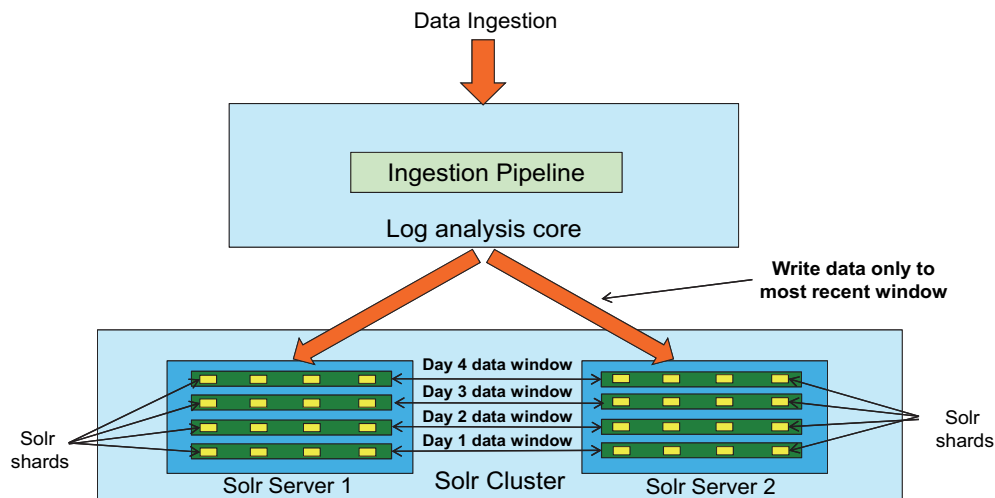
Data is organized into two tiers: hot and cold. New data is saved into the hot tier. As time passes and the data ages, data is moved from the hot tier to the cold tier. Data always moves from hot to cold. The age of data is based on writetime, which is the time a log record was processed by IBM Operations Analytics Log Analysis. By default, data is in the hot tier for 2 days. After the data is 2 days old, it is moved to the cold tier.

Most hot tier data is retained in memory for fast searches. The time that data stays in the hot tier directly affects the memory that is required on the Solr host. A longer hot tier period implies more memory. Cold tier data is stored to disk.

When a user searches for data, hot tier data loads into the search results first (loaded from memory), followed by cold tier data (loaded from disk). A bar above the search results shows the progress of cold-tier data loading into the search results.



Solr data tiers (continued)



© Copyright IBM Corporation 2015

10

Solr data tiers (continued)

The top of this slide shows the log analysis ingestion process. The bottom of the slide shows collections of data in the Solr system.

When new data comes into Solr, it is written to the most recent collection. A collection is a complete logical index in a SolrCloud cluster. Collections are organized by time range window. By default, a single collection holds 1 day of data. At the start of each day, a new collection is created to hold new, incoming data. Solr collections are also called collection windows.

In this diagram, the green bars in the Solr servers represent collection windows. The diagram shows four collection windows for 4 days of data.

Indexed data in each collection window is saved into shards. In this diagram, there are four shards for each collection window, represented by yellow boxes.

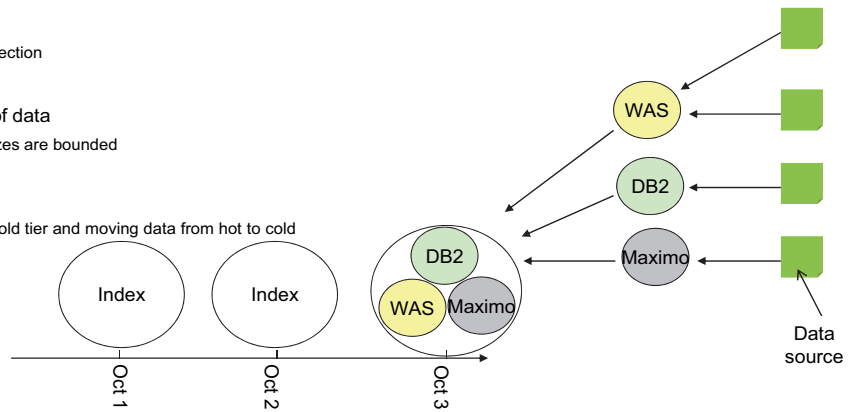
A shard is a logical piece, or slice, of data in the collection. Solr divides incoming data into shards, which saves the actual data to a physical partition. The physical manifestation of a logical shard is called a replica.

Multiple shards can reside on single server. Shards can be distributed across multiple servers in a SolrCloud cluster.

The benefit of saving data across multiple shards is more efficient searches. Running parallel queries across multiple shards lowers query times.

Index data organization

- Each data source is identified by a unique host name and path
 - These attributes might be logical identifiers
- When a data source is set up, these two attributes are configured
 - Each batch of data must contain these two attributes to map to the right data source
- All data sources are mapped to a common Solr collection or index
- Each collection can contain multiple shards
- A collection is time-based partition of data
 - So for specific time period, there is one single collection
 - Interval is configured per installation
- Advantages of having a time-based partition of data
 - Maintains ingestion performance as collections sizes are bounded
 - Easier to delete old data
 - Easier to query newer data first and update UI
 - Finer grain loading and unloading collections for cold tier and moving data from hot to cold



Index data organization

This slide illustrates how data is organized in Solr. Data is organized into time-based partitions.

Data sources are defined by the administrator to start monitoring of a log file. These data sources are identified by the host name and log path that you define when you create a data source. The incoming data from these data sources is saved to a collection, or collection window.

Collections are also called collection windows. These collection windows hold data from all data sources for a time period, for example, 1 day. Collection windows store data in logical partitions called **shards**.

Cold tier storage

Cold tier data is saved to disk, for example:

```
<LA_HOME>/solr-4.7.1/scala_instance1/solr/
UnityCollection_11_11_2014_00_00_00_UTC_shard1_replica1
UnityCollection_11_11_2014_00_00_00_UTC_shard2_replica1
UnityCollection_12_11_2014_00_00_00_UTC_shard1_replica1
UnityCollection_12_11_2014_00_00_00_UTC_shard2_replica1
```

Data is written first to a transaction log, for example:

```
<LA_HOME>/solr-4.7.1/scala_instance1/solr/
UnityCollection_11_11_2014_00_00_00_UTC_shard2_replica1/data/tlog/
```

Cold tier storage

You can view the ingested data in the Solr file system. Ingested data is saved in the `<LA_HOME>/solr-4.7.1/scala_instance1/solr/` directory.

Collection windows hold data for a time period, for example, 1 day. Data within each collection is saved into shards.

In this example, there are 2 days of data: November 11 and November 12. This system is configured to save data across two shards. Each day of data, which is defined by the collection window, stores data into two shards: **shard1** and **shard2**.

There is a `./data/index` subdirectory in each shard file system. This subdirectory holds the actual raw and annotated log data. This data is saved in compressed binary format.

Configuring data tiers

Edit the `unitysetup.properties` file to configure how long data is in the hot tier

This file is in the directory `<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/`

```
COLLECTION_ASYNC_WINDOW = 1d (d,h)
```

...

```
HOT_TIER_PERIOD=2
```

Important: You can configure the hot tier period only immediately after installation; you cannot change this setting later

Configuring data tiers

You can edit the `unitysetup.properties` file to configure these Solr properties:

- The time range of the collection window
- The time that data is held in the hot tier

The full path to the file is as follows:

```
<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties
```

Configuring the collection window size

Edit the `COLLECTION_ASYNC_WINDOW` property to configure the size of the collection window. In the example on this slide, the collection window size is set to 1 day. You can define the window size in days or hours.



Important: The minimum collection window size is 6 hours. If you define the collection window size in hours, you must use a multiple of six.

After you change the configuration, you must restart IBM Operations Analytics Log Analysis.

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

Configuring the hot tier period

Edit the `HOT_TIER_PERIOD` property to configure the amount of time that data is stored in the hot tier. The hot tier period is defined in multiples of the `COLLECTION_ASYNC_WINDOW` property.

In the example on this slide, the hot tier period is set to two. The collection window is 1 day. This setting means that data stays in the hot tier for 2 days. Any data that is older than 2 days is moved to the cold tier.



Important: You can configure the hot tier period only immediately after installation. You cannot change this setting later.

To change the hot tier period, install IBM Operations Analytics Log Analysis and stop all components *before you ingest any data*. Edit the `unitysetup.properties` file to configure the hot tier period, and start log analysis.

Number of shards

Edit the `unitysetup.properties` file to set the number of shards. This file is in this directory: `<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/`

```
INDEX_NUM_SHARDS=2
```

This property should be set to $\frac{1}{2}$ of the number of CPU cores for a dedicated Solr server

Increasing the number of shards allows N parallel operations

Number of shards

You can edit the `unitysetup.properties` file to configure the number of shards that each collection window stores data into.

The full path to the file is as follows:

```
<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties
```

Edit the `INDEX_NUM_SHARDS` property to configure the number of shards per collection.

For a log analysis environment where the Solr server is running on a dedicated host, set the number of shards to one-half of the total number of processor cores. With this configuration, Solr can use half of the available processor cores for indexing data and the other half for search queries.

Solr can index data into multiple shards simultaneously. Increasing the number of shards increases the number of parallel indexing operations.

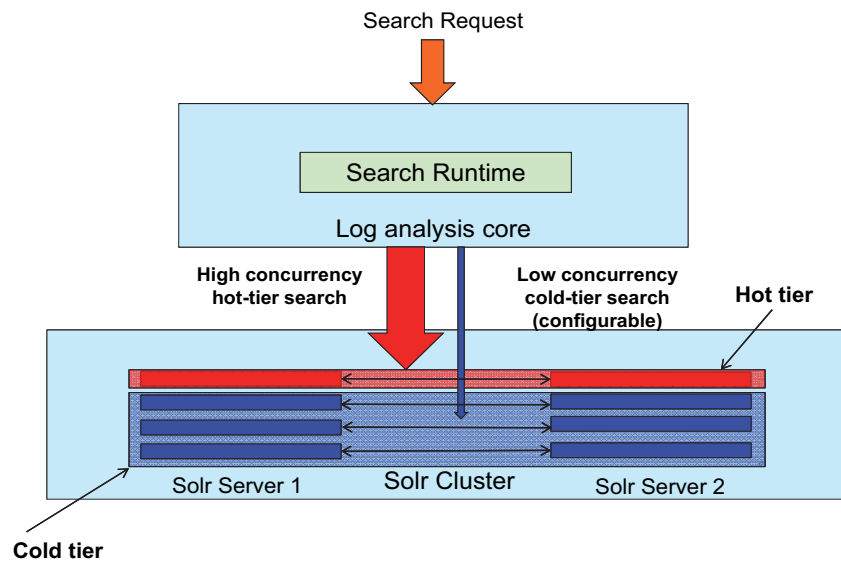
After you change the configuration, you must restart IBM Operations Analytics Log Analysis.

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



Note: The new shard configuration takes effect at the start of the next collection window. For example, if the collection window is 1 day (`COLLECTION_ASYNC_WINDOW = 1d`), you see the new shard directories at the start of the next day.

Concurrent searches



© Copyright IBM Corporation 2015

15

Concurrent searches

The number of allowed concurrent searches is different for hot tier queries and cold tier queries. The number of allowed concurrent searches in hot tier queries is limited by the available number of processor threads.

The number of allowed concurrent searches in cold tier queries is configurable. The default value is one. If you expect a high volume of user queries into cold tier data, you can increase this number.

Configuring concurrent cold tier queries

Edit the **unitysetup.properties** file to set the number of concurrent searches.

This file is in the directory:

```
<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/
```

```
MAX_CONCURRENT_COLD_TIER_QUERIES=1
```

Increase this number if you expect a high fraction of cold tier queries

Configuring concurrent cold tier queries

The full path to the `unitysetup.properties` file is as follows:

```
<LA_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties
```

Edit the `MAX_CONCURRENT_COLD_TIER_QUERIES` property to configure the number of concurrent queries that are allowed to cold tier data.

After you change the configuration, you must restart IBM Operations Analytics Log Analysis:

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

Hardware sizing considerations

- Enable hyperthreading in the IBM Log Analysis and Apache Solr servers
- The underlying storage system should consist of these components:
 - At least eight 300 GB hard disks that support at least 150 input/output operations per second (IOPS) per second
 - RAID 5 is the minimum level required for the Apache Solr file system; RAID 10 is the optimal configuration
- For the Apache Solr file system, use RAID 10 as the RAID configuration
 - RAID 5 is also supported
 - You must use one of these because Apache Solr does not support any of the other RAID levels
- For the Log Analysis server, use RAID 1 or higher, assuming that you install Apache Solr on a separate server
- Network-attached storage is not supported or recommended
- Remotely shared file systems are not recommend for either server
- All external-based file systems should be SAN only
- The underlying IP network must support a minimum of 1GB per second transfer rates

© Copyright IBM Corporation 2016

17

Hardware sizing considerations

This slide lists some general hardware considerations. You can find hardware recommendations and best practices at the IBM Knowledge Center:

<https://www-01.ibm.com/support/knowledgecenter/>

Summary

You now should be able to perform the following tasks:

- Tune host and operating system settings
- Tune application configuration settings

Student exercises



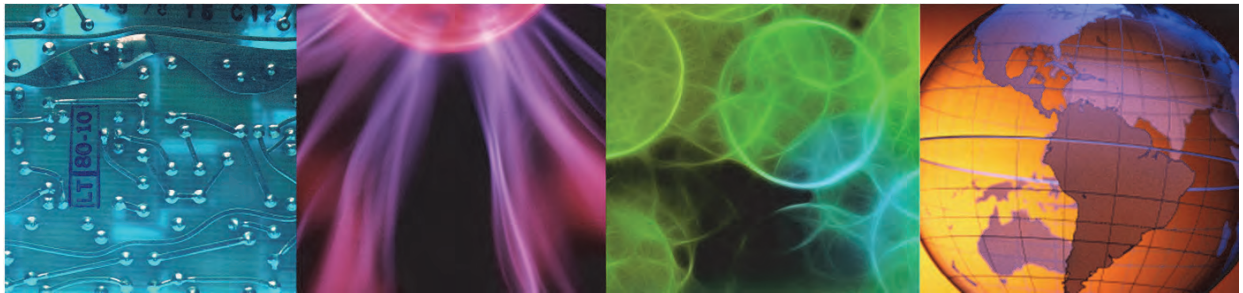
Student exercises



7 Backing up and restoring IBM Operations Analytics Log Analysis



7 Backing up and restoring IBM Operations Analytics Log Analysis



© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to back up IBM Operations Analytics Log Analysis data and restore it on another system.

Objectives

In this unit, you learn to perform the following tasks:

- Back up IBM Operations Analytics Log Analysis
- Restore IBM Operations Analytics Log Analysis

What is backed up

Items backed up and restored

- Ingested and indexed data
- Saved searches, tags, and data sources
- Data types including source types, rule sets, file sets, and collections
- Topology configuration files
- Usage statistics
- LDAP configuration files
- Dashboards
- Insight Packs
- Log File Agent (LFA) configuration files
- License configuration files
- Chart specifications

Items backed up but not restored

- Log files generated by IBM Operations Analytics Log Analysis
- Log files uploaded in batch mode

© Copyright IBM Corporation 2016

3

What is backed up

This slide lists all items that are backed up by the `backup_restore.sh` utility.

Some files that are not required for a new installation are also backed up and maintained for reference purposes. These are log files that are generated by the log analysis components and original log files that were uploaded in batch mode.

Back up and restore prerequisites

- All Log Analysis components must be stopped during back up or restore
- The main Log Analysis server must be able to connect to all remote components
- The source and target installations of Log Analysis must be done by the same operating system user
- Log Analysis should be installed in the same directory on the target host and the source host
- The target system must be a fresh installation of Log Analysis with an identical configuration as the source system
- Backed up data must be restored on a target system immediately after Log Analysis is installed
- Backed up data must not be restored more than once

© Copyright IBM Corporation 2016

4

Back up and restore prerequisites

Log Analysis, including all remote components, must be stopped during backup or restore. Log Analysis must be able to connect to all remote components, with public-key-based or password-based authentication.

The source and target installations of Log Analysis must be done by the same operating system user. The same user should back up and restore.

Log Analysis should be installed in the same directory on the target host (where data is to be restored) and the source host (where the backup is done). The target system must be a fresh installation of log analysis with an identical configuration as the source system. For example, there must be the same number of Solr nodes in the source and target systems.

Backed up data must be restored on a target system immediately after log analysis is installed and before any data is ingested into the system. During data restoration, backed up data is not merged with existing data on the server.

Backed up data must not be restored more than once on a target server. If errors occur restoring backed up data, you must attempt the restoration after removing and reinstalling IBM Operations Analytics Log Analysis on the target system.

Limitations

- New users and changes to passwords for default users are not migrated to the target server
- Custom applications that use Log Analysis passwords in encrypted format will need to be modified to update the password
- Only data contained in the default directories is backed up and restored
- When you restore a system that was extended with extra Insight Packs, the log files and data source directories are not restored

Limitations

New users and changes to passwords for default users are not migrated to the target server. You must modify any custom applications or dashboards that use log analysis passwords in encrypted format to update the password.

Only data that is contained in the IBM Operations Analytics Log Analysis default directories is backed up and restored. Any customization and modifications that are outside of these directories are not backed up or restored.

When you restore a system that was extended with extra Insight Packs, for example, those created with the DSV toolkit, the log files and data source directories are not restored. To resolve this issue, you must manually add these directories. Use the appropriate existing LFA configuration file as a reference.

Backing up Log Analysis

1. Stop all Log Analysis components
2. Change to the backup and restore directory on the source system:
`<LA_HOME>/utilities/migration`
3. Run the back up utility
`./backup_restore.sh <BACKUP_HOME> backup`
`<BACKUP_HOME>` is the directory where you want to save the backup archive files

Backed up data is stored in compressed files

Backing up Log Analysis

Use the `./backup_restore.sh` utility to back up the log analysis environment. In this example, the backup archive files are saved in the `/home/netcool/BACKUP/` directory.

```
cd /opt/IBM/LogAnalysis/utilities/migration

./backup_restore.sh /home/netcool/BACKUP/ backup

INFO: Log Analysis is not running
INFO: Backup home specified: /home/netcool/BACKUP
INFO: Getting status of Solr on host2.tivoli.edu
INFO: Performing backup...
INFO: 349.273 MBs of data to be copied from Log Analysis server and remote Solr
nodes, 27053.973 MBs of disk space available
Do you wish to continue (Y/N)? [N]: Y

INFO: Archiving backup-only data from Log Analysis server
      Done
INFO: Archiving restorable data from Log Analysis server
      Done
INFO: Starting to archive data from Solr nodes
INFO: Backing up Solr index data of SOLR_NODE_LOCAL on host2.tivoli.edu
INFO: Created backup directories on host2.tivoli.edu
INFO: Copied required tools to host2.tivoli.edu
INFO: Archiving data on host2.tivoli.edu
```

Done

INFO: Copying logs from host2.tivoli.edu

INFO: Copying archived data from host2.tivoli.edu

INFO: Finished backing up Solr data of SOLR_NODE_LOCAL on host2.tivoli.edu

INFO: Performing cleanup on host2.tivoli.edu

INFO: Finished archiving data from Solr nodes

INFO: SmartCloud Log Analysis data has been backed up from /opt/IBM/LogAnalysis to /home/netcool/BACKUP as zip files

INFO: Finished performing backup

Restoring Log Analysis

1. Stop all Log Analysis components
2. Copy the backup archive files to the target system
3. Change to the backup and restore directory on the target system:

```
<LA_HOME>/utilities/migration
```

4. Run the restore utility

```
./backup_restore.sh <BACKUP_HOME> restore
```

<BACKUP_HOME> is the directory where you copied backup archive files

Restoring Log Analysis

Use the `./backup_restore.sh` utility to restore up the log analysis environment to the target system. In this example, the backup archive files are in the `/home/netcool/BACKUP/` directory of the target system.

```
cd /opt/IBM/LogAnalysis/utilities/migration
```

```
./backup_restore.sh /home/netcool/BACKUP/ restore
```

```
INFO: Log Analysis is not running
```

```
INFO: Backup home specified: /home/netcool/BACKUP
```

```
INFO: Getting status of Solr on host2.tivoli.edu
```

```
INFO: Performing restore...
```

```
WARN: Before continuing with restoring backed up data, confirm:
```

```
- All archived files created during 'backup' step are present in the
specified backup directory
```

```
- This server is able to connect to all remote Solr nodes if any, with or
without passwords
```

```
- The remote Solr nodes have enough disk space for restoring the backed up
data
```

```
- If the backed up data contains sample scenarios, the same will be restored
```

```
- Restore process may take a long time, depending on amount of backed up data
and must not be interrupted
```

```
Failure to ensure these may lead to partial restoration of data, rendering
the system unusable
```

```
Do you wish to continue (Y/N)? [N]:Y
```

```
INFO: Backed up data version: 1_2_0_3_201409020630_IF0001
INFO: Restoring configuration data for Log Analysis
INFO: Extracting /home/netcool/BACKUP/LogAnalysis_13Nov2014_Restore_001.zip
      Done
INFO: Restoring backed up Solr data on host2.tivoli.edu
INFO: Created backup directories on host2.tivoli.edu
INFO: Copied required tools to host2.tivoli.edu
INFO: Executing restore steps on host2.tivoli.edu
      Done
INFO: Copying logs from host2.tivoli.edu
INFO: Performing cleanup on host2.tivoli.edu
INFO: Finished restoring Solr data on host2.tivoli.edu
INFO: Finished performing restore
INFO: Restore was successful, removing temporary directories
INFO: Configuring properties files...
INFO: Configuring Log File Agent...
INFO: Finished configuring Log File Agent...
```

INFO : SmartCloud Log Analysis data have been restored from the backup in /home/netcool/BACKUP.

Make sure that SmartCloud Log Analysis is running fine before deleting the backed up data. Note that some data (e.g. batch uploaded log files) have been backed up but not restored as they are not needed any more

INFO: Restoring of IBM SmartCloud Analytics Log Analysis data is complete.

Troubleshooting

- Information and errors recorded during back up and restore are stored in log files created in the `logs` subdirectory
- Error messages are displayed in the console if any of the requirements are not satisfied and the tool exits
- If errors occur while restoring backed up data, you must attempt the restoration after removing and reinstalling Log Analysis on the target system

Troubleshooting

Log files that capture messages during the backup and restore process are in the `<LA_HOME>_HOME/utilities/migration/logs` directory. Informational and error messages are also written to the console when you run the backup and restore utility.

Backed up data must not be restored more than once on a target server. If errors occur restoring backed up data, you must attempt the restoration after removing and reinstalling IBM Operations Analytics Log Analysis on the target system.

Summary

You now should be able to perform the following tasks:

- Back up IBM Operations Analytics Log Analysis
- Restore IBM Operations Analytics Log Analysis

