IBM Operations Analytics - Log Analysis
Version 1.3.1

Installation, Configuration, and Administration Guide

IBM

IBM Operations Analytics - Log Analysis
Version 1.3.1

Installation, Configuration, and Administration Guide

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 459.

Edition notice

This edition applies to IBM Operations Analytics - Log Analysis and to all subsequent releases and modifications until otherwise indicated in new editions.

References in content to IBM products, software, programs, services or associated technologies do not imply that they will be available in all countries in which IBM operates. Content, including any plans contained in content, may change at any time at IBM's sole discretion, based on market opportunities or other factors, and is not intended to be a commitment to future content, including product or feature availability, in any way. Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. Please refer to the developerWorks terms of use for more information.

Contents

Notices 459

About this publication

This guide contains information about how to use IBM® Operations Analytics - Log Analysis.

Audience

This publication is for users of the IBM Operations Analytics - Log Analysis product.

Publications

This section provides information about the IBM Operations Analytics - Log Analysis publications. It describes how to access and order publications.

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

http://www.ibm.com/software/globalization/terminology.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. In this release, the IBM Operations Analytics - Log Analysis user interface does not meet all accessibility requirements.

Accessibility features

This information center, and its related publications, are accessibility-enabled. To meet this requirement the user documentation in this information center is provided in HTML and PDF format and descriptive text is provided for all documentation images.

Related accessibility information

You can view the publications for IBM Operations Analytics - Log Analysis in Adobe Portable Document Format (PDF) using the Adobe Reader.

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: http://www.ibm.com/able (opens in a new browser window or tab)

Tivoli technical training

For Tivoli® technical training information, refer to the following IBM Tivoli Education Web site at http://www.ibm.com/software/tivoli/education.

Providing feedback

We appreciate your comments and ask you to submit your feedback to the IBM Operations Analytics - Log Analysis community.

Conventions used in this publication

This publication uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations**:)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents....

`Monospace`

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Installing

This section outlines the procedure that you must follow to install IBM Operations Analytics - Log Analysis. Details of prerequisites, the installation procedure, and additional post-installation tasks are outlined in this section.

Note: You cannot install IBM Operations Analytics - Log Analysis on a Network File System (NFS) based, shared resource.

Note:

If you want to install IBM Operations Analytics - Log Analysis on a system where the locale is not set to English United States, you need to set the locale of the command shell to `export LANG=en_US.UTF-8` before you run any IBM Operations Analytics - Log Analysis scripts. For example, when you install IBM Operations Analytics - Log Analysis you need to run the following command to change the locale before you run the `unity.sh` script:

```
export LANG=en_US.UTF-8
```

Prerequisites

Before you install IBM Operations Analytics - Log Analysis, ensure that the system meets the hardware and software requirements and complete the prerequisite tasks.

Downloading the installation package

Read this document to get an overview of how to download the installation package.

Before you download the installation package, you need to decided which edition you want to install. For more information, see Editions.

To download the Entry edition:

1. Go to the product page at http://www-03.ibm.com/software/products/en/ibm-operations-analytics---log-analysis
2. Click **Entry edition download**.

To download the Standard edition:

1. Go to Passport Advantage at https://www.ibm.com/developerworks/servicemanagement/ioa/log/downloads.html.
2. Choose the type of installation.

   IBM Operations Analytics - Log Analysis for Linux on System x86_64 or Linux on System z®. For more information, see "Installing on Linux on System z and Linux on System x based servers" on page 16.

   To install IBM Operations Analytics - Log Analysis on Linux on System x86_64, download the **IBM Operations Analytics - Log Analysis Standard Edition Linux 64 bit (CN5N9EN)** package.

   To install IBM Operations Analytics - Log Analysis on Linux on System z, download the **IBM Operations Analytics - Log Analysis Standard Edition Linux on System z 64 bit (CN5NAEN)** package.

Create a non-root user

You must use a non-root user to install IBM Operations Analytics - Log Analysis.

Procedure

If you do not have a non-root user defined on your system, you must create one. To create this user, log in as a root user and run the command:

```
useradd -m -d /home/<username> <username>
```

where `-m` creates a home directory for your user if one does not exist, `-d` is the path to the home directory, and *<username>* is the user name that you want to create.
Ensure that you have the necessary access rights so that you can add files to the location where you want to install IBM Operations Analytics - Log Analysis.

Recommended library installers

The following libraries are recommended for use with IBM Operations Analytics - Log Analysis.

The YUM (Yellow dog update, modified) package manager is recommended for installing the required libraries on Red Hat Enterprise Linux (RHEL).

The YaST (Yet another Setup Tool) tool is recommended for installing the required libraries on SUSE Linux Enterprise Server (SLES).

Verifying the operating system version

IBM Operations Analytics - Log Analysis requires Red Hat Enterprise (RHEL) for Linux version 5, 6, or 7 or SUSE Linux Enterprise Server (SLES) version 11.

Procedure

- To verify that the correct version of RHEL, is installed, log in as a root user and enter the following command:

  ```
  cat /etc/redhat-release
  ```

  If the version is not 5, 6, or 7, you need to upgrade to one of these versions.

  Note: RHEL 7.1 is not supported.
- To verify that the correct version of SLES, is installed, log in as a root user and enter the following command:

  ```
  cat /etc/SuSE-release
  ```

  If the version is not 11, you need to upgrade to that version.

Verifying the 64-bit library requirement

For Red Hat Enterprise Linux, IBM Operations Analytics - Log Analysis requires the 64-bit compat-libstdc++ library:

Procedure

1. To determine if the required libraries are installed, run the command:

   ```
   sudo /usr/bin/yum --noplugins list installed "libstdc++"
   ```

   If this command indicates that the required libraries are installed, no additional action is required. For example:

```
libstdc++.x86_64
```

2. If the required libraries are not installed, search the Red Hat Network repository to list the libraries that are available for your operating system:

   ```
   sudo /usr/bin/yum --noplugins search libstdc++
   ```

3. To install the required libraries, run this command:

   ```
   sudo /usr/bin/yum --noplugins install libstdc++.x86_64
   ```

Disabling Security-Enhanced Linux (SELinux)

If SELinux is in enforcing mode, an exception occurs during the installation of IBM Operations Analytics - Log Analysis. Ensure that the SELinux policy is set to a permissive or disabled state. To disable the SELinux:

Procedure

1. Log in as a root user.
2. Edit the `config` file that is in the `/etc/selinux/` directory.
3. Change the `SELINUXTYPE` value to a permissive or disabled state. The possible values are:

   `permissive`
   SELinux prints warnings instead of enforcing them.

   `disabled`
   SELinux is fully disabled.

   For example, to disable the kernel, change the value to `disabled`:

   ```
   SELINUX=disabled
   ```

4. Save your changes.
5. Restart the operating system.

Verifying KornShell library

Verify that the KornShell library is part of your operating system.

Procedure

1. To verify that KornShell is part of your operating system, enter one of the following commands:

   ```
   usr/bin/ksh
   ```

   or

   ```
   /bin/ksh
   ```

2. If these commands do not work, enter the following command to confirm that KornShell is installed:

   ```
   rpm -qa | grep ksh
   ksh-20100202-1.el5
   ```

3. If KornShell is not installed, download it and use the following command to install it:

   ```
   rpm -ivh ksh-<version>.rpm
   ```

   where *<version>* is the version that you downloaded.

Verifying the Python version

Python Version 2.4.3 and 2.6.6 to 2.6.8 are supported by IBM Operations Analytics - Log Analysis.

If you did not use an rpm package to install Python, the IBM Operations Analytics - Log Analysis installer might not recognize it and a warning message might be displayed. This warning can be ignored and the installation continues.

To verify that you are using the correct version of Python, enter the following command:

```
rpm -qa | grep "^python-2"
```

If successful, the command returns the version of Python. For example:

```
python-2.4.3-27.el5
```

If you are not using the correct version of Python, download and install it.

Install `python-simplejson` package for Python

IBM Operations Analytics - Log Analysis requires Python Version 2.4.3 including the `python-simplejson` rpm or Python Version 2.6.6 to 2.6.8. If you use Python Version 2.4.3, you must also install the simple JSON rpm, `python-simplejson`. Python Version 2.6.6 to 2.6.8 include the required rpm.

Procedure

1. Download the `python-simplejson` package from http://pkgs.org/centos-5-rhel-5/centos-rhel-x86_64/python-simplejson-2.0.9-8.el5.x86_64.rpm/download/.
2. Log in to the IBM Operations Analytics - Log Analysis server as a root user.
3. Change directory to the folder that contains the package.
4. Run the following command:

   ```
   rpm -i python-simplejson-2.0.9-8.el5.x86_64.rpm
   ```

Installing unzip utility

You must install the unzip utility on any servers where you install IBM Operations Analytics - Log Analysis or remote Indexing Engine.

Procedure

- To install the utility on Red Hat Enterprise Linux (RHEL), log in as a root user and run the following command:

  ```
  yum install unzip
  ```

- To install the utility on SUSE Enterprise Linux Server (SELS), log in as a root user and run the following command:

  ```
  zypper install unzip
  ```

Verifying the host server IP address and names

Before you install IBM Operations Analytics - Log Analysis, you must ensure that the details for each host server are maintained correctly in the `etc/hosts` directory on the target system.

About this task

If you do not complete this task, you may encounter issues when you log in or run a custom app. For more information, see *Cannot run custom apps after IP address change* in the *Troubleshooting Guide*.

Procedure

- For a server that uses a static IP address, define the static IP address and the required values in the following format:

  ```
  IP     LONG-HOSTNAME    SHORT-HOSTNAME
  ```

  For example:

  ```
  9.124.111.162     scaserver1.example.com     scaserver1
  ```
- For a Dynamic Host Configuration Protocol (DHCP) server that uses a loop back IP address, define the loop back IP address and the required values in the following format:

  ```
  LOOPBACK-IP     LONG-HOSTNAME    SHORT-HOSTNAME
  ```

  For example:

  ```
  127.0.0.1    ibmscala.example.com     ibmscala
  ```

Number of open files and virtual memory limits

Update your open files and virtual memory limits to match the recommended values.

The recommended value of the `ulimit -n` setting, which governs number of open files that are allowed for a process, is `4096`.

The recommended value of the `ulimit -v` setting, which governs the virtual memory limit for a process, is `unlimited`.

For more information, see the Performance and Tuning Guide at: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBMLogAnalyticsBeta/page/PerformanceandTuningGuide

Changing the default number of open files limit

The recommended `ulimit -n` setting, which governs the number of open files that are allowed for a process, is `4096` on each of the servers where a Indexing Engine node is installed. To modify the number of files open limit, complete the following steps:

Procedure

1. Log in to the server where a Indexing Engine node is installed using the Indexing Engine**user id** for that server.
2. Run the following command in a terminal window:

   ```
   ulimit -n
   ```

   If the returned value is less than `4096`, proceed to step 3. If the returned value is more than `4096`, proceed to step 7.
3. Log in to the server as the **root user**.
4. Open the `/etc/security/limits.conf` file.
5. Modify or add the following lines:

   ```
   <user-id> hard nofile 4096
   <user-id> soft nofile 4096
   ```

   where `<user-id>` is the `user id` used to install the Indexing Engine on the server.

   To modify the value for all users on this server, modify or add the following lines:

```
* hard nofile 4096
* soft nofile 4096
```

6. Save your changes.
7. Repeat for each Indexing Engine instance.

What to do next

To ensure that the changes are updated, restart IBM Operations Analytics - Log Analysis.

Changing the virtual memory limit

The recommended `ulimit -v` setting, which limits the virtual memory for processes, is `unlimited` on each of the servers where a Apache Solr node is installed. To modify the limit, complete the following steps:

Procedure

1. Log in to the server where a Indexing Engine node is installed using the Indexing Engine **user id** for that server.
2. Run the following command in a terminal window:

   ```
   ulimit -v
   ```

   If the returned value is not `unlimited`, proceed to step 3. If the returned value is `unlimited`, proceed to step 7.
3. Log in to the server as the **root user**.
4. Open the `/etc/security/limits.conf` file.
5. Modify or add the following lines:

   ```
   <user-id> hard as unlimited
   <user-id> soft as unlimited
   ```

   where `<user-id>` is the `user id` used to install the Indexing Engine on the server.

   To modify the value for all users on this server, modify or add the following lines:

   ```
   * hard as unlimited
   * soft as unlimited
   ```
6. Save your changes.
7. Repeat for each Apache Solr instance.

What to do next

To ensure that the changes are updated, restart IBM Operations Analytics - Log Analysis.

Verifying the IP address and host name configurations

You need to verify that the IP address and host name settings are configured correctly.

Procedure

1. To verify that the host name is configured correctly, enter the following command:

   ```
   hostname
   ```

   If it is configured correctly, the command returns the host name. For example:

```
example
```

2. To verify that the host name uses the fully qualified host name, enter the following command:

   ```
   hostname -f
   ```

   If successful, the command returns the fully qualified host name. For example:

   ```
   example.ibm.com
   ```

3. To confirm that the IP address is configured correctly, ping the host name:

   ```
   ping example
   ```

   If successful, the IP address is returned.

Time server

To ensure that some of the features of IBM Operations Analytics - Log Analysis work correctly, you need to set up a time keeping server.

Some features of IBM Operations Analytics - Log Analysis use relative time such as days, weeks, or months. This time is calculated from the time on the server on which IBM Operations Analytics - Log Analysis is installed. To ensure that these features generate correct results, you must ensure that the time on the IBM Operations Analytics - Log Analysis server is correct. You can use a time server to synchronize the time on the IBM Operations Analytics - Log Analysis server with the correct time. You must do this before you install IBM Operations Analytics - Log Analysis.

Installing

Before you use the command-line interface, the Installation Manager UI, or a silent installation to install IBM Operations Analytics - Log Analysis, read the prerequisites.

Log Analysis includes IBM Installation Manager Version 1.8.2. You can use this version to install Log Analysis immediately. You can also use an existing version of IBM Installation Manager to install Log Analysis. For more information, see the product documentation at http://www-01.ibm.com/support/knowledgecenter/SSDV2W/im_family_welcome.html.

Prerequisites

- Ensure that you complete all the prerequisites. For more information, see Prerequisite tasks.
- Ensure that your user has the access rights that are required to add files to the location where you want to install IBM Operations Analytics - Log Analysis.
- If you previously installed IBM Tivoli Monitoring Log File Agent 6.3 or lower, the installation fails. To solve this problem, stop the existing IBM Tivoli Monitoring Log File Agent installation or rename the folder that was created when it was installed. For detailed information, see the topic about the installation failure if IBM Tivoli Monitoring Log File Agent was installed in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.
- Before you install IBM Operations Analytics - Log Analysis, you must ensure that the details for each host server are maintained correctly in the `/etc/hosts` directory on the target system. Failure to complete this task can result in Oauth errors when you run a custom app. For more information, see the *Verify host server IP address and names* section in Prerequisite tasks.

- Ensure that IBM Installation Manager is not configured to search the service repositories. If the server is not connected to the internet, this setting can cause the installation to fail or stall.

Installing IBM Operations Analytics - Log Analysis with the IBM Installation Manager UI

You must complete the following steps to install IBM Operations Analytics - Log Analysis using the IBM Installation Manager User Interface (UI).

Before you begin

- Ensure that the **Search service repositories during installation and updates** check box is not selected. If you do select this check box and the server is not connected to the internet, the installation can stall or fail.

About this task

When you run the installation script, IBM Operations Analytics - Log Analysis and IBM Installation Manager are installed. IBM Installation Manager Version 1.8.2 is installed when you install IBM Operations Analytics - Log Analysis. Where necessary, IBM Operations Analytics - Log Analysis upgrades the currently installed version of IBM Installation Manager.

Note: The installation path accepts ASCII characters only. Other characters, such as native language characters are not supported.

Procedure

1. Download the appropriate edition. For more information see “Downloading the installation package” on page 3.
2. Copy and extract the installation archive to a location on your server.
3. From the directory to which you extracted the installation files, run the command:

   ```
   ./install.sh
   ```

   Note: Add a `-c` parameter to start the installation in console only mode. For more information, see “Installing with the IBM Installation Manager command-line interface” on page 11.

   Note: To install IBM Operations Analytics - Log Analysis on a remote server using GUI mode, ensure that virtual desktop software is installed on the server that you want to install IBM Operations Analytics - Log Analysis on.
4. The Install Packages screen is displayed.
5. To install IBM Operations Analytics - Log Analysis in the default directory, click **Next**. To install IBM Operations Analytics - Log Analysis to a different location, click **Browse**, select an alternative location, and click **Next**.
6. To continue, accept the license agreement, and click **Next**.
7. To accept the default IBM Log File Agent, Apache Solr, and IBM Operations Analytics - Log Analysis options, click **Next**.

   Note: If you cannot select the IBM Log File Agent, ensure that the hardware and software requirements are implemented.
8. The default ports that are used by IBM Operations Analytics - Log Analysis are displayed. Accept the default option if these ports are not in use by any other application or change them if necessary. Click **Next**.

9. If you want to install a local Indexing Engine instance, ensure that the **Apache Solr** check box is selected. The check box is selected by default. If you want to use a local Indexing Engine installation, you must install it now. You cannot install it after the IBM Operations Analytics - Log Analysis is installed. However, you can install instances of Indexing Engines on remote servers after the installation is completed. To enable search and indexing, you must install at least one Indexing Engine instance locally or remotely.
10. Review the summary information that is provided for the installation and click **Install**.
11. To complete the installation click **Finish**.

What to do next

Download a license for IBM Operations Analytics - Log Analysis from Passport Advantage®: http://www-01.ibm.com/software/lotus/passportadvantage/ and complete the steps that are outlined in the license file readme file. If you are upgrading from a previous version of IBM Operations Analytics - Log Analysis, you can manually copy your license details to the new IBM Operations Analytics - Log Analysis installation.

To verify that the installation is complete, log in to IBM Operations Analytics - Log Analysis. For more information, see "Logging in to IBM Operations Analytics - Log Analysis" on page 337.

To install Indexing Engines on remote servers, you can create them after IBM Operations Analytics - Log Analysis is installed. For more information, see "Installing Apache Solr on remote machines" on page 62.

If you do not install a local Indexing Engine instance, you must install Indexing Engine on a remotely connected server. If you do not install at least one Indexing Engine instance either locally or on a remote server, the search and indexing features do not work.

If you do not verify the server details as described in the *Prerequisites* topic, the following error message is displayed when you try to run a custom app:

```
Failed to launch. Could not retrieve OAuth access token.
```

To correct this error, you must ensure that the server details are correct and start the installation again. For more information, see *Could not retrieve Oauth access token* in the *Troubleshooting* section.

Installing with the IBM Installation Manager command-line interface

You can use the IBM Installation Manager command-line interface. to install IBM Operations Analytics - Log Analysis.

Before you begin

- Do not enter `Control -C` to cancel the installation because this setting can cause the installer to behave inconsistently. Instead, to cancel the installation, enter `c` when prompted.

About this task

When you run the installation script, IBM Operations Analytics - Log Analysis and IBM Installation Manager are installed. IBM Installation Manager Version 1.8.2 is installed when you install IBM Operations Analytics - Log Analysis. Where necessary, IBM Operations Analytics - Log Analysis upgrades the currently installed version of IBM Installation Manager.

Note: The installation path accepts ASCII characters only. Other characters, such as native language characters are not supported.

Procedure

1. Download the appropriate edition. For more information see "Downloading the installation package" on page 3.
2. Copy and extract the installation archive to a location on your server.
3. To install IBM Operations Analytics - Log Analysis:
   a. From the directory location of the extracted installation files, run the command:
      ```
      ./install.sh -c
      ```
   b. Select the default IBM Log File Agent, Apache Solr, and IBM Operations Analytics - Log Analysis packages.

      Note: If you cannot select the IBM Log File Agent, ensure that the hardware and software requirements are implemented.
   c. Accept the license agreement.
   d. If required, change the installation location. Otherwise, accept the default location.
   e. Choose the IBM Operations Analytics - Log Analysis feature and, if required, the IBM Tivoli Monitoring Log File Agent feature.
   f. If necessary, change the default port numbers. Otherwise, accept the default ports.
   g. To install a local Indexing Engine instance, ensure that **Apache Solr** is selected. **Apache Solr** is selected by default. To use a local Indexing Engine installation, you must install it now. You cannot install it after the IBM Operations Analytics - Log Analysis is installed. However, you can install Indexing Engine instances on remote machines after the installation is completed. To enable search and indexing, you must install at least one Indexing Engine instance locally or remotely.
   h. If you want to generate an installation response file for future silent installations, select **Generate an Installation Response File**.
   i. When the installation is complete, select **Finish**.
4. To check the details of the installation, choose **View Installed Packages**.

What to do next

Download a license for IBM Operations Analytics - Log Analysis from Passport Advantage: http://www-01.ibm.com/software/lotus/passportadvantage/ and complete the steps that are outlined in the license file readme file. If you are upgrading from a previous version of IBM Operations Analytics - Log Analysis, you can manually copy your license details to the new IBM Operations Analytics - Log Analysis installation.

To verify that the installation is complete, log in to IBM Operations Analytics - Log Analysis. For more information, see "Logging in to IBM Operations Analytics - Log Analysis" on page 337.

To install Apache Solr on remote server, you can create them after IBM Operations Analytics - Log Analysis is installed. For more information, see "Installing Apache Solr on remote machines" on page 62.

If you do not install a local Indexing Engine instance, you must install Indexing Engine on a remotely connected server. If you do not install at least one Indexing Engine instance either locally or on a remote server, the search and indexing features do not work.

If you do not verify the server details as described in the *Prerequisites* topic, the following error message is displayed when you try to run a custom app:

```
Failed to launch. Could not retrieve OAuth access token.
```

To correct this error, you must ensure that the server details are correct and start the installation again. For more information, see *Could not retrieve Oauth access token* in the *Troubleshooting* section.

Silently installing IBM Operations Analytics - Log Analysis

You can install IBM Operations Analytics - Log Analysis silently by using the sample response file that is provided with the product. This automates the installation procedure.

Before you begin

- Download and extract the IBM Operations Analytics - Log Analysis installation archive. The archive contains the product files and a sample response file, `sample_smcl_silent_install.xml`, that is required for silent installation.
-

Note: A silent installation can fail if the IBM Installation Manager repository changed since the last installation or uninstall. This problem can occur even when you update the response file with the correct repository location. If you are installing from a new repository, remove or close any old repository connections.

About this task

When you run the installation script, IBM Operations Analytics - Log Analysis and IBM Installation Manager are installed. IBM Installation Manager Version 1.8.2 is installed when you install IBM Operations Analytics - Log Analysis. Where necessary, IBM Operations Analytics - Log Analysis upgrades the currently installed version of IBM Installation Manager.

Note: The installation path accepts ASCII characters only. Other characters, such as native language characters are not supported.

Silently installing IBM Operations Analytics - Log Analysis involves modifying the sample response file and then calling IBM Installation Manager from the command line or from a script to install the product. IBM Installation Manager obtains the installation settings from the response file.

Procedure

To silently install IBM Operations Analytics - Log Analysis:

1. Download the appropriate edition. For more information see "Downloading the installation package" on page 3.
2. Copy the sample response file to a suitable local directory (for example, your home directory). Use a suitable name for the local response file, for example: `smcl_silent_install.xml`
3. Modify the response file to suit your environment:
   a. Locate the line and edit this line to reflect your home directory:

      ```
      <preference name='com.ibm.cic.common.core.preferences.eclipseCache'
      value='/home/MYUSERID/IBM/IBMIMShared'/>
      ```

      where `/home/MYUSERID` is the location of your `home` directory.
   b. Locate and edit these lines to specify the directory where you extracted the installation image:

      ```
      <repository location='/home/MYUSERID/IMAGEDIRECTORY/im.linux.x86'/>
      <repository location='/home/MYUSERID/IMAGEDIRECTORY'/>
      ```

      where `/home/MYUSERID` is your home directory and `IMAGEDIRECTORY` is the name of the directory to which you extracted the installation package.

      Note: The paths that are given assume that you extracted the installation package to your home directory.
   c. Locate and edit these lines to reflect your `home` directory:

      ```
      <profile id='IBM Installation Manager' installLocation=
      '/home/MYUSERID/IBM/InstallationManager/eclipse' kind='self'>
      <data key='eclipseLocation' value='/home/MYSERUSERID/IBM/
      InstallationManager/eclipse'/>
      ```

      and

      ```
      <profile id='IBM Log Analytics'
      installLocation='/home/MYUSERID/IBM/LogAnalysis'>
      ```

      and

      ```
      <data key='eclipseLocation' value='/home/MYUSERID/IBM/
      LogAnalysis'/>
      ```

      where `/home/MYUSERID` is the location of your `home` directory.
   d. Optional: If necessary, change the following default port numbers:

      Note: Default port numbers are used by IBM Operations Analytics - Log Analysis, only modify the values if necessary.

      ```
      <!-- Application WebConsole Port -->
      <data key='user.unity.port.number,com.ibm.tivoli.scloganalytics'
       value='9988'/>
      <!-- Application WebConsole Secure Port -->
      <data key='user.unity.secureport.number,com.ibm.tivoli.scloganalytics'
       value='9987'/>
      <!-- Database Server Port -->
      <data key='user.database.port.number,com.ibm.tivoli.scloganalytics'
       value='1627'/>
      <!-- Data Collection Server Port -->
      <data key='user.eif.port.number,com.ibm.tivoli.scloganalytics'
       value='5529'/>
      <!-- ZooKeeper Port -->
      <data key='user.zookeeper.port.number,com.ibm.tivoli.scloganalytics'
       value='12181'/>
      ```

```
<!-- Apache Solr Search Port -->
<data key='user.searchengine.port.number,com.ibm.tivoli.scloganalytics'
 value='9983'/>
<!-- Apache Solr Stop Port -->
<data key='user.searchengineQS.port.number,com.ibm.tivoli.scloganalytics'
 value='7205'/>
```

   e. Save your changes.

4. Optional: To exclude IBM Tivoli Monitoring Log File Agent from the installation, remove the `LOG_FILE_AGENT` parameter from the `offering id` element.

   For example, change the following default entry:

   ```
   <offering id='com.ibm.tivoli.scloganalytics'
    profile='IBM Log Analytics'
    features='IBM Log Analytics,LOG_FILE_AGENT'
    installFixes='none'/>
   ```

   to:

   ```
   <offering id='com.ibm.tivoli.scloganalytics'
    profile='IBM Log Analytics'
    features='IBM Log Analytics'
    installFixes='none'/>
   ```

5. Optional: To exclude Indexing Engine, remove the `Data Explorer Application` parameter from the `offering id` element.

   For example, change the following default entry:

   ```
   <offering id='com.ibm.tivoli.scloganalytics'
    profile='IBM Log Analytics'
    features='IBM Log Analytics,LOG_FILE_AGENT,
   Data Explorer Application'
    installFixes='none'/>
   ```

   to:

   ```
   <offering id='com.ibm.tivoli.scloganalytics'
    profile='IBM Log Analytics'
    features='IBM Log Analytics, LOG_FILE_AGENT,'
    installFixes='none'/>
   ```

6. If you already have IBM Installation Manager installed, use this command to start the silent installation:

   ```
   ~/IBM/InstallationManager/eclipse/tools/imcl -s -input
   <HOME_DIR>/smcl_silent_install.xml -sVP -acceptLicense
   ```

   Where <*HOME_DIR*> is the directory where you stored the response file.

   If you do not have IBM Installation Manager installed, use the `install.sh` command to install both IBM Installation Manager and IBM Operations Analytics - Log Analysis. From the directory to which you extracted the installation archive, run the command:

   ```
   ./install.sh -s <HOME_DIR>/smcl_silent_install.xml
   ```

   where <*HOME_DIR*> is your home directory. This command silently installs IBM Operations Analytics - Log Analysis and IBM Installation Manager Version 1.8.2, if no other version of IBM Installation Manager is installed.

Results

The progress of the installation is displayed in an IBM Installation Manager console. To install without displaying the console, leave out the `-sVP` option (which shows Verbose Progress).

What to do next

Download a license for IBM Operations Analytics - Log Analysis from Passport Advantage: http://www-01.ibm.com/software/lotus/passportadvantage/ and complete the steps that are outlined in the license file readme file. If you are upgrading from a previous version of IBM Operations Analytics - Log Analysis, you can manually copy your license details to the new IBM Operations Analytics - Log Analysis installation.

To verify that the installation is complete, log in to IBM Operations Analytics - Log Analysis. For more information, see "Logging in to IBM Operations Analytics - Log Analysis" on page 337.

To install Apache Solr on remote machines, you can create them after IBM Operations Analytics - Log Analysis is installed. For more information, see "Installing Apache Solr on remote machines" on page 62.

If you do not install a local Indexing Engine instance, you must install Indexing Engine on a remotely connected server. If you do not install at least one Indexing Engine instance either locally or on a remote server, the search and indexing features do not work.

If you do not verify the server details as described in the *Prerequisites* topic, the following error message is displayed when you try to run a custom app:

```
Failed to launch. Could not retrieve OAuth access token.
```

To correct this error, you must ensure that the server details are correct and start the installation again. For more information, see *Could not retrieve Oauth access token* in the *Troubleshooting* section.

Installing and configuring the IBM Tivoli Monitoring Log File Agent

When you install IBM Operations Analytics - Log Analysis, you can also choose to install the IBM Tivoli Monitoring Log File Agent that is delivered with the product.

After you install the IBM Tivoli Monitoring Log File Agent, you need to configure it. If you choose not to install the internal IBM Tivoli Monitoring Log File Agent, for example if you have an existing installation that you want to use instead, you still need to configure the integration with IBM Operations Analytics - Log Analysis. For more information, see "IBM Tivoli Monitoring Log File Agent configuration scenarios" on page 226.

Installing on Linux on System z and Linux on System x based servers

Before you can install IBM Operations Analytics - Log Analysis on a Linux on System z or Linux on System x based operating system, read and complete the prerequisites.

Hardware requirements

Linux on System z runs on System z based hardware. Linux on System x runs on Intel or AMD-based hardware. Both types of hardware are supported but there are some minor differences in the software requirements.

Note: If you install IBM Operations Analytics - Log Analysis on Intel or AMD-based hardware, you must install IBM Operations Analytics - Log Analysis components like the Indexing Engine server on Intel or AMD-based hardware. You cannot install IBM Operations Analytics - Log Analysis components on Linux on System z based hardware. Likewise, if you install IBM Operations Analytics - Log Analysis on Linux on System z based hardware, you must install IBM Operations Analytics - Log Analysis components like the Indexing Engine server on System z based hardware.

For more information about the hardware requirements, see Hardware and software requirements.

Prerequisites

For more information about the prerequisite tasks that you must complete, see Prerequisite tasks.

Supported operating systems for cross-platform data ingestion

You can choose one of the following methods for loading data into IBM Operations Analytics - Log Analysis:

1. Using the internal IBM Tivoli Monitoring Log File Agent that is bundled with IBM Operations Analytics - Log Analysis to stream data.
2. Using an external IBM Tivoli Monitoring Log File Agent which you install separately to stream data.
3. Using the z/OS® Log Forwarder that is bundled with the z/OS Insight Packs.
4. Using the data collector client or FTP to load a batch of data.

Each of these scenarios offers varying cross-platform support as outlined in the following table:

Table 1. Supported operating systems for data loading scenarios

Data loading scenario	Supported operating systems
1	See the *Supported Operating Systems* section in “Loading and streaming data” on page 223
2	See the *Requirements for the monitoring agent* topic documentation for your version of IBM Tivoli Monitoring at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring
3	See the *System requirements for the z/OS Log Forwarder* topic in the product documentation that is linked in the *Insight Packs* section
4	See the *Supported Operating Systems* section in “Loading and streaming data” on page 223

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

Insight Packs for z/OS

You can install Insight Packs for z/OS SYSLOG and WebSphere® Application Server for z/OS separately on both Linux on System z and Linux on System x based servers. For more information, see:

- Documentation for IBM Operations Analytics - Log Analysis z/OS - Insight Packs SYSLOG
- Documentation for IBM Operations Analytics - Log Analysis z/OS - Insight Packs for IBM WebSphere Application Server

Linux on System z logstash support

logstash 1.4.2 is bundled with the IBM Operations Analytics - Log Analysis Linux on System z 64-bit installation package. However, logstash cannot be deployed on Linux on System z, except for log aggregation on Linux x86 based hardware.

Installing on non-English-language systems

If you want to install IBM Operations Analytics - Log Analysis on a system where the locale is not set to English United States, you need to set the locale of the command shell to `export LANG=en_US.UTF-8` before you run any IBM Operations Analytics - Log Analysis scripts. For example, when you install IBM Operations Analytics - Log Analysis you need to run the following command to change the locale before you run the `unity.sh` script:

```
export LANG=en_US.UTF-8
```

Tuning the operating system

To tune the Linux operating system, ensure that the following resource configuration settings are made:

Number of concurrent files: 4096
Use the `ulimit -n` command to change this setting to 4096.

Virtual memory: Unlimited
Use the `ulimit -v` command to change this setting to unlimited.

Removing IBM Operations Analytics - Log Analysis

You can use the command-line, the IBM Installation Manager UI or a silent removal process to remove IBM Operations Analytics - Log Analysis.

Removing IBM Operations Analytics - Log Analysis

This topic outlines the steps that you must complete to remove IBM Operations Analytics - Log Analysis.

About this task

This procedure outlines how to remove IBM Operations Analytics - Log Analysis and IBM Installation Manager from your environment. Both of these components are installed when you install IBM Operations Analytics - Log Analysis. To complete the uninstallation, remove IBM Operations Analytics - Log Analysis and then, if required, complete the procedure to remove IBM Installation Manager.

Note: If you have remote installations of Apache Solr, you must remove these before you remove IBM Operations Analytics - Log Analysis. For more information about how to do this, see "Removing Apache Solr instances" on page 64

Procedure

1. To remove IBM Operations Analytics - Log Analysis:
   a. Navigate to the `<HOME>/IBM/InstallationManager/eclipse/launcher` directory and execute the command:
      ```
      ./launcher
      ```

      Note: If you are accessing the installation environment remotely, ensure that your virtual desktop software is configured to allow you to view the graphical user interface for the IBM Installation Manager.
   b. Click **Next**.
   c. Select the IBM Operations Analytics - Log Analysis package and click **Next**.
   d. Click **Uninstall**. Allow the removal to proceed and when complete, click **Finish**.
2. (Optional) To remove IBM Installation Manager:
   a. From the `<HOME>/var/ibm/InstallationManager/uninstall` directory, execute the command:
      ```
      ./uninstall
      ```

      where <HOME> is the directory to which you have installed IBM Operations Analytics - Log Analysis
   b. Complete the uninstallation steps and click **Finish**.

Using the console to remove IBM Operations Analytics - Log Analysis

This topic outlines the steps that you must complete to remove IBM Operations Analytics - Log Analysis using the console.

About this task

This procedure outlines how to remove IBM Operations Analytics - Log Analysis and IBM Installation Manager from your environment using the console. Both of these components are installed when you install IBM Operations Analytics - Log Analysis. To complete the uninstallation, remove IBM Operations Analytics - Log Analysis and then, if required, complete the procedure to remove IBM Installation Manager.

Note: If you have remote installations of Apache Solr, you must remove these before you remove IBM Operations Analytics - Log Analysis. For more information about how to do this, see "Removing Apache Solr instances" on page 64.

Procedure

1. To remove IBM Operations Analytics - Log Analysis:
   a. Navigate to the `<HOME>/InstallationManager/eclipse/tools` directory and execute the command:

      ```
      ./imcl -c
      ```

   b. Enter5 and press `Enter`.
   c. Enter `1` to select the IBM Operations Analytics - Log Analysis package group and press `N` to proceed.
   d. Enter `1` to select the IBM Operations Analytics - Log Analysis package, press `Enter`.
   e. Enter `N` to proceed.
   f. Enter `U` to start the uninstallation.
   g. After the uninstallation has completed, enter `F` to complete the uninstallation.
   h. Enter `X` to close the IBM Installation Manager.
2. (Optional) To remove IBM Installation Manager:
   a. From the `<USER_HOME>/var/ibm/InstallationManager/uninstall` directory, execute the command:

      ```
      ./uninstallc
      ```

   b. The uninstallation proceeds and completes.

Silently removing IBM Operations Analytics - Log Analysis

You can remove IBM Operations Analytics - Log Analysis silently by using the sample response file that is provided with the product.

Before you begin

You require the sample response file `<HOME>/IBM/LogAnalysis/work_files/removal/sample_silent_loganalytics_removal.xml`.

Note: A silent removal can fail if the IBM Installation Manager repository changed since the last installation or removal. This problem can occur even when you update the response file with the correct repository location. If the repository changed, remove or close the old repository connections before you remove the product.

Note: If you have remote installations of Apache Solr, remove them before you remove IBM Operations Analytics - Log Analysis. For more information about how to do this, see "Removing Apache Solr instances" on page 64.

About this task

Silently removing IBM Operations Analytics - Log Analysis involves modifying the sample response file and then calling IBM Installation Manager from the command line or from a script to remove the product. IBM Installation Manager obtains the required settings from the response file.

Procedure

To silent removal the product:

1. Copy the sample response file to a suitable local directory (for example, your home directory). Use a suitable name for the local response file, for example: `smcl_removal.xml`
2. Modify the response file to suit your environment:
   a. Specify the IBM Installation Manager repository location for your environment. For example: `repository location='/home/smcl/smcl_build/'`
   b. If you changed the default IBM Operations Analytics - Log Analysis port numbers during or after installation, update the following entries with the new port numbers:

```
<!-- IBM Log Analytics WebConsole Port Number -->
<data key='user.unity.port.number,com.ibm.tivoli.scloganalytics'
 value='9988'/>

<!-- IBM Log Analytics WebConsole Secure Port Number -->
<data key='user.unity.secureport.number,com.ibm.tivoli.scloganalytics'
 value='9987'/>

<!-- IBM Log Analytics Database Port Number -->
<data key='user.database.port.number,com.ibm.tivoli.scloganalytics'
 value='1627'/>

<!-- IBM Log Analytics DataCollection (EIF) Server Port Number -->
<data key='user.eif.port.number,com.ibm.tivoli.scloganalytics'
 value='5529'/>

<!-- IBM Log Analytics Search Engine WebConsole Server Port Number -->
<data key='user.searchengine.port.number,com.ibm.tivoli.scloganalytics'
 value='9989'/>

<!-- IBM Log Analytics Distributed Application Management Server
 Port Number -->
<data key='user.zookeeper.port.number,com.ibm.tivoli.scloganalytics'
 value='12181'/>
```

   c. Save your changes.
3. Use the following command to removal the product:

```
~/IBM/InstallationManager/eclipse/tools/imcl -s -input
<HOME_DIR>/smcl_removal.xml -sVP -acceptLicense
```

   Where *<HOME_DIR>* is the directory where you stored the response file.

Results

The progress of the removal is displayed in an IBM Installation Manager console. To run the removal script displaying the console, omit the `-sVP` option (which shows Verbose Progress).

Installing reference

Read the information about the scripts and properties that you can configure when you install Log Analysis.

Configuration properties file

After you complete the installation, to modify the configuration properties edit the `unitysetup.properties` file.

Modify only the properties that are listed in the table.

Note: You cannot change any of the other parameters in the file. These parameters are identified as such in the file. If you do change one of these parameters, IBM Operations Analytics - Log Analysis may not work correctly or at all.

Table 2. `unitysetup.properties` parameters

Parameter	Description
`MAX_SEARCH_RESULTS=1000`	The maximum number of search results that can be returned by the search query.
`MAX_DATA_FACETS_IN_CHART=1000`	The maximum number of facets or data points that can be included in returned search results.
`#SEARCH_QUERY_FOR_DEFAULT_SEARCH=*`	Uncomment this parameter to enable the default search.
`MAX_WORK_QUEUE_SIZE=100000`	Determines the maximum size of the queue of documents that are being held for annotation and indexing.
`NUM_DOCUMENT_PROCESSOR_THREADS=30`	Determines the thread pool that is used to select, annotate, and index a document.
`batchsize=500000`	Determines the maximum number of records that can be loaded by the Generic Receiver in single batch.
`UNITY_APP_TIME_OUT=180`	Determines the default time-out value in seconds for the apps. The default value is 180 seconds.
`ENABLE_SOLR_FACET_CACHE=false`	To enable the facet cache for wildcard searches, set this parameter to true. Use this setting if you run wildcard searches on data that is older than 1 day. When it is enabled, the facets are counted before they are indexed by Log Analysis.

Default ports

IBM Operations Analytics Log Analysis uses a number of default ports during the installation process.

Default ports

Table 3. The default ports for IBM Operations Analytics Log Analysis are as follows:

Ports	Default Value	Description
Application WebConsole Port	9988	Use this port for unsecured http communication with the web application of IBM Operations Analytics Log Analysis.
Application WebConsole Secure Port	9987	Use this port for secured http communication with the web application of IBM Operations Analytics Log Analysis.
Database Server Port	1627	This port is used by IBM Operations Analytics Log Analysis for its internal database.

Table 3. The default ports for IBM Operations Analytics Log Analysis are as follows: (continued)

Ports	Default Value	Description
Data Collection Server Port	5529	This port is used by IBM Operations AnalyticsLog Analysis to collect data.
ZooKeeper Port Number	12181	This port is used by ZooKeeper service of IBM Operations AnalyticsLog Analysis to manage its Apache Solr nodes
Apache Solr Search Port	8983	This port is used by the Apache Solr server to listen for search queries.
Apache Solr Stop Port	7205	This port is used by IBM Operations AnalyticsLog Analysis to stop the Apache Solr server.

Note: The default ports listed in Table 1 apply to local IBM Operations AnalyticsLog Analysis installations. Remote IBM Operations AnalyticsLog Analysis installations may require different port numbers.

install.sh command

Use the `install.sh` command to install IBM Operations Analytics - Log Analysis.

The `install.sh` command is in the <HOME>/IBM/LogAnalysis/remote_install_tool/ directory on the local installation of IBM Operations Analytics - Log Analysis.

install.sh command parameters

To install IBM Operations Analytics - Log Analysis with IBM Installation Manager, run the command:

```
./install.sh
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades, IBM Installation Manager if no other version is installed. For more information, see “Installing IBM Operations Analytics - Log Analysis with the IBM Installation Manager UI” on page 10.

To install IBM Operations Analytics - Log Analysis with the console, run the command:

```
./install.sh -c
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager, if no other version of IBM Installation Manager is installed. For more information, see “Installing with the IBM Installation Manager command-line interface” on page 11.

To silently install IBM Operations Analytics - Log Analysis, run the command:

```
./install.sh -s <HOME_DIR>/smcl_silent_install.xml
```

where <*HOME_DIR*> is your home directory. This command silently installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager Version 1.8.2. For more information, see "Silently installing IBM Operations Analytics - Log Analysis" on page 13.

To install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server, run the `remote_deploy.sh` script. For more information, see "Installing Apache Solr on remote machines" on page 62.

`backup_restore.sh` script

To back up or restore data, use the `backup_restore.sh` script.

Syntax

```
backup_restore.sh <backup_directory> -backup| -restore
```

where <*backup_directory*> is the path for the directory that you create to store the backed up files.

Parameters

<*backup_directory*>
The path for directory that you create to store the backed up files

backup Backs up the current data to the backup directory.

restore
Restores the data stored in the backup directory.

Upgrading, backing up, and migrating data

You use the same script to back up, restore, and migrate your data during an upgrade.

Before you begin

- Read about the limitations that apply to this procedure. For more information, see "Backup and migration limitations" on page 26
- If you do not configure a key-based Secure Shell (SSH) authentication as part of the installation, you are prompted for the password during the restoration. For more information about setting up SSH, see "Setting up Secure Shell to use key-based authentication" on page 65
- If you use the `Overlaps()` function in the Annotation Query Language (AQL) for any Insight Packs, you must update your code before you migrate the data. The function used four parameters, including two null statements in the Insight Packs in versions previous to 1.3. The null statements are no longer required and can be removed. For example, if your code contains a statement such as `where Overlaps(D.OrginalDateSpan, T.nonnormative_time, 0, 0);`, you must remove the null statements, changing it to `where Overlaps(D.OrginalDateSpan, T.nonnormative_time);`. If you do not make this correction, the ingestion into datasources that uses these Insight Packs fails.
- IBM Operations Analytics - Log Analysis 1.3.0 and earlier stored user and group information in the `unityUserRegistry.xml` file in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory. To migrate data to IBM Operations Analytics - Log Analysis 1.3.1, copy the user and group information in the `unityUserRegistry.xml` file to the `unityUserRegistry_130.xml` file on IBM Operations Analytics - Log Analysis 1.3.1 before you run the restore script.

About this task

These items are backed up and migrated by this procedure:

- 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
- Custom Apps
- Insight Packs
- Log File Agent (LFA) configuration files
- License configuration files
- All chart specifications, including custom chart specifications

In addition to these files, a number of files that are not required for a new installation are also backed up and maintained for reference purposes. These files include:

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

Data other than the types of listed previously are not backed up or restored. Any customization that is made outside the files that are listed here must be migrated manually. For example, user information and changes to passwords for default users are not migrated to the target server.

LDAP information for one LDAP server is migrated automatically. If you have more than one DAP server, you must migrate and configure the information from the other LDAP server manually. The migrated information is stored in the `ldapRegistry.xml` file.

Procedure

To back up and restore data in IBM Operations Analytics - Log Analysis 1.3.1:

1. Back up your data. For more information, see "Backing up data" on page 27.
2. Restore your data. For more information, see "Restoring data" on page 27.

To migrate data from IBM Operations Analytics - Log Analysis 1.3.0 to 1.3.1:

1. Back up your data in 1.3.
2. Move the backed up compressed files to the `<Backup_dir>` on the 1.3.1 server.
3. Restore the data on the 1.3.1 server.

What to do next

If you use the Generic Annotation Insight Pack, run the following command to upgrade the Insight Pack:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -upgrade
$<HOME>/IBM/LogAnalysis/unity_content/GA/GAInsightPack_v1.1.1.3.zip -f
```

If you migrate data to a new target server, you must complete the following steps to update the data source with the relevant information for the server:

1. Log in to IBM Operations Analytics - Log Analysis and open the **Data sources** workspace.
2. Identify the data sources that use **Local file** as the location in their configuration. For each of these data sources, open the data source and complete the steps in the wizard without changing any data. This action updates the data sources with the relevant information for the new server.
3. Identify the data sources that use **Custom** as the location in their configuration. For each of these data sources, open the data source and complete the steps in the wizard, updating the host name to match the new one.

Backup and migration limitations

Before you back up, restore or migrate data between the same or different versions, read the limitations.

The following restrictions apply:

- To restore backed up data from Apache Solr nodes, you must set up at least the same number of Apache Solr nodes in the target system as existed previously in the source system.
- During data restoration, backed up data is not merged with existing data on the server. Backed up data must be restored on a target system immediately after IBM Operations Analytics - Log Analysis is installed.

- Restoring backed up data must not be completed more than once on a target server. If errors occur restoring backed up data, you must attempt the restoration after you remove and install IBM Operations Analytics - Log Analysis.
- 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 completed outside these directories are not backed up or restored.
- If you create a custom application that points to the Derby database and, which has an encrypted user ID and password, you must update the application to reflect changes to the encryption of the Derby database user ID and password when you migrate to a new version. The custom application cannot run until the application is edited to reflect the encryption that is used by the updated installer.
- When you restore a system that was extended with extra Insight Packs (for example, Insight Packs created with DSV toolkit), the log files and data source directories are not restored. To resolve this issue, you must manually add these directories, using the appropriate LFA configuration file as a reference.

Backing up data

To back up data, complete the procedure.

Procedure

1. To stop IBM Operations Analytics - Log Analysis, use the `unity.sh` script that is in the `<HOME>/IBM/LogAnalysis/utilities` directory:

   ```
   ./unity.sh -stop
   ```

2. Create a backup directory where you will store the backup files. Do not create this directory in the same directory in which IBM Operations Analytics - Log Analysis is installed. This directory is called the backup directory or *<backup_dir>* in this procedure.
3. From the `<HOME>/IBM/LogAnalysis/utilities/migration` directory, run the following command:

   ```
   ./backup_restore.sh <backup_dir> backup
   ```

   where *<backup_dir>* is the directory that you created in step 2. The backup directory that you specify must be empty.

   The backup command creates a set of archive files in the backup directory.

Results

When you run the backup command, IBM Operations Analytics - Log Analysis creates a zipped file that contains the archived data.

The zipped files have the word restore in the file name and are numbered sequentially. For example, `LogAnalytics_30Jul2014_Restore_001.zip`. These are the files that you can use later to restore the data.

The command also generates reference files. These files are stored in a separate archive file. The file name contains the words BackupOnly, for example `LogAnalytics_30Jul2014_BackupOnly_001.zip`. You do not need to restore these.

Restoring data

To restore backed up data, complete the following procedure.

Procedure

1. To stop IBM Operations Analytics - Log Analysis, use the `unity.sh` script that is in the `<HOME>/IBM/LogAnalysis/utilities` directory:

   ```
   unity.sh -stop
   ```

2. From the `<HOME>/IBM/LogAnalysis/utilities/migration` directory, run the command:

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

   where `<Backup_dir>` is the path to the directory containing your backed up files.

   If you migrate Apache Solr data, you must create the same number of Apache Solr nodes in the target system. If you specified a password when you created the nodes, you are prompted for the password.

3. If LDAP is configured on your source IBM Operations Analytics - Log Analysis server, you are prompted for the LDAP bind password during the restoration of your data on the target server.
4. If you have configured IBM Operations Analytics - Log Analysis to stream data from the same server on which it has been installed, and have migrated IBM Operations Analytics - Log Analysis to a new target server, review and update the host name setting in your data sources to reflect the host name of the target server.
5. To start IBM Operations Analytics - Log Analysis, use the `unity.sh` script that is in the `<HOME>/IBM/LogAnalysis/utilities` directory:

   ```
   unity.sh -start
   ```

What to do next

Verify that the data migration and restoration has been successful. Confirm that all artifacts and data have been restored before you delete the back up archive. Progress information and errors recorded during back up and restore are stored in the `<HOME>/IBM/LogAnalysis/utilities/migration/logs` directory.

Configuring

You can configure IBM Operations Analytics - Log Analysis through the user interface and command-line interfaces. You can also administer and manage application security and single sign-on. This section outlines how to configure IBM Operations Analytics - Log Analysis.

Postinstallation configuration

After you install IBM Operations Analytics - Log Analysis, you must complete the required postinstallation tasks.

The postinstallation configuration includes the following tasks:

- Installing the sample data. This is optional.
- Creating users and assigning roles

After you complete the postinstallation configuration, you can scale your installation. For more information, see “Configuring scalable data streaming from multiple, remote sources” on page 61.

If you want to install Apache Solr on remote servers, you can create them after IBM Operations Analytics - Log Analysis is installed. For more information, see “Installing Apache Solr on remote machines” on page 62.

If you do not install a local instance of Apache Solr, you must install Apache Solr on one of the remotely connected servers. If you do not install at least one instance of Apache Solr either locally or on a remote server, the search and indexing feature do not work.

Changing the default timezone

IBM Operations Analytics - Log Analysis uses Coordinated Universal Time as the default timezone. To change the default timezone, complete this procedure.

About this task

You must change this setting after you install IBM Operations Analytics - Log Analysis but before you load any data, including the sample data that is provided on the **Getting Started** page.

Note: You cannot change the timezone after you load data. IBM Operations Analytics - Log Analysis cannot resolve the different time stamps and this conflict causes errors in the search that cannot be resolved. After you change the timezone and load data, do not change the timezone again.

IBM Operations Analytics - Log Analysis supports the Java timezone classification. For a list of supported timezone names, see the “Supported timezone names” on page 210 topic in the *Configuring* reference section

Procedure

1. Install IBM Operations Analytics - Log Analysis. Do not load or install any data.

2. To stop IBM Operations Analytics - Log Analysis, enter the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
   ```

3. To change the default timezone value, edit the `UNITY_TIME_ZONE` parameter in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file. For example, to change the timezone to Western Europe for Paris, France, edit the parameter as follows:

   ```
   UNITY_TIME_ZONE=Europe/Paris
   ```

   Note: You must use the full timezone name rather than the timezone abbreviation in the timezone parameter.
4. Save your changes.
5. To restart IBM Operations Analytics - Log Analysis, enter the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Users and roles

You can create and modify users and roles in the IBM Operations Analytics - Log Analysis UI to assign role-based access control to individual users.

You can use the IBM Operations Analytics - Log Analysis UI to create and modify users and user roles to provide individual accounts and roles to users. This role-based access control enables the administrator to assign individual access and does not support user or object groups.

IBM Operations Analytics - Log Analysis includes the following default users and roles.

`unityuser`
: This default user is assigned the default `unityusers` role. All users are assigned to the `unityusers` role by default.

`unityadmin`
: This default user is assigned the default `unityadmins` role. This user has access to all data in IBM Operations Analytics - Log Analysis. No explicit permission needs to be set for this user.

You can use the default users and roles as outlined in Table 1, or create new users and roles. For more information about creating users and roles, see *Create and modify users* and *Create and modify roles* in the *Postinstallation configuration* section of the *Configuring IBM Operations Analytics - Log Analysis* guide.

Table 4. Default users and roles

Default user	Default role
`unityuser`	`unityusers`
`unityadmin`	`unityadmins`, `unityusers`

Users with the `unityuser` role can access the search workspace. Users with the `unityadmin` role can access the search and administration workspaces.

By default, all users are assigned the `unityuser` role.

The `unityadmin` user has access to all data. Only one `unityadmin` user with the `unityadmins` role is supported in IBM Operations Analytics - Log Analysis.

Note: To create or modify users or roles, you must have access to the administration workspace.

Creating a user

You can use the IBM Operations Analytics - Log Analysis UI to create a user.

Before you begin

To create new user accounts, you must have administrative access.

Procedure

To create a new user account, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings.**
2. Select the **Users** tab.
3. Click the add icon to open a new **Add user** pane.
4. Complete the new user details in the **Add user** pane.
5. To add a role to the user, click the add icon in the **Add user** pane. Select the required role and click **OK**.
6. To save the new user, click **OK**.

Adding or deleting user roles:

You can add or delete user roles in the IBM Operations Analytics - Log Analysis UI to create a user.

About this task

Adding and deleting user roles in the IBM Operations Analytics - Log Analysis UI enables management of the individual profiles of distinct users.

Procedure

To add or delete user roles, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings.**
2. Select the **Users** tab.
3. Select the user that you want to edit.
4. Click the edit icon . This opens a new **Edit <*username*>** pane.
5. To add a user role to the selected user, click the add icon in the **Edit <*username*>** pane.
6. To delete a user role from the selected user, click the delete icon in the **Edit <*username*>** pane.
7. Select the required role and click **OK**
8. To save your changes, click **OK**.

Editing a user

You can use the IBM Operations Analytics - Log Analysis UI to edit a user.

Before you begin

To edit user accounts, you must have administrative access.

Procedure

To edit a user account, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings.**
2. Select the **Users** tab.
3. Click the edit icon 📝 . This opens a new **Edit <*username*>** pane.
4. Edit the user details in the **Edit <*username*>** pane.
5. To save your changes, click **OK**.

Changing a user password

Users passwords are changed by using the IBM Operations Analytics - Log Analysis UI.

Users can choose to change their own passwords or the `unityadmin` user can change the password of an existing user.

Changing your password:

You can use the IBM Operations Analytics - Log Analysis UI to change your password.

Procedure

To edit your password, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI.
2. On the **Getting Started** page, expand the **username**.
3. Select **Change Password**.
4. Complete the fields Edit Password pane.
5. Click **OK**.

Changing a user password:

You can use the IBM Operations Analytics - Log Analysis UI to change the password of an existing user.

Before you begin

To change the password of an existing user, you must have administrative access.

Procedure

1. To edit a user password, complete the following steps.
   a. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings.**
   b. Select the **Users** tab.

c. Click the edit icon 📝 . This opens a new **Edit <*username*>** pane.

d. Change the users password details in the **Edit <*username*>** pane.

a. To save your changes, click **OK**.

2. If you want to change the `unityadmin` password, you must update the encrypted password in the following files to match the updated password.

   a. To generate the encrypted password, use the `unity_securityUtility.sh` utility in the `<HOME>/IBM/LogAnalysis/utilities` directory. For example:

   ```
   unity_securityUtility.sh encode password
   ```

   - `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`. For example:

     ```
     #The password to use to access the unity rest service
     password={aes}EF712133E0677FEBB30624BA5EE62BC2
     ```

   - `<HOME>/IBM/LogAnalysis/remote_install_tool/config/rest-api.properties`. For example:

     ```
     ibm.scala.rest.password={aes}EF712133E0677FEBB30624BA5EE62BC2
     ```

   - `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`. For example:

     ```
     unity.data.collector.password={aes}EF712133E0677FEBB30624BA5EE62BC2
     ```

   - `<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh`. For example:

     ```
     PASSWD={aes}EF712133E0677FEBB30624BA5EE62BC2
     ```

   b. If you change the password that is used by the `unityuser`, you must update the `password` parameter in the `<HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance.sh` script.

   ```
   password={aes}7A0B2401A8E29F37CD768CB78E205CAD
   ```

Deleting a user

You can use the IBM Operations Analytics - Log Analysis UI to delete a user.

Before you begin

To delete user accounts, you must have administrative access.

Note: If a user is deleted while they are logged in to IBM Operations Analytics - Log Analysis, their session continues until they log out. The user is not logged out automatically when they are deleted.

Procedure

To delete a user account, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings.**
2. Select the **Users** tab.
3. Select the user that you want to delete.
4. Click the delete icon ✗ .
5. A message opens asking you to confirm that you want to delete the selected users, click **OK**.

Creating a role

You can use the IBM Operations Analytics - Log Analysis UI to create an application role.

Before you begin

To create new application roles, you must have administrative access.

Procedure

To create a new role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings.**
2. Select the **Roles** tab.
3. Click the add icon .
4. Complete the new role details in the **Add Role** pane.
5. To save the new role, click **OK**.

Editing a role

You can use the IBM Operations Analytics - Log Analysis UI to edit role profiles.

Before you begin

To edit role profiles, you must have administrative access.

Procedure

To edit role profiles, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings.**
2. Select the **Roles** tab.
3. Select the role that you want to edit, and click the edit icon .
4. Modify the fields in the **Edit *<username>*** pane.
5. To save the changes to the role, click **OK**.

Adding users to roles

You can use the IBM Operations Analytics - Log Analysis UI to add users to new or existing role.

Before you begin

To add a user to a new or existing role, you must have administrative access.

Procedure

To add a user to a new or existing role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings.**
2. Select the **Roles** tab.
3. Select the role that you want to add users to:
   - To add a user to a new role, click the add icon .
   - To add a user to an existing role, select the role and click the edit icon .
4. Select the **Assign Users to Role** tab in the **Add Role** or **Edit *<username>*** pane.

5. Click the add icon .
6. Select the users that you want to add to the role from the list, and click **OK**.

   To clear a selected user, hold down the **Ctrl** or **Command** key, and click the user row.
7. To save the changes to the user, click **OK**.

Deleting a user from a role

You can use the IBM Operations Analytics - Log Analysis UI to delete a user from a role.

Before you begin

To delete a user from a role, you must have administrative access.

Procedure

To delete a user from a role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings.**
2. Select the **Roles** tab.
3. Select the role that you want to edit, and click the edit icon .
4. Select the **Assign Users to Role** tab in the **Edit <*username*>** pane.
5. Select the users that you want to delete, and click the delete icon .
6. To save the changes to the role, click **OK**.

Adding permissions to roles

You can use the IBM Operations Analytics - Log Analysis UI to add permissions to new or existing roles.

Before you begin

To add permissions to a new or existing role, you must have administrative access.

The default permission is none. Users with the default permissions do not have access to the data.

Note: If you run a saved search, or create a dashboard or custom app that includes a list of data sources, the search or dashboard is rendered with data from data sources for which you have permission.

Procedure

To add permissions to a new or existing role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings.**
2. Select the **Roles** tab.
3. Select the role that you want to add permissions to:
   - To add permissions to a new role, click the add icon .

- To add permissions to an existing role, select the role and click the edit icon .

4. Select the **Assign Permissions to Role** tab in the **Add Role** or **Edit <*username*>** pane.
5. Click the add icon .
6. Select the permissions that you want to add to the role from the list, and click **OK**.

   Note: IBM Operations Analytics - Log Analysis only supports read permissions for datasource objects. Users with read permissions can run ingestion and deletion operations.

   To clear a selected permission, hold down the **Ctrl** or **Command** key, and click the permission row.
7. To save the changes to the role, click **OK**.

Deleting permissions from roles

You can use the IBM Operations Analytics - Log Analysis UI to delete permissions from roles.

Before you begin

To delete permissions from roles, you must have administrative access.

Note: If you run a saved search, or create a dashboard or custom app that includes a list of data sources, the search or dashboard is rendered with data from data sources for which you have permission.

Procedure

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings.**
2. Select the **Roles** tab.
3. Select the role from which you want to delete permissions, and click the edit icon .
4. Select the **Assign Permissions to Role** tab in the **Edit <*username*>** pane.
5. Select the permissions that you want to delete, and click the delete icon .
6. To save the changes to the role, click **OK**.

Security considerations

A number of files in the IBM Operations Analytics - Log Analysis environment contain encoded or encrypted passwords. Access to these files must be controlled either through controlling access to the file system or through access rights or file permissions. This can be achieved by limiting the groups and users that have access to these files. The files that contain this information are:

- `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`
- `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml`
- `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityUserRegistry.xml`
- `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/keystore/unity.ks`
- `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/keystore/unity_statistics.ks`

- <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties
- <HOME>/IBM/LogAnalysis/eif_remote_install_tool/config/rest-api.properties
- <HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance
- <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh
- <HOME>/IBM/LogAnalysis/utilities/alerts/alerts.properties

LDAP configuration

You can implement an LDAP user registry in place of the database-managed custom user registry that is provided in IBM Operations Analytics - Log Analysis.

IBM Operations Analytics - Log Analysis uses database-managed custom user registry as the default setting for user authentication. This setting is configured in the server.xml file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory.

You can modify the user registry XML file yourself or you can use the ldapRegistryHelper.sh command that is in the <HOME>/IBM/LogAnalysis/utilities directory to generate an XML file that you can use to enable basic authentication.

Note: IBM Operations Analytics - Log Analysis supports Tivoli Directory Server (TDS) and Microsoft Active Directory (AD) LDAP servers.

Before you begin

- Configure LDAP immediately after IBM Operations Analytics - Log Analysis is installed.
- When LDAP is configured, you cannot revert to database-managed custom user registry.
- The IBM Operations Analytics - Log Analysis administrator, or unityadmin user, must be present in the LDAP repository.
- After the unityadmin user logs in to IBM Operations Analytics - Log Analysis for the first time, they must register other LDAP users with IBM Operations Analytics - Log Analysis by adding users from the **UI**.
- LDAP must be registered by the unityadmin user to use the IBM Operations Analytics - Log Analysis functionality.

Using the ldapRegistryHelper.sh command

You can use the ldapRegistryHelper.sh command to generate the ldapRegistry.xml file. This file is in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory. It contains the minimum properties that are required to enable a basic connection to an LDAP directory server. You can use the file to enable a basic LDAP authentication or as an example to guide you setting up your own LDAP.

You can use one of the following options to configure LDAP authentication:

- To set up LDAP authentication with an IBM Tivoli Directory Server or Microsoft Active Directory server, use the ldapRegistryHelper.sh command to generate the ldapRegistry.xml file. Use this file to enable a basic connection to an LDAP directory server. See “Configuring LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory” on page 38
- To set up LDAP authentication with a Tivoli Directory Server (TDS) or Microsoft Active Directory (AD) server, you must create your own XML file or

modify the ldapRegistry.xml file to enable a basic connection to an LDAP directory server. See “Manually configuring LDAP authentication” on page 42.

Manually enabling LDAP authentication

Use the ldapRegistryHelper.sh command to generate the ldapRegistry.xml file.

Modify the ldapRegistry.xml file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory manually.

To disable database-managed custom user registry, comment out the following line in the server.xml file:

```
<!--Include the basic registry predefined with default users and groups -->
<include optional='true' location="${server.config.dir}/unityUserRegistry.xml"/> -->
```

To enable LDAP authentication, remove the comment from the following line:

```
<!--Include the LDAP registry -->
<include optional='true' location="${server.config.dir}/ldapRegistry.xml"/> -->
```

where ldapRegistry.xml is the file that contains the properties that are required to connect to the LDAP server.

User group names

The valid IBM Operations Analytics - Log Analysis user group names for LDAP are UnityAdmins and UnityUsers. These group names are mapped to security roles in the unityConfig.xml file that is in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory.

To use other LDAP groups, update the unityConfig.xml file to map the groups to security roles in IBM Operations Analytics - Log Analysis.

Compatible LDAP directories

IBM Operations Analytics - Log Analysis supports Tivoli Directory Server (TDS) and Microsoft Active Directory (AD) LDAP servers.

Configuring LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory:

You can use the ldapRegistryHelper.sh command to help you to create and enable an IBM Tivoli Directory Server or Microsoft Active Directory server for IBM Operations Analytics - Log Analysis user authentication.

Before you begin

- The unityadmin user must be a member of the UnityAdmins group in the LDAP registry. Other users must be members of the UnityUsers group.
- To use other LDAP groups, update the unityConfig.xml file to map the groups to security roles in IBM Operations Analytics - Log Analysis.

About this task

Only one LDAP server can be configured by using the utility.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the following command:

   ```
   ./ unity.sh -stop
   ```

2. To specify the LDAP server details, edit the `ldapRegistryHelper.properties` file that is in the `<HOME>/IBM/LogAnalysis/utilities/` directory. For more information about the `ldapRegistryHelper` properties, see the "`ldapRegistryHelper.properties`" on page 41 topic in the *Configuration* guide.
3. Navigate to the <HOME>/IBM/LogAnalysis/utilities directory and run the following command:

   ```
   ./ldapRegistryHelper.sh config
   ```

4. Run the following command:

   ```
   ./ldapRegistryHelper.sh enable
   ```

5. If the `UnityAdmins` or `UnityUsers` groups are not in your LDAP server, you can map other groups in the LDAP registry to security roles in IBM Operations Analytics - Log Analysis. To map groups to security roles, edit the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityConfig.xml` file. For example:

   ```
    <security-role name="UnityUser">
             <group name="UnityUsers" />
             <group name="UnityAdmins" />
             <group name="TestLANonAdmin"/>
             <group name="TestLAAdmin"/>
         </security-role>
         <security-role name="UnityAdmin">
             <group name="UnityAdmins" />
              <group name="TestLAAdmin"/>
         </security-role>
   ```

6. To start the IBM Operations Analytics - Log Analysis server, use the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
   ```

7. To add LDAP users to IBM Operations Analytics - Log Analysis, log in as `unityadmin`.

   Note: You can delete the LDAP user registered with IBM Operations Analytics - Log Analysis but you cannot edit or delete the actual LDAP users.
8. To add roles and permissions to LDAP users, open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**. For more information about adding roles and permissions, see the *Adding users to roles* and *Adding permissions to roles* topics in the *Users and roles* section of the *Configuring guide*.
9. Users who are deleted from the LDAP registry must be removed from IBM Operations Analytics - Log Analysis by the `unityadmin` user to prevent storage of obsolete information in the IBM Operations Analytics - Log Analysis Derby database.

Results

Basic LDAP authentication between IBM Operations Analytics - Log Analysis and the IBM Tivoli Directory Server or the Microsoft Active Directory server is enabled.

What to do next

After you configure LDAP, you must update the password in the configuration files. For more information, see "Updating passwords in the configuration files" on page 45.

Configuring multiple LDAP servers:

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the following command:
   ```
   ./ unity.sh -stop
   ```
2. To specify the LDAP server details, edit the `ldapRegistryHelper.properties` file that is in the `<HOME>/IBM/LogAnalysis/utilities/` directory. For more information about the `ldapRegistryHelper` properties, see the "`ldapRegistryHelper.properties`" on page 41 topic in the *Configuration* guide.
3. Navigate to the <HOME>/IBM/LogAnalysis/utilities directory. Use the `ldapRegistryHelper.sh` command to generate the `ldapRegistry.xml` file and run the following command to generate the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml` file:
   ```
   ./ldapRegistryHelper.sh config
   ```
   Back up the `ldapRegistry.xml` file and repeat this step for each LDAP server.
4. To ensure that the servers are distinct, edit the **`realm`** and **`id`** properties in the `ldapRegistryHelper.properties` file that is in the `<HOME>/IBM/LogAnalysis/utilities/` directory. For more information about the `ldapRegistryHelper` properties, see the "`ldapRegistryHelper.properties`" on page 41 topic in the *Configuration* guide.
5. To map groups in your LDAP servers to security roles in IBM Operations Analytics - Log Analysis, edit the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityConfig.xml`. For example:
   ```
   <server>
     <application type="war" id="Unity" name="Unity"
                location="${server.config.dir}/apps/Unity.war">
       <application-bnd>
          <security-role name="UnityUser"
              <group name="UnityUsers" />
              <group name="UnityAdmins" />
              <group name="TestLANonAdmin"/
              <group name="TestLAAdmin"/
          </security-role>
          <security-role name="UnityAdmin">
              <group name="UnityAdmins" />
          /security-role>
       /application-bnd>
     /application>
     oauth-roles>
       authenticated>
         <group name="UnityUsers"/>
         <group name="TestLANonAdmin"/>
         <group name="TestLAAdmin"/>
       /authenticated>
     /oauth-roles>
   </server>
   ```
6. To start the IBM Operations Analytics - Log Analysis server, use the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
   ```

7. Log in as the `unityadmin` user and add the LDAP users to IBM Operations Analytics - Log Analysis.

   Note: You can delete the LDAP user registered with IBM Operations Analytics - Log Analysis but you cannot edit or delete the actual LDAP users.
8. To add roles and permissions to LDAP users, open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**. For more information about adding roles and permissions, see the *Adding users to roles* and *Adding permissions to roles* topics in the *Users and roles* section of the *Configuring guide*.
9. Users who are deleted from the LDAP registry must be removed from IBM Operations Analytics - Log Analysis by the `unityadmin` user to prevent storage of obsolete information in the IBM Operations Analytics - Log Analysis Derby database.

ldapRegistryHelper.properties:

You can edit the `ldapRegistryHelper.properties` to specify LDAP server details.

The following properties are required and define the connection information for the target LDAP server.

Table 5. LDAP server connection information properties

Property	Description
`ldap_hostname_property=`	The LDAP hostname.
`ldap_port_property=`	The LDAP port.
`ldap_baseDN_property=`	The LDAP baseDN. For example, `"dc=com"` for TDS users, and `"CN=Users,DC=sflab,DC=local"`for AD users.

The following properties are optional and define the connection information for the target LDAP server. Where applicable, default settings are assigned.

The **`bindPassword`** value for AD users is encrypted in the **`ldapRegistryHelper_config.xml`**.

Table 6. Optional LDAP server connection information properties

Property	Description
`ldap_bindDN_property=`	The LDAP bindDN. For example, `"CN=Administrator,CN=Users,DC=sflab, DC=local"` for AD users.
`ldap_bindPassword_property=`	The LDAP bind password.
`ldap_realm_property=`	The LDAP realm. The default value is `LdapRegistryRealm`.
`ldap_id_property=`	The LDAP ID. The default value is `LdapRegistryId`.
`ldap_ignoreCase_property=`	The LDAP ignore case. The default value is `true`.

`ldapRegistryHelper.sh` **command:**

You can use the `ldapRegistryHelper.sh` command to enable a basic connection for user authentication in IBM Operations Analytics - Log Analysis.

For more information about how to use the command to set up LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory, see "Configuring LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory" on page 38.

Supported integrations

This command currently supports connections to the IBM Tivoli Directory Server and Microsoft Active Directory.

Prerequisites

Before you use this command, you must update the `ldapRegistryHelper.properties` file in the `<HOME>/IBM/LogAnalysis/utilities/` directory with the connection and configuration information for the target LDAP server.

Syntax

The `ldapRegistryHelper.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
 ldapRegistryHelper.sh    config | enable
```

Note:

To run the script, the `JAVA_HOME` variable must be set correctly for IBM Operations Analytics - Log Analysis. If the script fails, run the following command to set the `JAVA_HOME` variable:

```
JAVA_HOME=$<HOME>/IBM-java
```

Parameters

The `ldapRegistryHelper.sh` command has the following parameters:

`config` Use the `config` parameter to create an XML file that is called `ldapRegistry.xml` in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory. This file uses the connection and configuration information that is defined in the `ldapRegistryHelper.properties` file.

`enable`

Use the `enable` parameter to enable LDAP authentication that uses the information that is specified in the `ldapRegistry.xml` file. This parameter also disables the reference to the database-managed custom user registry.

Manually configuring LDAP authentication:

If you want to manually configure LDAP authentication, you can manually configure the settings in your own XML file or you can modify the `ldapRegistry.xml` that is output by the `ldapRegistryHelper.sh` command to meet your requirements.

About this task

The following procedure describes the steps that are automated by the ldapRegistryHelper.sh command. Read this procedure to help you understand the necessary steps for configuring LDAP authentication. For more information, see Configuring an LDAP user registry with the Liberty profile.

Procedure

1. Manually create an LDAP configuration file that is named ldapRegistry.xml and save it in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory or modify the ldapRegistry.xml that is output by the ldapRegistryHelper.sh command.
2. Update the ldapRegistry.xml with the appropriate configuration information:
   - For IBM Tivoli Directory Server, add the text:

     ```
     <ldapRegistry id="IBMDirectoryServerLDAP" realm="SampleLdapIDSRealm"
             host="host.domain.com" port="389" ignoreCase="true"
             baseDN="o=domain,c=us"
             bindDN="cn=root"
             bindPassword="password"
             ldapType="IBM Tivoli Directory Server">
             <idsFilters
                 userFilter="(&amp;(uid=%v)(objectclass=ePerson))"
                 groupFilter="(&amp;(cn=%v)(|(objectclass=groupOfNames)
     (objectclass=groupOfUniqueNames)(objectclass=groupOfURLs)))"
                 userIdMap="*:uid"
                 groupIdMap="*:cn"
                 groupMemberIdMap="ibm-allGroups:member;ibm-allGroups:
     uniqueMember;groupOfNames:member;groupOfUniqueNames:uniqueMember" />
         </ldapRegistry>
     ```

   - For Microsoft Active Directory, add the text:

     ```
     <ldapRegistry id="ActiveDirectoryLDAP" realm="SampleLdapADRealm"
             host="host.domain.com" port="389" ignoreCase="true"
             baseDN="cn=users,dc=domain,dc=com"
             bindDN="cn=myuser,cn=users,dc=domain,dc=com"
             bindPassword="password"
             ldapType="Microsoft Active Directory" />
     ```

3. Update these attributes to reflect your LDAP server configuration:
   - ID
   - realm
   - host
   - port
   - baseDN
   - bindDN
4. AD users must run the securityUtility command that is in the <HOME>/IBM/LogAnalysis/bin directory to encode the bindPassword password. This step is optional for TDS users as they do not require the bindPassword password.

   After you run the command, copy the encrypted value that is output by the command to the bindPassword property.

   For more information about this command, see "unity_securityUtility.sh command" on page 205
5. (Optional) If your implementation uses a Microsoft Active Directory LDAP that uses different object classes to define users and groups, update the userFilter and groupFilter attributes as required.

6. (Optional) If your implementation uses Microsoft Active Directory, update the user and group mapping attributes as required for your LDAP environment.
7. Open the server.xml file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory and add these lines:
   a. Only one type of user registry can be configured for authentication, therefore you must disable the database-managed custom user registry to enable LDAP. Comment out the following lines in the server.xml file that reference the database-managed custom user registry:

      ```
      <!-- Include the basic registry predefined with default users
      and groups -->
      <!--    <include optional="true" location="${server.config.dir}/
      unityUserRegistry.xml"/>
      -->
      ```

      If you do not remove this reference, an error message is displayed.
   b. Add an include tag to replace the reference to the custom user registry with a reference to the ldapRegistry.xml file. For example:

      ```
      <!-- Include the LDAP registry for user and groups -->
          <include optional="true" location="${server.config.dir}/
      ldapRegistry.xml"/>
      ```

8. If the UnityAdmins or UnityUsers groups are not in your LDAP server, you can map other groups in the LDAP registry to security roles in IBM Operations Analytics - Log Analysis. To map groups to security roles, edit the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityConfig.xml file. For example:

   ```
    <security-role name="UnityUser">
             <group name="UnityUsers" />
             <group name="UnityAdmins" />
             <group name="TestLANonAdmin"/>
             <group name="TestLAAdmin"/>
         </security-role>
         <security-role name="UnityAdmin">
             <group name="UnityAdmins" />
              <group name="TestLAAdmin"/>
         </security-role>
   ```

9. To start the IBM Operations Analytics - Log Analysis server, use the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
   ```

10. To add LDAP users to IBM Operations Analytics - Log Analysis, log in as unityadmin.

    Note: You can delete the LDAP user registered with IBM Operations Analytics - Log Analysis but you cannot edit or delete the actual LDAP users.
11. To add roles and permissions to LDAP users, open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**. For more information about adding roles and permissions, see the *Adding users to roles* and *Adding permissions to roles* topics in the *Users and roles* section of the *Configuring guide*.
12. Users who are deleted from the LDAP registry must be removed from IBM Operations Analytics - Log Analysis by the unityadmin user to prevent storage of obsolete information in the IBM Operations Analytics - Log Analysis Derby database.

What to do next

After you configure LDAP user registry, you must update the unityadmin password in the IBM Operations Analytics - Log Analysis configuration files. For more information, see "Updating passwords in the configuration files."

Adding custom groups to the OAuth security role:

To fully enable LDAP authentication, you need to add the OAuth security role to the LDAP server.

Procedure

1. Open the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/UnityConfig.xml file.
2. Add the following code:
   ```
   <oauth-roles>
       <authenticated>
         <group name="UnityUsers"/>
         <group name="<Group_name1>/>"
         <group name="<Group_name2>/>"
       </authenticated>
   </oauth-roles>
   ```
   where *<Group_name>* are the names of any custom groups that you use. The UnityUsers role is added by default.
3. Save your changes.
4. To start the IBM Operations Analytics - Log Analysis server, use the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
   ```

Updating passwords in the configuration files:

In most cases, after you create or change a user or password in your Lightweight Directory Access Protocol (LDAP) application, you do not need to update the passwords in IBM Operations Analytics - Log Analysis. However, if the new or changed password is specified in the IBM Operations Analytics - Log Analysis configuration files, you must update the files with the new or changed information.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the unity.sh script that is in the <HOME>/IBM/LogAnalysis/utilities:
   ```
   ./ unity.sh -stop
   ```
2. Optional: If you do update or add a password, you can encrypt the password. To encrypt it, run the unity_securityUtility.sh script that is in the <HOME>/IBM/LogAnalysis/utilities directory. For more information, see "unity_securityUtility.sh command" on page 205.
3. If you change the password that is used by the unityadmin, you must update the encrypted password in the following files to match the updated password. To generate the encrypted password use the unity_securityUtility.sh utility in the <HOME>/IBM/LogAnalysis/utilities directory. For example:
   ```
   unity_securityUtility.sh encode password
   ```
   - <HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties. For example:
     ```
     #The password to use to access the unity rest service
     password={aes}EF712133E0677FEBB30624BA5EE62BC2
     ```
   - <HOME>/IBM/LogAnalysis/remote_install_tool/config/rest-api.properties. For example:

```
ibm.scala.rest.password={aes}EF712133E0677FEBB30624BA5EE62BC2
```

- `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`. For example:

  ```
  unity.data.collector.password={aes}EF712133E0677FEBB30624BA5EE62BC2
  ```

- `<HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance.sh`. For example:

  ```
  PASSWD={aes}EF712133E0677FEBB30624BA5EE62BC2
  ```

4. If you change the password that is used by the `unityadmin`, you must update the `password` parameter in the `<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh` script.

   ```
   password={aes}7A0B2401A8E29F37CD768CB78E205CAD
   ```

5. To start the IBM Operations Analytics - Log Analysis server, use the `unity.sh` script that is in the `<HOME>/IBM/LogAnalysis/utilities` directory:

   ```
   ./ unity.sh -start
   ```

Configuring single sign-on (SSO) with the Tivoli Integrated Portal

You can configure SSO authentication between the Tivoli Integrated Portal and IBM Operations Analytics - Log Analysis.

Before you begin

- The Tivoli Integrated Portal server and the IBM Operations Analytics - Log Analysis server must use the same LDAP server for authentication.
- The Tivoli Integrated Portal server must use a Lightweight Directory Access Protocol (LDAP) server for authentication.
- You must configure SSO for the Tivoli Integrated Portal. To configure SSO:
  1. Log in to the Tivoli Integrated Portal server
  2. In the **Security** area, click **Global security**.
  3. In the **Authentication** area, click **Single-sign on (SSO)**.
  4. Ensure that the **Enabled** check box is selected.
  5. The domain value that you must have to complete step 4 is displayed in the **Domain name** field. If this field is blank, enter the domain name and click **Apply**.

Procedure

1. To export the Lightweight Third-Party Authentication (LTPA) keys file from the Tivoli Integrated Portal, complete the following steps:
   a. Log on to the Tivoli Integrated Portal as an administrator.
   b. In the **Security** area, click **Global security**.
   c. In the **Authentication** area, click **LTPA**.
   d. In the **Cross-cell single sign on** area, enter a password for the keys file in the **Password** field. Confirm the password.
   e. Create a blank plain text file to use as the keys file. Note the directory that you store the file in.
   f. Enter the location where the keys file that you created in the previous step is stored in the **Fully qualified key file name** field. The value must point to the properties file that contains the keys that you want to export. For example, for a Windows operating system, enter `C:\keys.properties`. For a Unix-based operating system, enter `<`*`tip_home_dir`*`>/profiles/TIPProfile`.
   g. Click **Export keys**.
2. Add the Tivoli Integrated Portal LDAP realm to the IBM Operations Analytics - Log Analysis LDAP configuration. Ensure that the LDAP realm that is specified

here is the same as the one used by Tivoli Integrated Portal. To specify the realm, edit the `ldap_realm_property` property in the `ldapRegistryHelper.properties` file:

```
ldap_realm_property=<LdapRegistryRealm>
```

where <*LdapRegistryRealm*> is the realm that is used by the Tivoli Integrated Portal. To find this value:

a. Log on to the Tivoli Integrated Portal.
b. In the **Security** area, click **Global security**.
c. In the **User account repository** area, click **Configure**.
d. The LDAP realm value is displayed in the **Realm name** field. You specify this same value in the `ldapRegistryHelper.properties` file.

3. To add the updated realm to the LDAP configuration for IBM Operations Analytics - Log Analysis and to enable LDAP authentication, run the `ldapRegistryHelper.sh` script. For more information, see "`ldapRegistryHelper.sh` command" on page 42.
4. Configure LTPA on the Liberty Profile for the WebSphere Application Server:
   a. Copy the LTPA keys file that you exported from the Tivoli Integrated Portal server in step 1 to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security` directory on the IBM Operations Analytics - Log Analysis server. The folder contains a default keys file. Do not change this file. Use a different name for your own key file.
   b. Go to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory.
   c. To add the SSO tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:

```
<webAppSecurity ssoDomainNames="<SSO_domain>" />
```

   where <*SSO_domain*> is the SSO domain, for example `example.com`. This value must match the SSO domain that is used by the Tivoli Integrated Portal server. Specify the same value as the one that is entered in the **Domain name** field on the Tivoli Integrated Portal UI.

   d. To add the LTPA tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:

```
<ltpa keysFileName="${server.output.dir}/resources/security/<ltpa_key_file>"
keysPassword="<keysPassword>" expiration="120" />
```

   - where <*ltpa_key_file*> is the LTPA key file, for example `example_ltpa.keys`.
   - <*keysPassword*> is the LTPA password that you entered in step 1 when you created the LTPA key file on the Tivoli Integrated Portal server.

   (Optional) You can use the `unity_securityUtility` command that is in the `<HOME>/IBM/LogAnalysis/wlp/bin/` directory to generate an encrypted password. After you generate the encrypted password, enter it as the value for the `keysPassword` parameter.
5. Restart the IBM Operations Analytics - Log Analysis server and verify that the SSO connection between the two servers is working.

Results

To verify that the SSO connection is correctly set up, log in to the Tivoli Integrated Portal server. Open a new tab page in the browser and log in to IBM Operations Analytics - Log Analysis. If you are not prompted for the user name and password, the SSO connection is set up correctly. If you are prompted for the login details, the SSO connection is not configured correctly.

Configuring single sign-on (SSO) with Jazz for Service Management

If you want to integrate data from IBM Operations Analytics - Log Analysis with the Dashboard Application Services Hub component of Jazz for Service Management, you need to configure SSO between IBM Operations Analytics - Log Analysis and Jazz for Service Management.

Before you begin

- Jazz for Service Management server must use a Lightweight Directory Access Protocol (LDAP) server for authentication.
- Jazz for Service Management server and the IBM Operations Analytics - Log Analysis server must use the same LDAP server for authentication.
- You must configure SSO for the Jazz for Service Management server. For more information, see the Configuring SSO on the application server topic in the Jazz for Service Management Knowledge Center at http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome?lang=en.

Procedure

1. Export the Lightweight Third-Party Authentication (LTPA) keys file from the Jazz for Service Management server. For more information, see the Exporting LTPA keys topic in the Jazz for Service Management Knowledge Center at http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome?lang=en.
2. Add the Jazz for Service Management LDAP realm to the IBM Operations Analytics - Log Analysis LDAP configuration. Ensure that the LDAP realm that is specified here is the same as the one used by Jazz for Service Management. To specify the realm, edit the `ldap_realm_property` property in the `ldapRegistryHelper.properties` file:

   ```
   ldap_realm_property=<LdapRegistryRealm>
   ```

   where <*LdapRegistryRealm*> is the realm that is used by Jazz for Service Management.
3. To add the updated realm to the LDAP configuration for IBM Operations Analytics - Log Analysis and to enable LDAP authentication, run the `ldapRegistryHelper.sh` script. For more information, see "`ldapRegistryHelper.sh` command" on page 42.
4. Configure LTPA on the Liberty Profile for the WebSphere Application Server:
   a. Copy the LTPA keys file that you exported from the Jazz for Service Management server in step 1 to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security` directory on the IBM Operations Analytics - Log Analysis server. The folder contains a default keys file. Do not change this file. Use a different name for your own key file.
   b. Go to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory.
   c. To add the SSO tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:

   ```
   <webAppSecurity ssoDomainNames="<SSO_domain>" />
   ```

   where <*SSO_domain*> is the SSO domain, for example `example.com`. This value must match the SSO domain that is used by the Jazz for Service Management server. Specify the same value as the one that is entered in the **Domain name** field on the Jazz for Service Management UI.

d. To add the LTPA tag to the IBM Operations Analytics - Log Analysis server, add the following line to the server.xml file before the final server tag:

```
<ltpa keysFileName="${server.output.dir}/resources/security/<ltpa_key_file>"
keysPassword="<keysPassword>" expiration="120" />
```

- where *<ltpa_key_file>* is the LTPA key file, for example example_ltpa.keys.
- *<keysPassword>* is the LTPA password that you entered in step 1 when you created the LTPA key file on the Tivoli Integrated Portal server.

(Optional) You can use the unity_securityUtility command that is in the <HOME>/IBM/LogAnalysis/wlp/bin/ directory to generate an encrypted password. After you generate the encrypted password, enter it as the value for the keysPassword parameter.

5. Restart the IBM Operations Analytics - Log Analysis server and verify that the SSO connection between the two servers is working.

Results

To verify that the SSO connection is correctly set up, log in to the Jazz for Service Management server. Open a new tab page in the browser and log in to IBM Operations Analytics - Log Analysis. If you are not prompted for the user name and password, the SSO connection is set up correctly. If you are prompted for the login details, the SSO connection is not configured correctly.

System

Before you can use Log Analysis, you need to configure the system.

Configuring the data archive

To optimize performance, configure the length of time that IBM Operations Analytics - Log Analysis stores data in the archive.

About this task

You must configure the archive period before you load any data.

To facilitate query performance, IBM Operations Analytics - Log Analysis uses data tiers to prioritize the retrieval of the most recent information.

IBM Operations Analytics - Log Analysis divides data into two tiers, the current tier and the archive tier. Data is held in the current tier for the specified period. After this time elapses and more data is loaded, it is moved to the archive tier. Data that is stored in the current tier is stored in memory. Therefore, the queries can access this data more quickly than data in the archive tier.

You use the HOT_TIER_PERIOD parameter to specify the number of days that data is held in the current tier. For example, if the HOT_TIER_PERIOD parameter is set to 2, data is held in the current tier for two days until the next ingestion of data.

The length of the time period that you specify for the HOT_TIER_PERIOD parameter affects the amount of memory that IBM Operations Analytics - Log Analysis uses. The longer the period, the greater the memory used.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

2. Open the `unitysetup.properties` file that is in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
3. To specify the number of days that data is stored in the current tier, change the value for the `HOT_TIER_PERIOD` parameter. The default value is 2:

   ```
   HOT_TIER_PERIOD=2
   ```
4. Save the file.
5. To start the IBM Operations Analytics - Log Analysis server, use the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
   ```

Results

Data is held in the current tier until the next data ingestion for the specified period.

Configuring Hadoop for long term data storage

Standard

Before you can use Hadoop for long term data storage, you must configure the integration with IBM Operations Analytics - Log Analysis.

Why should I use Hadoop?

Hadoop offers a more efficient method for long term data storage that you can use to store long term data from annotated log files. The integration with IBM Operations Analytics - Log Analysis is facilitated by a service that is bundled with IBM Operations Analytics - Log Analysis 1.3.1 and ensures that you can continue to search this data without need to store the data in the main database.

Hadoop integrations

There are two Hadoop integration options available to IBM Operations Analytics - Log Analysis users.

IBM InfoSphere® BigInsights® Hadoop
IBM InfoSphere BigInsights delivers a rich set of advanced analytics capabilities that allows enterprises to analyze massive volumes of structured and unstructured data in its native format. For more information, see the IBM InfoSphere BigInsights documentation at http://www-01.ibm.com/support/knowledgecenter/SSPT3X/SSPT3X_welcome.html

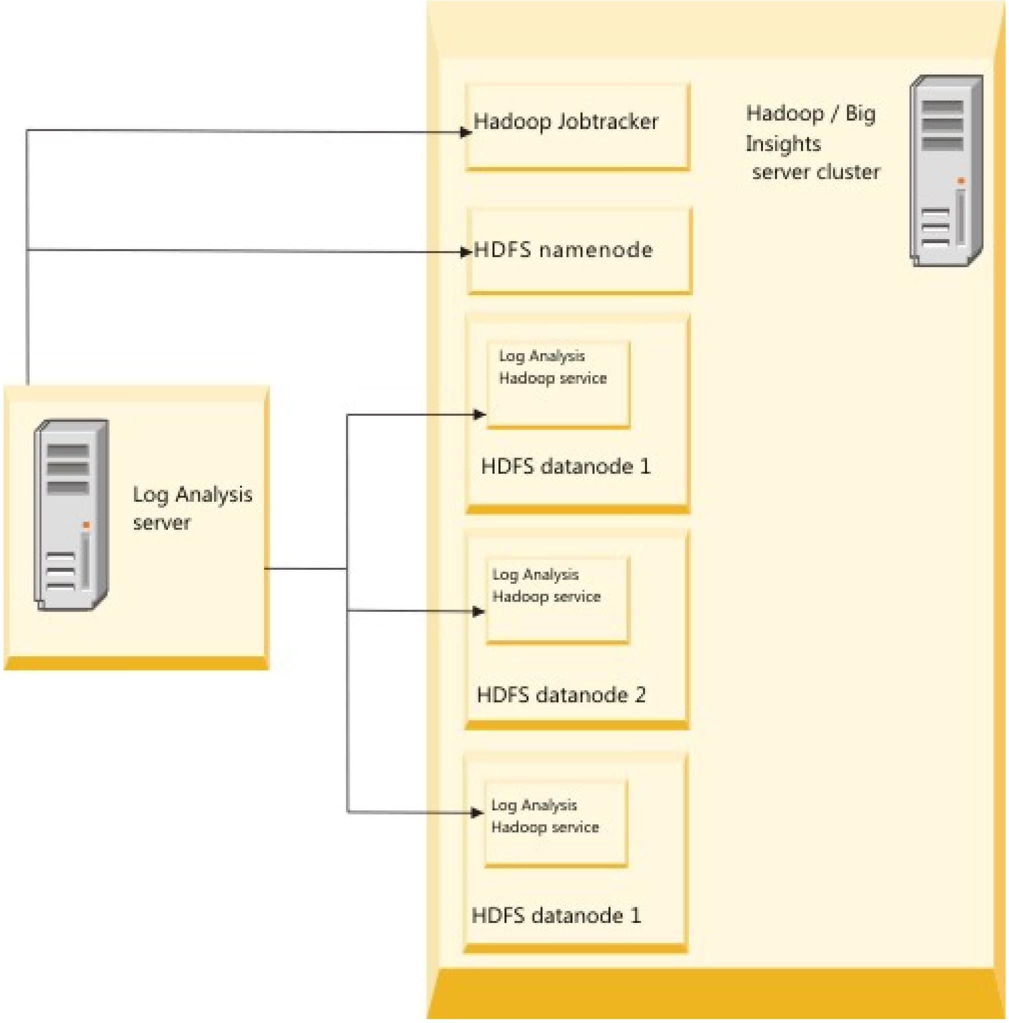
Cloudera Hadoop
Cloudera provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of data in your enterprise. For more information, see the Cloudera documentation at http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/introduction.html

Architecture

IBM Operations Analytics - Log Analysis stores the log data on to the Hadoop cluster and allows users to search data that is stored in Hadoop. IBM Operations Analytics - Log Analysis, with the help of the IBM Operations Analytics - Log Analysis service that is installed on each datanode, writes the data to the Hadoop cluster. The data is written in the `avro` object container file format. For more information about object container files, see http://avro.apache.org/docs/1.7.4/spec.html#Object+Container+Files. Data is then written to each datanode where the service is installed. You can use IBM Operations Analytics - Log Analysis to search this data.

You can also run IBM Operations Analytics - Log Analysis searches on the data stored on the Hadoop cluster.

The following graphic displays an overview of the service architecture:


Hadoop Jobtracker
Hadoop / Big Insights server cluster
HDFS namenode
Log Analysis Hadoop service
HDFS datanode 1
Log Analysis server
Log Analysis Hadoop service
HDFS datanode 2
Log Analysis Hadoop service
HDFS datanode 1

Prerequisite tasks: Standard

To enable the Hadoop tier in IBM Operations Analytics - Log Analysis, complete the prerequisite tasks.

Complete these tasks before you configure the Hadoop tier.

Hadoop cluster

Ensure that a IBM InfoSphere BigInsights Hadoop or Cloudera Hadoop is installed in your environment, and that the IBM Operations Analytics - Log Analysis

Hadoop tier was tested with IBM InfoSphere BigInsights Hadoop 3.0.0.1 or Cloudera Hadoop 5.3.0.

IBM Operations Analytics - Log Analysis

Ensure that IBM Operations Analytics - Log Analysis is installed.

Datanode username configuration

1. Create a user for each datanode on the Hadoop cluster.
2. Ensure that the username for each datanode is the same as the IBM Operations Analytics - Log Analysis username on the IBM Operations Analytics - Log Analysis server.

Enable the Hadoop tier: Standard

Before you can use Hadoop for long-term data storage, you must enable the Hadoop tier.

Integrating the Hadoop client: Standard

Before you can enable the Hadoop tier, you must prepare the Hadoop client.

Procedure

1. Create a folder that is called `hadoop-jars` on the IBM Operations Analytics - Log Analysis server.
2. Copy the Hadoop client from the Hadoop cluster. Choose one of the following options for your chosen Hadoop integration for IBM Operations Analytics - Log Analysis.
   - IBM InfoSphere BigInsights Hadoop
     a. Open the IBM InfoSphere BigInsights administration webpage. For example, `http://<BI_Cluster_Manager_Host>:8080`
     b. Click **Download client library and development software**.
     c. To download the client package, select **Job Submission API package**.
     d. Download and extract the client package.
     e. Copy the `jar` and `xml` files from the client package to the `hadoop-jars` folder created in step 1.

     Note: If IBM Operations Analytics - Log Analysis is installed on Linux on System z based operating system with IBM InfoSphere BigInsights 3.0.0 on x-86, you must replace the `hadoop-core.jar` with the `hadoop-core-2.2.0-mr1.jar`. The `hadoop-core.jar` was copied from the IBM InfoSphere BigInsights Hadoop cluster. The `hadoop-core-2.2.0-mr1.jar` is supplied with IBM Operations Analytics - Log Analysis, and located here: `<HOME>/IBM/LogAnalysis/utilities/hadoop/BigInsights_3.0.0`
   - Cloudera Hadoop
     a. Navigate to the Cloudera parcels folder on the Cloudera Hadoop cluster. For example: `/opt/cloudera/parcels/CDH-5.3.0-1.cdh5.3.0.p0.30/lib/hadoop/client`
     b. Copy the `jar` files from the Cloudera parcels folder to the `hadoop-jars` folder created in step 1. For information about obtaining jar files, see the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_vd_hadoop_api_dependencies.html

c. Open the cluster managers webpage. For example, `http://<CDH_Cluster_Manager_Host>:7180/cmf/home`.

d. Download the **Client Configuration Files** for `YARN` services. For more information about downloading the client configuration files, see the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_mc_client_config.html

e. Extract and copy the `xml` files to the `hadoop-jars` folder created in step 1.

Setting up the Hadoop tier: Standard

Before you can enable the Hadoop tier, you must set up the Hadoop tier.

Procedure

1. Create a folder that is called `LA_HADOOP_TIER` on the `Hadoop Distributed File System` (`HDFS`). For example, to use the command-line utility to create the folder, run the following command:

   ```
   hadoop fs -mkdir /<la-hadoop-tier>
   ```

   This folder is exclusively used by the IBM Operations Analytics - Log Analysis Hadoop tier.

2. Create the following folders in the `LA_HADOOP_TIER` folder.

   `LA_HADOOP_TIER`

   This folder is used to ingest the data.

   `LA_HADOOP_TIER/jars`

   This folder is used to store the `jar` files required by the map-reduce job.

   `LA_HADOOP_TIER/output`

   This folder is used as temporary storage during a IBM Operations Analytics - Log Analysis search query execution over the Hadoop tier.

   For example, to use the command-line utility to create these folders, enter the following commands in the command-line:

   ```
   hadoop fs -mkdir /la-hadoop-tier/data
   hadoop fs -mkdir /la-hadoop-tier/jars
   hadoop fs -mkdir /la-hadoop-tier/output
   ```

3. Change the ownership of these folders to the `LA` use. For example, to use the command-line utility to change the ownership of the folders created in step 2 to the `LA` user, enter the following command:

   ```
   ./hadoop fs -chown -R LA:LA /la-hadoop-tier
   ```

4. Verify the creation and ownership of these folders. For example, to use the command-line utility to verify folder details, enter the following commands:

   ```
   hadoop fs -ls /
   hadoop fs -ls /la-hadoop-tier
   ```

5. Copy and extract the `search.zip` from `<HOME>/IBM/LogAnalysis/utilities/hadoop` to a temporary folder on the Hadoop cluster. For example: `/tmp/la-search-jars`

6. Upload the `jars` from the temporary folder to HDFS. For example, to use the command-line utility to load the files from the temporary folder to HDFS, enter the following command:

   ```
   hadoop fs -copyFromLocal /tmp/la-search-jars/*.jar /la-hadoop-tier/jars
   ```

7. To ensure that the `LA` user can launch a MapReduce job to the Hadoop cluster, log in as the `LA` user and launch a test map-reduce job.

For more information about how to launch a map-reduce test on the Cloudera Hadoop, see *Running a MapReduce Job* in the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_ig_testing_the_install.html

Installing the IBM Operations Analytics - Log Analysis service: Standard

The IBM Operations Analytics - Log Analysis Hadoop tier requires a IBM Operations Analytics - Log Analysis service on each datanode server in the Hadoop cluster.

About this task

The IBM Operations Analytics - Log Analysis server pushes the log data to the IBM Operations Analytics - Log Analysis service. The data is then written to the `HDFS`.

Procedure

To install the IBM Operations Analytics - Log Analysis service, complete the following steps.

1. Log in as the `LA` user to one of the datanodes in the Hadoop cluster.
2. Create a folder that is called `LA_SERVICE_HOME`. For example, `<HOME>/IBM/LogAnalysis/LA_SERVICE_HOME`
3. Copy and extract the `service.zip` from `<HOME>/IBM/LogAnalysis/utilities/hadoop` to the `LA_SERVICE_HOME` folder.
4. Copy the `LA_HADOOP_TIER/jars` folder to the `LA_SERVICE_HOME/lib` folder.
5. Review, and modify if required, the values for the environment variables in `LA_SERVICE_HOME/bin/env.sh`.
6. Copy the `LA_SERVICE_HOME` folder to all datanodes in the Hadoop cluster.

Setting up IBM Operations Analytics - Log Analysis for the Hadoop tier: Standard

Before you can enable the Hadoop tier, you must set up IBM Operations Analytics - Log Analysis for the Hadoop tier.

Procedure

1. Stop the IBM Operations Analytics - Log Analysis server. For example, to use the command-line utility to stop the IBM Operations Analytics - Log Analysis server, enter the following command:

   ```
   ./ unity.sh -stop
   ```

2. Open the *`unitysetup.properties`* in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory. Change the following properties:

   - `HADOOP_TIER_HDFS_BASE_DIR`

     Specify the value of `<`*`LA_HADOOP_TIER`*`>` on `HDFS`.

   - `INDEX_IMPLEMENTATION=`*`SOLR`*

     Add Hadoop to the `INDEX_IMPLEMENTATION` property. For example, change `INDEX_IMPLEMENTATION=`*`SOLR`* to `INDEX_IMPLEMENTATION=`*`SOLR`*`,` *`HADOOP`*

   Update default `HADOOP_TIER.properties` if different from your environment. Ignore the `HADOOP_TIER_JOB_TRACKER_URI` property.

3. Copy the hadoop-jars folder to the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/lib directory
4. Start the IBM Operations Analytics - Log Analysis server. For example, to use the command-line utility to start the IBM Operations Analytics - Log Analysis server, enter the following command:

   ```
   ./ unity.sh -start
   ```

Testing the IBM Operations Analytics - Log Analysis Hadoop tier: Standard

To ensure that the IBM Operations Analytics - Log AnalysisHadoop tier is correctly set up and configured you can run some basic tests.

Procedure

1. Ingest log data to the IBM Operations Analytics - Log Analysis server.
   a. You can install sample data on the IBM Operations Analytics - Log Analysis UI, https://*<LA_server>*:9987/Unity. To ensure that log data is correctly ingested, you can also ingest logs from the Insight Pack that will be used in the set up.
   b. Ingestion results in writing the log data in avro files on the HDFS in the *<LA_HADOOP_TIER>*/data folder in the form *<UnityCollection_Timestamp>/<DATA_SOURCE_NAME>/<DAY_BASED_ON_TIMESTAMP_IN_LOG_RECORDS>/<counter>*.avro.
2. Perform searches on the Hadoop tier through the IBM Operations Analytics - Log Analysis web UI, https://*<LA_server>*:9987/Unity.
   a. Prepend the search query in the UI with [_hq]. For example, [_hq]*
   b. The search on the Hadoop tier is run via a map reduce job on the Hadoop cluster.
   c. Prepend the same search query in the UI with [_sq], for example, [_sq]*, and perform the query on Apache Solr.
   d. Compare the search results.
3. To identify errors, open the following files:
   - UnityApplication.log on the IBM Operations Analytics - Log Analysis server.
   - <HOME>/IBM/LogAnalysis/logs/hadooptier.log on the IBM Operations Analytics - Log Analysis server.
   - *<LA_SERVICE_HOME>*/logs on the IBM Operations Analytics - Log Analysis services server.

Managing the Hadoop service on datanodes: Standard

After you configure the Hadoop service, you can use the server.sh script to manage the service.

Procedure

You can choose to manage the service on an individual datanode or manage all of the service instances together.

- To manage the service on an individual datanode, use the LA user that you created when you configured the service to log in to a Hadoop datanode server.
  1. Run the *<LA_SERVICE_HOME>*/bin/server.sh script with one of the following parameters:

     Start Starts the service on the Hadoop datanode server.

Stop Stops the service on the Hadoop datanode server.

Status Retrieves the status for the Hadoop datanode server.

- To manage all of the instances, select one datanode to act as a `LA_Service_Controller_Node`. This will manage the service on all of the datanodes.
  1. (Optional) Create password-less SSH for the `LA` user from this datanode to all of the datanodes, including this datanode, in the Hadoop cluster.
  2. Use the `LA` user to login to the `LA_Service_Controller_Node` datanode.
  3. Run the `<`*`LA_SERVICE_HOME`*`>/bin/server.sh` script with one of the following parameters:

     clusterStart
     Starts the service on each Hadoop datanode server.

     clusterStop
     Stops the service on each Hadoop datanode server.

     clusterStatus
     Retrieves the status for each Hadoop datanode server.

  If you do not configure password-less SSH connections during the configuration, you are prompted for the password for each datanode server.

Sharing a Hadoop cluster across multiple IBM Operations Analytics - Log Analysis instances: Standard

Procedure

1. To share a Hadoop cluster across multiple IBM Operations Analytics - Log Analysis instances, you must integrate the Hadoop service for each IBM Operations Analytics - Log Analysis instance.

   For more information, see the *Integrating the Hadoop service* topic in *Configuration* guide.

   a. You must use a different value for each of the following folders:

   ```
   <LA_HADOOP_TIER> on the HDFS datanode
   <LA_SERVICE_HOME> in the LA home directory
   ```

   Alternatively to repeating the steps for each IBM Operations Analytics - Log Analysis instance, you can create a copy of the resultant folders from a IBM Operations Analytics - Log Analysis instance of the same version.
2. Modify the `PORT` and `PROCESS_ID` values in the <*LA_SERVICE_HOME*>/bin/env.sh file

Globalization

Some of the text in IBM Operations Analytics - Log Analysis is only available in English. Some of these texts can be globalized but others cannot. Read the following information to get an overview of what you can and cannot manually globalize.

To manually globalize some of the content that is only available in English by default, you can create a resource bundle. For more information, see Globalizing Insight Packs and Custom Apps.

Limitations

The following data cannot be globalized:

Log file data

Log file data is always displayed in the language that it is used in the original log files. This text is displayed in the **Search** UI.

Installation directory and files

The directory where IBM Operations Analytics - Log Analysis is installed and the files that are created during the installation are only available in English.

IBM Operations Analytics - Log Analysis log files

The log files that are generated by IBM Operations Analytics - Log Analysis and the associated applications such as IBM Tivoli Monitoring Log File Agent, Apache Solr, and the WebSphere Application Server are only available in English.

Artifacts that are created in the Admin UI

Artifacts such as Source Types, Collections, Data Sources, File Sets, and Rule Sets are only available in English. If a user creates one of these objects in another language, the artifacts are only available in that language.

Sample log files

The log files that are used by the sample scenarios are only available in English.

Insight Packs

Only the following Insight Packs are available in languages other than English by default. If you want to localize other content, you need to configure this manually. The index configurations that are used by these Insight Packs are only available in English and are not globalized.

- DB2InsightPack_v1.1.0.2
- WASInsightPack_v1.1.0.3
- JavacoreInsightPack_v1.1.0.3
- GenericAnnotationInsightPack_v1.1.1.2
- SyslogInsightPack_v1.1.0.3
- WebAccessLogInsightPack_v1.1.0.2

IBM Operations Analytics - Log Analysis scripts

Scripts such as `unity.sh`, `eifutil.sh`, and `ldapRegistryHelper.sh` are only available in English and cannot be globalized. This limitation encompasses Shell, Python, and Ruby scripts.

Time and date formats

The time and date format can only be changed for the entire IBM Operations Analytics - Log Analysis installation.

Globalizing Custom Apps:

Complete this procedure to globalize Custom App to ensure that messages, errors and exceptions, and the custom application generated output is globalized.

About this task

To enable globalization of the Custom App generated output, the IBM Operations Analytics - Log Analysis framework passes the locale that is used by your application.

For information about the limitations of the globalization process, see "Globalization" on page 56.

To globalize your Custom App, complete the following steps:

Procedure

1. Open the script that the Custom App is based on. In most cases, this script is the `<custom_app_name>.app` file that is stored in the relevant Insight Pack folder, for example `<HOME>/IBM/LogAnalysis/unity_content/WindowsOSEventsInsightPack_<version>/unity_apps/apps`. The Custom App can be based on other languages, such as Python and Java.
2. To extract the locale information that is passed from IBM Operations Analytics - Log Analysis to your Custom App, you must add a JSON compatible code to your Custom App script. For example:

   ```
   {
   "parameters":[
   {}
   ],
   "_fwParameters":[
   {
   "name": "locale",
   "value": "<locale>",
   "type": "String"
   }
   ]
   }
   ```

   The following example is in the Python language and it shows how you can extract the locale information that is passed to the Insight Pack or Custom App from IBM Operations Analytics - Log Analysis:

   ```
   if len(sys.argv) > 1:
     filename = str(sys.argv[1])
     fk = open(filename,"r")
     data = json.load(fk)
     locale = data["_fwParameters"][0]["value"]
   ```

   In this example, the locale is sent to the script from IBM Operations Analytics - Log Analysis in the following format:

   ```
   {
      "parameters":[
         {
         }
   ],
   "_fwParameters":[
     {
       "name": "locale",
       "value": "<locale>",
       "type": "String"
         }
     ]
   }
   ```

   where *<locale>* is the locale that you want to use. For example, `"value": "en_US"`.
3. Save and start the application to see the globalized Custom App.

Results

The extracted locale globalizes messages, errors and exceptions, and the custom application generated output.

Globalizing dashboards, chart labels, and index configuration fields:

Complete this procedure to globalize index configuration fields, dashboards, and chart labels.

About this task

Dashboard, charts and index configuration fields are globalized by using a resource bundle. The resource bundle is based on the Java resource bundle mechanism and Insight Pack developer needs to create the resource bundle in supported languages. There is one resource bundle for the Insight Pack that contains keys for all the artifacts.

The following Insight Packs are available in languages other than English by default:

- DB2InsightPack_v1.1.0.2
- WASInsightPack_v1.1.0.3
- JavacoreInsightPack_v1.1.0.3
- WindowsOSEventsInsightPack_v1.1.0.3
- GenericAnnotationInsightPack_v1.1.1.2
- SyslogInsightPack_v1.1.0.3
- WebAccessLogInsightPack_v1.1.0.2

All other Insight Packs are only available in English. If you want to globalize these objects and the associated content into another language, you must configure the locale and create a resource bundle.

To globalize your dashboards, charts, and index configuration fields complete the following steps:

Procedure

1. Create a folder that is named `i18n` in the directory where your Insight Pack is stored.
2. Create a resource bundle file in the `i18n` folder. The Insight Pack name that you use in the resource bundle file must match the Insight Pack exactly.

   ```
   <Insight_Pack_Name>_locale.properties
   ```

   where *<Insight_Pack_Name>* is the exact name of the Insight Pack that you want to globalize.
3. Specify the keys for each artifact that you want to globalize. Keys are values of the fields to be translated for globalization. For example, the value of the "name" field for your custom application or dashboard in the `.app` file. If you do not specify a value in the resource bundle file, the existing name is used by default. Keys that are not specified in the resource bundle are displayed in English. The resource bundle supports three different types of specification:

   Global keys

   ```
   key=value
   ```

   Global key applies to all artifacts in the Insight Pack.

   Global artifact keys

   ```
   artifact_type.key=value
   ```

   Global artifact key applies to all artifacts under a specific artifact type.

Specific artifact keys

```
artifact_type.artifact_name.key=value
```

Specific artifact keys override the general key.

The following artifact_types are supported:

- sourcetype
- customapp

The artifact name is the name that you created, for example WASSystemOut.

You can specify globalized text for each of the following artifacts:

Index configuration

To globalize the field names in the index configuration, specify the field names and the localized text in the resource bundle:

```
sourcetype.<Source_Type_name>.<Index_Configuration_Field> = Localized_text
```

For example:

```
sourcetype.WASSystemOut.severity = severity_translated
sourcetype.WASSystemErr.message = message_translated
sourcetype.WASTrace.msgclassifier = msgclassifier_translated
```

Custom App name

To specify a globalized version of a Custom App name, you must specify the name that is used in the Custom App specification file in the resource bundle. For example, the name in the Custom App specification file is:

```
"name":"Error Analysis"
```

You specify the following information in the resource bundle:

```
customapp.<custom_app_name>.Error\ Analysis = Log Severity Trend
```

Log Severity Trend is displayed as the application name on the UI.

Tags in the Search Dashboard

When you create a Custom App, it is displayed in the **Search Dashboards** pane in the **Search** UI. The Custom App is grouped under a directory. The name of this directory is globalized by using tags. To globalize this tag name, you need to specify the tag in the resource bundle as:

```
customapp.tag.<Insight_Pack_Name> = SEVERITY ANALYSIS
```

where <*Insight_Pack_Name*> is the name of the Insight Pack.

For example:

```
customapp.tag.NewAppPack_v1.1.0.0 = SEVERITY ANALYSIS
```

The tag name SEVERITY ANALYSIS is displayed on the UI.

Chart titles

The chart titles are specified as follows in the chart specification:

```
spec: "title": "Event Diagnostics"
```

To globalize the title, add it to the resource bundle:

```
customapp.<custom_app_name>.Event\ Diagnostics = Error while ingesting data
```

Labels for chart axis

The labels that are used to identify chart axis are specified by the `label` parameter in the chart specifications. For example:

```
"parameters": [{"type": "Axis", "name": "xaxis", "label":"xlabel"},
{"type": "Axis", "name": "yaxis", "datatype":"number",
"label":"ylabel"},],
```

The new axis labels from chart specification should be used in your Custom App specification.

```
"charts": [{ "title": "Event Diagnostics", "labels":
{"xlabel":"timestamp", "ylabel":"Counter"},]
```

To globalize the label names, specify the globalized text in the resource bundle in the following format:

```
customapp.<custom_app_name>.<Label_name> = <Localized_name>
```

where *<Label_name>* is the name that is specified in the chart specifications. *<Localized_name>* is the name that you want to display on the UI. For example:

```
customapp.<custom_app_name>.Counter = Counter_t
```

Counter_t is the axis name that is displayed on the UI.

4. Package the Insight Pack as the next version and use `pkg_mngt.sh` utility to update the Insight Packs.

Enabling facet cache for wildcard searches

If you use the wildcard search term (*) to search data that is older than 1 day, you can configure Log Analysis to count facets before they are indexed for search. This setting can help optimize this type of search.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file.
2. Change the `ENABLE_SOLR_FACET_CACHE=false` parameter to true.
3. Save the file.
4. To restart Log Analysis, enter the following command:

```
./<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

Configuring scalable data streaming from multiple, remote sources

To facilitate dynamic data streaming that is scalable across multiple remote sources, you must configure IBM Operations Analytics - Log Analysis after you install it.

To enable data collection from remote hosts, you must complete the following steps:

1. Install Apache Solr on the remote machine.
2. Set up Secure Shell (SSH) communication.
3. Configure SSH to work with the remote installer utility.
4. Use the remote installer utility to install instances of the Event Integration Facility (EIF) or the IBM Tivoli Monitoring Log File Agent (LFA) on remote machines.

5. Configure the EIF so that it is compatible with the remote instances that your create. If you use the LFA, you do not have to configure the local installation. However, you do have to manually configure the sub nodes.

You can also maintain and administer these connections after you set them up.

As an alternative to streaming data, You can batch load data. For more information, see “Loading and streaming data” on page 223.

Installing Apache Solr on remote machines:

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

About this task

If no local instances of Apache Solr exist, then you need to install the instances on the remote machine as soon as you install IBM Operations Analytics - Log Analysis. If there is a local instance of Apache Solr, you can install the remote instances whenever you want.

You must use a non-root user to run the script.

You cannot use the installer to install Apache Solr on a local machine.

You cannot use the installer to install multiple Apache Solr nodes on a single remote machine.

To install Apache Solr on multiple remote machines, run the script separately for each remote machine. You cannot use the installer to install instances of Apache Solr simultaneously or in parallel.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:
   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```
2. To run the `remote_deploy.sh` script, enter the following command:
   ```
   ./remote_deploy_solr.sh -install
   ```
3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password-less SSH authentication is disabled. If password-less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the <HOME>/IBM/LogAnalysis/utilities/config directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

SSH Port
Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

Top-level Installation Directory
To use the default value, which is `<HOME>`, press enter. Alternatively, you can enter the path to the directory where you want to install the DE.

Apache Solr Search Port
To use the default value, 9989, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

Apache Solr Query Service Port
To use the default value, 7205, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

4. To start the installation, press enter. In most cases, the installation takes about 5 minutes to complete.

Results

The results of the installation are output in the log file in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs/ManageSolrnodes.log` file.

To view the status for the instances of Apache Solr that are installed remote machines, run the `unity.sh -status` command.

Example

Here is an example script output:

```
Remote Hostname in FQDN format:12345.example.com
username:unity
password:*********
SSH port: [22]
Top-level Installation Directory: [/home/unity]
Solr Search Port: [9989]
Solr Query Service Port: [7205]

Script is ready for remote installation of Solr:
Review the following inputs ....
-----------------------------------------------------------------------------
Remote Host Name: 12345.example.com
Remote User Name: unity
Remote SSH Port: 22
Top-level remote installation directory: /home/unity
Solr v9.0 - remote installation directory:
/home/unity/IBM/LogAnalysis
Solr - remote ports: 9989, 7205
------------------------------------------------------------------------
['q' - Abort]['Enter' - Install]

Sat Nov 16 03:08:38 CST 2013 Starting remote installation of Solr
, this will take couple of minutes to complete  ....
Sat Nov 16 03:08:38 CST 2013 Waiting for remote installation to complete ....
Sat Nov 16 03:11:47 CST 2013 Successfully installed Solr
Solr on remote host:12345.example.com ....
```

Removing Apache Solr instances:

Before you remove an installation of IBM Operations Analytics - Log Analysis, you must remove Apache Solr.

About this task

Note: Do not remove Apache Solr if IBM Operations Analytics - Log Analysis is still being used. IBM Operations Analytics - Log Analysis does not function properly when any instances of Apache Solr are removed. For this reason, only remove Apache Solr when you are about to uninstall IBM Operations Analytics - Log Analysis.

If you installed Apache Solr locally and remotely, remove the local instance first, then remove the remotely installed instances.

This process uses Installation Manager to remove Apache Solr instances. You can also do so silently. To run the silent removal, run following `imcl -c` command, enter `3` to modify the installation, and remove the instance.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:
   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```
2. To run the `remote_deploy.sh` uninstall script, enter the following command:
   ```
   ./remote_deploy.sh -uninstall
   ```
3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password less SSH authentication is disabled. If password less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the `<UNITY_HOME>/utilities/config` directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

   SSH Port
   Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

   Top-level Installation Directory
   To use the default value, which is `<HOME>/IBM/LogAnalysis`, press enter. Alternatively, you can enter the path to the directory where Apache Solr is installed.
4. To start the removal, press enter. You can view the logs in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs` directory.

Results

When all the remote nodes are removed, you can safely uninstall IBM Operations Analytics - Log Analysis.

Setting up Secure Shell to use key-based authentication:

Secure Shell (SSH) is a cryptographic network protocol for secure data communication between different computers. You set up key-based authentication between the IBM Operations Analytics - Log Analysis servers and the remote computers to which it connects.

About this task

Benefits of using key-based authentication:
- Data is transferred across a secure channel.
- The administrator is no longer concerned about the password changes for the remote servers.
- The passphrase is independent of the individual server password policy.
- One passphrase is used for multiple servers. Only the public key file must be copied to the client server.

For more information you can view the man pages for **ssh-keygen** by running this command:

```
man ssh-keygen
```

Procedure

1. To generate public and private keys, enter the following command:

   ```
   ssh-keygen -t rsa
   ```

   or either of the following commands:

   ```
   ssh-keygen
   (This command generates the same results as ssh-keygen -t rsa.)
   ```

   ```
   ssh-keygen -t dsa
   (If you specify dsa, the generated keys include _dsa in their file names.)
   ```

   The following example shows what a valid output might look like:

   ```
   bash-3.2$
   bash-3.2$ ssh-keygen -t rsa
   Generating public/private rsa key pair.
   Enter file in which you want to save the key (/home/unity/.ssh/id_rsa):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /home/unity/.ssh/id_rsa.
   Your public key has been saved in /home/unity/.ssh/id_rsa.pub.
   The key fingerprint is:
   4a:ef:d5:7a:d8:55:b3:98:a1:1f:62:be:dd:c4:60:6e unity@<variable>.example.com
   The key's randomart image is:
   +--[ RSA 2048]----+
   |                 |
   |                 |
   |                 |
   |             . ..|
   |      . S   .o+.o|
   |     . o   =o++. |
   |      . . +o+E.o |
   |       . ..o=.o  |
   |        . .o.. . |
   +-----------------+
   bash-3.2$
   ```

Enter the passphrase. (The **Enter passphrase** field can remain blank to specify an empty passphrase.)

2. To view the contents of the public key file, run the following commands:

   ```
   cd ~/.ssh
   ls -l id_rsa*
   cat id_rsa.pub
   ```

   The command output is:

   ```
   bash-3.2$
   bash-3.2$ cat .ssh/id_rsa.pub
   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDg0/GGoxGzyC7Awjbwnp0hCaeztIRt6yhAg
   GKdwM7nb7Iiv0RgwT4/48E26K1Ur9HrI1W/j0K0JHQw
   vaAFibqeLmqLdK9ctCE901ywTOPFcYeBYPUF9vp/MgaypgGxVwDbW/e0SNPb7YAtZpjRoqeUq
   oYoKzFXXspQkxdhcQfpx0RYMbQdGGg03hDCM2wr2KP
   VuTVniF2IvDu1C4fcRkUPr8aQNMiuEcJgV3VHhlau/0Uo0YpH53NXKhn/sx8xdyTVsKQ1rhW8
   g07HIVc2Tf9ZF2gYXn/HbjE509xK/APu2nztt0h+Air
   JyT5jYMi/IvSI0zbPyc0p9WijPeG8r/v unity@<variable>.in.ibm.com
   bash-3.2$
   ```

3. Create a directory called `.ssh` on the remote server. Use this to store the public key.
4. Copy the public key file (`id_rsa.pub`) to the `.ssh` directory on the remote client:

   ```
   scp /home/unity/.ssh/id_rsa.pub
   <username>@<remotehostname>:/
   <HOME>/.ssh/id_rsa.pub
   ```

   where <*hostname*> is the system host name and <*username*> is the system user name.

5. Add the content of the public key to the `authorized_keys` file on the remote host.

   ```
   bash-3.2$ ssh <username>@<remotehostname>
   bash-3.2$ cd ~/.ssh
   bash-3.2$ cat id_rsa.pub >> authorized_keys
   bash-3.2$ rm id_rsa.pub
   bash-3.2$ exit
   ```

6. Ensure that there are no duplicate keys for the same client in the authorized_keys file.
7. Log in to the remote computer to ensure that key-based SSH is working:

   ```
   ssh <username>@<hostname>
   ```

   Enter the passphrase, if prompted.

   ```
   bash-3.2$ bash-3.2$ ssh <username>@<remotehostname>
   Enter passphrase for key '/home/unity/.ssh/id_rsa':
   Last unsuccessful login: Mon Jul 15 14:22:37 2013 on ssh from <variable>.example.com
   Last login: Mon Jul 15 14:26:54 2013 on ssh from <variable>.example.com
   $
   ```

   Configuration of key-based authentication is complete.

Results

The steps may not work because different versions of SSH are supported by the operating systems that are used by the remote servers. For more information about how to solve this issue, see the *Secure Shell (SSH) configuration does not work* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Configuring secure shell (SSH) communication for multiple remote hosts:

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

Before you begin

Before you configure SSH for multiple remote hosts, you must configure SSH between IBM Operations Analytics - Log Analysis and the remote hosts. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the Information Center.

About this task

By default, the SSH properties file, `ssh-config.properties` file, is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory. If you save the file to another location, the utility requests that the user enters values for the remote host, user, and password. In this case, the utility does not use the values specified in the file.

If you save the `ssh-config.properties` file in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory, the `eif_remote_install_tool` utility uses the properties specified in the file.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

If you specify values for both the password and the private key file path, the utility uses the file to create a password-less SSH connection.

If you do not specify a value for the password or the private key file path, IBM Operations Analytics - Log Analysis cannot create a connection and instead generates an error message in the log:

```
    ERROR:
    example.unity.remote.SshConfigException:
Property file config/ssh-config.properties must contain at least one of:
PASSWORD, PATH_OF_PASSWORD_LESS_SSH_KEY
    Correct SSH configuration OR reconfigure and retry
    Installation Aborted....!
```

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory and open the `ssh-config.properties` file.
2. Specify values for the following properties for each remote host:
   - Remote host
   - Remote user ID
   - Port
   - Connection timeout in milliseconds. The default is 6000.

   For example:

```
REMOTE_HOST=<REMOTE_HOST>
PORT=<PORT>
TIME_OUT=60000
USER=<REMOTE_USER>
```

3. For password-based authentication, you also need to specify the password in the configuration file. For example:

   ```
   PASSWORD=password1
   ```
4. For public key based authentication, specify the path to the directory that contains the private key file. For example:

   ```
   PATH_OF_PASSWORD_LESS_SSH_KEY=/home/pass/.ssh/id_rsa
   ```
5. If your installation of SSH requires a passphrase, specify the passphrase. For example:

   ```
   PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=passphrase1
   ```

Configuring data collection for scalability on multiple remote nodes:

To facilitate scalable data collection on multiple remote nodes, use the `install.sh` command to install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server.

Before you begin

Before you run the command, you must configure secure shell (SSH) communication between the local installation of IBM Operations Analytics - Log Analysis and the remote host. For more information about how to do so, see "Configuring secure shell (SSH) communication for multiple remote hosts" on page 66.

About this task

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis.

You can use the remote installer in the following scenarios:

- If you have a high rate of data ingestion on multiple data sources. For example, if you have 100 or more events per second and 20 or more data sources.
- If you require improved throughput performance on the remote server.
- If the hardware resources on the remote server are restrained.
- If you want to optimize performance according to the conditions described on the Performance developer works page here: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM Log Analytics Beta/page/Performance and tuning

You can use the command to deploy up to 20 instances of the Tivoli Event Integration Facility Receiver or a single instance of the IBM Tivoli Monitoring Log File Agent on a remote node. The command deploys and configures IBM Java™ 1.7. The command also configures the deployed Tivoli Event Integration Facility Receiver instance to communicate with the IBM Operations Analytics - Log Analysis Data Collector interface.

However, this command does not configure the IBM Tivoli Monitoring Log File Agent subnode. You must configure this setting manually. Both the remote and local instance of the IBM Tivoli Monitoring Log File Agent can monitor remote data sources. For more information about configuring IBM Tivoli Monitoring Log File Agent, see "Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data" on page 233.

To ensure that the remote instances of the Tivoli Event Integration Facility work with the local Data Collector interface, you must create the remotely deployedTivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances as part of the same installation. This is because the encryption configuration and signature generation is done during the main installation. If you install IBM Operations Analytics - Log Analysis after you set up the remote nodes, you must install the remote Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances again. However, you can remove remote instances of the Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent without installing IBM Operations Analytics - Log Analysis again.

Note: If you use the script to install the remote instance on a server that uses the SUSE Linux Enterprise Server 11 operating system, the script fails. To resolve this issue, see the *Cannot install remote EIF instance on SUSE* topic in the *Troubleshooting* IBM Operations Analytics - Log Analysis guide.

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory and run the `install.sh` command. You are prompted for a series of inputs.
2. Enter the remote installation directory. This value must be the location where the deployed artifacts are installed on the remote host.
3. If you want to deploy the Tivoli Event Integration Facility Receiver, select it. If you do, enter the Tivoli Event Integration Facility Receiver instances that you want to deploy.
4. If you want to deploy the IBM Tivoli Monitoring Log File Agent instance on the remote node, select it.

Results

After you complete the procedure, you can now collect data from the remote hosts.

What to do next

After the initial setup, you will want to periodically change the configuration. IBM provides two commands to start and stop the instances so that you can update the configuration.

To administer Tivoli Event Integration Facility Receiver instances, use the `eifutil.sh` command.

To administer IBM Tivoli Monitoring Log File Agent instances, use the `lfautil.sh` command.

`eifutil.sh` command:

To administer EIF Receiver instances, use the `eifutil.sh` command.

Syntax

The `eifutil.sh` command has the following syntax and is in the *<USER_HOME_REMOTE>*`/DataForwarders/EIFReceivers/utilities` where *<USER_HOME_REMOTE>* is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where <*`Inst_ID`*> is the ID for the specific EIF instance.

Parameters

`-status`
Displays the status for the installed instances. For example:

```
============================================================================
COMPONENT            Instance         PID            PORT           STATUS
============================================================================
EIF Receiver         eif_inst_1      13983           6601              UP
EIF Receiver         eif_inst_2      14475           6602              UP
EIF Receiver         eif_inst_3      14982           6603              UP
EIF Receiver         eif_inst_4      15474           6604              UP
EIF Receiver         eif_inst_5      15966           6605              UP
============================================================================
```

`-start <`*`Inst_id`*`>`
Starts the specified instance.

`-stop <`*`Inst_id`*`>`
Stops the specified instance.

`-startAll`
Starts all instances.

`-stopAll`
Stops all instances.

`-restart`<*Inst_id*>
Restarts the specified instance.

`-restartAll`
Restarts all the instances.

`lfautil.sh` command:

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the `lfautil.sh` command.

Syntax

The `lfautil.sh` command has the following syntax and is in the *<USER_HOME_REMOTE>*`/utilities/` directory on the remote host where *<USER_HOME_REMOTE>* is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start Starts all the LFA instances on the remote host.

-stop Stops all the LFA instances on the remote host.

-status
Displays the status for the LFA instances on the remote host. For example:

```
==========================================
COMPONENT            PID            STATUS
==========================================
Log File Agent        23995             UP
==========================================
```

-restart
Restarts the LFA instances on the remote host.

Configuring email notifications for the data ingestion limit

Entry

If you are using the Entry edition of Log Analysis, you must configure the Simple Mail Transfer Protocol (SMTP) server so that you can receive email notifications as you reach the data ingestion limit.

About this task

After you configure the email settings, a single email notification is sent when you reach 50%, 70%, and 90% of the daily limit of 2 gigabytes (GBs).

You can use the `IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/configs/IngestionAlertEmailConfig.json` example file to help you to create your own.

Procedure

1. Create a JSON file and note the directory where you save it.
2. Add values for the SMTP server host name, the sender's email address and the recipients email address. The structure is as follows:

   ```
   {
       "smtpMailServer": "<SMTP_server_hostname>",
       "secure": false,
       "from": "<Sender_email_address>",
       "to": ["<Recipients_email_address>"],
       "cc": [<cc_recipients_email_address>],
       "bcc": [<bcc_recipients_email_address>],
       "subjectPrefix": "Log Analysis Ingestion Alert!",
       "header": "Hi,",
       "footer": "*** This is a system generated e-mail,
        please do not reply to this e-mail ***\n",
       "attachLogRecordAnnotations": false
   }
   ```

   where<*SMTP_server_hostname*> is the name of the SMTP server that you want to use to send the mail. <*Sender_email_address*> is the email address for the SMTP server user who sends the mail.

   For example:

   ```
   {
       "smtpMailServer": "1234.example.com",
       "secure": false,
   ```

```
    "from": "johndoe@example.com",
    "to": ["janedoe@example.com"],
    "cc": [],
    "bcc": [],
    "subjectPrefix": "Log Analysis Ingestion Alert!",
    "header": "Hi,",
    "footer": "*** This is a system generated e-mail,
     please do not reply to this e-mail ***\n",
    "attachLogRecordAnnotations": false
}
```

3. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file and add the `INGESTION_ALERT_MAIL_CONFIG_FILE` property, specifying the directory where you saved the JSON file in step 1:

   `INGESTION_ALERT_MAIL_CONFIG_FILE = <`*`JSON_file_directory`*`>`

   For example:

   ```
   INGESTION_ALERT_MAIL_CONFIG_FILE = /home/unity/IBM/LogAnalysis/wlp/usr/servers/
   Unity/apps/Unity.war/configs/IngestionAlertEmailConfig.json
   ```
4. Save the file.
5. Restart Log Analysis.

Results

An email notification is sent to the specified addresses when you reach 50%, 70%, and 90% of the daily limit.

If you do not configure the email settings correctly, an error is created in the Generic Receiver log file.

Configuring alerts

Standard

You can use the alert management features of Log Analysis to monitor real-time data ingestion and trigger events based on specified conditions.

Use the alert management feature to help you to complete the following tasks:

- Generate alerts in real time as data is streamed into Log Analysis.
- Specify conditions and trigger actions such as sending an email notification, running a custom script, or logging an alert. You can also specify custom actions.
- Detect user specified conditions when the data is streamed into Log Analysis and trigger custom actions when the conditions are met.
- Index the generated alerts for reporting.

You can configure and create alerts in one of four ways:

Alerts UI
Use the Log Analysis user interface (UI) to create, edit, and delete alerts, alert actions, and conditions.

Command-line utility
Use the command line utility to create, edit, and delete alerts, alert actions, and conditions.

JSON template
Edit the JSON template to create, edit, and delete alerts, alert actions, and conditions.

REST API
Use the REST API for Alerts to create, edit, and delete alerts, alert actions, and conditions.

Conditions

To create an alert, you need to define the conditions that trigger the alert action. There are two types of condition:

Base condition
To trigger alerts based on a single data source and log record, you define a base condition. For example, to send an email notification when the response time exceeds 5 minutes for a specified data source and log record, you define a base condition.

Composite conditions
To trigger alerts that are based on time windows, the frequency of base conditions, or multiple data sources, you define a composite condition.

Actions

If a condition is met, IBM Operations Analytics - Log Analysis triggers an action. Built-in actions are:

Email notifications
An email notification is sent to one or more users when a condition is met.

Debugging Log Analysis alerts
Log Analysis logs alerts to a file. You can use this type of action to debug your alerting implementation. To debug alerts and verify that they are working correctly, you can use an alert action template to log events to a specified log file. For more information about how to use this feature, see the blogs at https://developer.ibm.com/itoa/blog/.

Indexing
An alert is indexed when a condition is met. You can use this action to search for alerts that occurred in a specific time period or to build alert dashboards from the search UI.

Script You can invoke an external script when a condition is met.

Configuring alerts in the Manage Alerts UI

Standard

Use the **Manage Alerts** user interface (UI) to create, edit, and delete alerts, alert actions, and conditions.

Creating alerts

Standard

To trigger actions based on events, create an alert.

Procedure

1. To open the **Manage Alerts** UI, click the **Manage Alerts** icon (📢).
2. To create an alert, click **Create New**.
3. Enter a name.
4. To ensure that the alert is active, select the **Enabled** check box.
5. Select a severity level. When this level is reached, the alert action is triggered.
6. Optional: You can also enter a note.
7. To create a condition, click **Create condition**
8. Select a template and enter the required keywords or search queries. The table summarizes the conditions.

Table 7. Conditions

Condition	Description
Keyword match	Use this condition to trigger an alert when a keyword is found in all or only in the specified data sources. You can enter a keyword or you can enter a search query.
Keyword match based on threshold	Use this condition to trigger an alert when a specified number of keywords or search results occur in all or specified data sources during a specified time period. Enter the keyword or search query, specify the number of occurrences that are required to trigger the alert in the time period that you specify in seconds, minutes, or hours.
Keyword match with de-duplicates	Use condition to trigger an alert when a keyword or search query occurs in all or the specified data sources during the specified time period. If a condition is met, a single alert is sent. Log Analysis does not send multiple, duplicate actions.
Co-Occurence Match	Use this condition to trigger an alert when 2 or more keywords or query search results occur during the same specified time period.

9. Save the condition.
10. Select the action that occurs when the alert is triggered. The options are explained in the table.

Table 8. Alert actions

Action	Description
Index	Use this action to index any alerts in the **_alerts** data source. This does not require any configuration. The **_alerts** data source contains 3 indexed fields: **`conditionName`** The name of the condition. **`conditionsDatasource`** The name of the data source which met the condition. **`timestamp`** The time that the condition was met.
Send Email	Use this action to send an email. You can select a template or enter the sender, receiver, subject prefix and body text manually. If you want to send the mail to multiple recipients, you need to separate each address with a comma. Before you can use the email action, you need to configure the `<HOME>/wlp/usr/servers/Unity/apps/Unity.war/configs/AlertActionEmailConfig.json` file.
Write to Log	Use this action to record triggered alerts in a specific log file. You can select a recently viewed log file or you can enter the log path file manually. The triggered alerts are updated every 10 seconds.
Script	Use this action to run a script when a condition is met. You can select a recently viewed script or you can enter the details manually. You enter the directory where the script is stored, the directory where the script is run, and any command line parameters that need to be passed to the script when it is run.

11. Click **Create**.

Manage Alerts UI

Standard

Use the **Manage Alerts** user interface (UI) to create, edit, and delete alerts, conditions, and alert actions.

Buttons, fields, and check boxes

Table 9. Buttons, fields and check boxes on the **Manage Alerts** *UI*

Button, field or check box	Description
Create New	Create an alert.
Edit	Edit an existing alert.
Delete	Delete an existing alert.

Table 9. Buttons, fields and check boxes on the **Manage Alerts** *UI (continued)*

Button, field or check box	Description
Search box	Search for an existing alert.
Refresh icon	Refresh the results table to include any new alerts.

Columns

Table 10. Columns on the Manager Alerts UI

Column	Description
Selection check box	Use this check box to select an alert for editing.
Status	Indicates whether the alert is active or inactive.
Alert Name	The name of the alert.
Severity	The level of severity.
Author	The person who created the alert action template.
Condition Template	The condition template that is used by the alert.
Actions	The actions that are triggered by the alert.

Alerts editor

Table 11. Fields and check box on the Alerts editor

Fields and check box	Description
Alert Name	Enter a name for the alert.
Last Modified	The time and date when the alert was last modified. This field is read only.
Enabled check box	To deactivate the alert, clear this check box.
Severity	Select the severity level of the alert.
Author	The user who created the alert. This field is read only.
Notes	Enter any notes that you want to add.

Conditions editor

Table 12. Conditions

Condition	Description
Keyword match	Use this condition to trigger an alert when a keyword is found in all or only in the specified data sources. You can enter a keyword or you can enter a search query.

Table 12. Conditions (continued)

Condition	Description
Keyword match based on threshold	Use this condition to trigger an alert when a specified number of keywords or search results occur in all or specified data sources during a specified time period. Enter the keyword or search query, specify the number of occurrences that are required to trigger the alert in the time period that you specify in seconds, minutes, or hours.
Keyword match with de-duplicates	Use condition to trigger an alert when a keyword or search query occurs in all or the specified data sources during the specified time period. If a condition is met, a single alert is sent. Log Analysis does not send multiple, duplicate actions.
Co-Occurence Match	Use this condition to trigger an alert when 2 or more keywords or query search results occur during the same specified time period.

Alert actions editor

Table 13. Alert actions

Action	Description
Index	Use this action to index any alerts in the **_alerts** data source. This does not require any configuration. The **_alerts** data source contains 3 indexed fields: **`conditionName`** The name of the condition. **`conditionsDatasource`** The name of the data source which met the condition. **`timestamp`** The time that the condition was met.
Send Email	Use this action to send an email. You can select a template or enter the sender, receiver, subject prefix and body text manually. If you want to send the mail to multiple recipients, you need to separate each address with a comma. Before you can use the email action, you need to configure the `<HOME>/wlp/usr/servers/Unity/apps/Unity.war/configs/AlertActionEmailConfig.json` file.
Write to Log	Use this action to record triggered alerts in a specific log file. You can select a recently viewed log file or you can enter the log path file manually. The triggered alerts are updated every 10 seconds.

Table 13. Alert actions (continued)

Action	Description
Script	Use this action to run a script when a condition is met. You can select a recently viewed script or you can enter the details manually. You enter the directory where the script is stored, the directory where the script is run, and any command line parameters that need to be passed to the script when it is run.

Configuring email alert actions

Standard

Before you can use the email alert action, you need to specify the Simple Mail Transfer Protocol (STMP) server host name in Log Analysis.

About this task

The email alert action feature is compatible with normal and secure STMP servers.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/Unity.war/configs/AlertActionEmailConfig.json` file.
2. Specify the host name of the STMP server:

   ```
   {
    "smtpMailServer":"<mail_server_host_name>"
   }
   ```

   where *<mail_server_host_name>* is the host name of the STMP server.
3. Save the file.

Example email alert

Standard

The email here is an example of one sent by the email alert action feature. The sender, receiver, subject prefix and body text that are specified in the alert action configuration.

```
Dear User,
Alert condition d1-severity was triggered at time 2015-01-09T16:45:00.000Z for the
datasource(s) d1.

The following log record caused the alert condition to trigger:[01/9/15 11:45:00:000 -0500]
00000010 TraceResponse E DSRA1120E: Application2 did not explicitly close all handles to
this connection. Connection cannot be pooled.

*** This is a system generated e-mail, please do not reply to this e-mail ***
```

Monitoring irregular field values with alerts

Standard

You can configure Log Analysis to trigger an alert when it detects changes in the value of a specified field in a data source.

For example, you can configure Log Analysis to check the severity field every 10 minutes. If a change is detected in the specified time period, Log Analysis sends an email to responsible parties to warn them about the possible issue.

To implement this feature, modify the example base condition file and use the command line utility to implement it. You cannot use the **Manage Alerts** UI to create these alerts.

This feature is best suited for monitoring fields like error codes that have a small range of distinct values. If you monitor a field with a large range of values, you can generate too many alerts.

Configuring alerts to monitor irregular field values

Standard

Before you can use Log Analysis to monitor irregular field values in a data source, you need to create and implement the base condition.

Procedure

1. To create a base condition, open the `<HOME>/IBM/LogAnalysis/utilities/alerts/anomalyBaseCondition.json` file and specify a name for the base condition, description, and data source name.
2. To configure the field that you want to monitor and how often this field is checked, specify values for the `resetPeriod` and `fieldName` parameters. For example, the following base condition checks the `severity` field and creates an alert if the value changed in the last 20 minutes:
   ```
   resetPeriod: 20m
   fieldName: severity
   ```
3. To create the base condition, enter the following command:
   ```
   ./alerts.sh -createBaseCondition anomalyBaseCondition.json
   ```
4. To specify an alert action, you can create a alert action JSON file, modify an existing template like `alertTemplate.json` or use an existing alert action JSON file.
5. After you create the alert action file, use the command line utility to create an alert action that uses the condition that you created in the previous steps. For example:
   ```
   ./alerts.sh -createAlertAction <alert_action>.json
   ```

   where `<`*`alert_action`*`>.json` is the name of the JSON file template.

Results

When the value of the monitored field changes, Log Analysis creates an alert.

Configuring alerts with the command-line utility

Standard

In addition to the Log Analysis UI, you can also use the command-line utility to create and manage alerts.

To administer the alerts, use the `alerts.sh` command-line utility. For example, to view all the action templates, enter the following command:

```
alerts.sh –getAlertActionTemplate
```

If you create an alert with the command line utility, then you must use the utility to administer it. You cannot use the **Manage Alerts** UI to edit or delete it.

Creating alerts with the command line utility

Standard

You can use the command line utility to create composite and base conditions and alert actions. If you want to monitor anomalous fields in a data source, you must use the command line utility.

Procedure

1. Create a base or composite condition JSON file and specify the required values. You can use one of the templates in the `<HOME>/IBM/LogAnalysis/utilities/alerts/` folder or you can create your own. If you create a new file, ensure that you save it in the `<HOME>/IBM/LogAnalysis/utilities/alerts/` folder.
2. To create the base condition, enter the following command:

   ```
   ./alerts.sh -createBaseCondition <basecondition>.json
   ```

   To create a composite condition, enter the following command:

   ```
   ./alerts.sh -createCompositeCondition <compositecondition>.json
   ```

   where `<`*`basecondition`*`>.json` and `<`*`compositecondition`*`>.json` is the name of the file that you created in step 1.
3. Create an alert action JSON file and specify the required values. You can use one of the templates in the `<HOME>/IBM/LogAnalysis/utilities/alerts/` folder or you can create your own. If you create a new file, ensure that you save it in the `<HOME>/IBM/LogAnalysis/utilities/alerts/` folder.
4. To create the alert action, enter the following command:

   ```
   ./alerts.sh -createAlertAction <alertaction>.json
   ```

Results

When the condition is met, an alert is created according to the logic specified in the `<`*`alertaction`*`>.json` file.

`alerts.sh` command-line utility

Standard

Use the `alerts.sh` command-line utility to manage the alerting features.

Syntax

The `alerts.sh` utility is in the `<HOME>/IBM/LogAnalysis/utilities/alerts` directory. The syntax is:

```
alerts.sh <parameter>
```

Parameters

Add one of the following parameters to the syntax to complete the action.

Alert action templates

<alert_action_template_file> is the name of the file that contains details of the template in JSON format.

<alert_action_template_name> is the name of a template. Use the following parameters to manage alert action templates:

Create an alert action template
: `–createAlertActionTemplate <alert_action_template_file>`

Return a specified alert action template
: `–getAlertActionTemplate <alert_action_template_name>`

Return all the alert action templates
: `–getAlertActionTemplate`

Delete a specified action template.
: `–deleteAlertActionTemplate <alert_action_template_name>`

Alert action templates cannot be updated. To update a template, you must delete and re-create the template.

Alert actions

<alert_action_file_name> is the name of the file that contains details of the alert action in JSON format.

<alert_action_name> is the name of the action alert that you want to create or manage. Use the following parameters to manage alert actions:

Create an alert action
: `–createAlertAction <alert_action_file_name>`

Return a specified alert action
: `–getAlertAction <alert_action_name>`

Return all the alert actions
: `–getAlertAction`

Update an alert action
: `–updateAlertAction <alert_action_file_name>`

You can only update the description and parameters.

Enable an alert action
: `–enableAlertAction <alert_action_name>`

Disable an alert action
: `–disableAlertAction <alert_action_name>`

Delete an alert action
: `–deleteAlertAction <alert_action_name>`

Composite condition templates

<composite_conditon_template_file> is the name of the file that contains details of the composite condition template in JSON format.

<composite_conditon_template_name> is the name of the composite condition template that you want to create or manage. Use the following parameters to manage composite condition templates:

Create a composition condition template
: `–createCompositeConditionTemplate <composite_conditon_template_file_name>`

Return a specified composite condition template
`–getCompositeConditionTemplate <composite_conditon_template_name>`

Return all the composite condition templates
`–getCompositeConditionTemplate`

Delete a composition condition template
`–deleteCompositeConditionTemplate <composite_conditon_template_name>`

Composite condition templates cannot be updated. To update a template, you must delete and re-create the template.

Composite condition

<*composite_conditon_file*> is the name of the file that contains details of the composite condition in JSON format.

<*composite_conditon_name*> is the name of the composite condition that you want to create or manage.

<*action_name*> is the name of the alert action.

Use the following parameters to manage composite conditions:

Create a composition condition
`–createCompositeCondition <composite_conditon_name>.json`

Return a specified composite condition
`–getCompositeCondition <composite_conditon_name>`

Return all the composite condition
`–getCompositeCondition`

Update a specified composite condition
`–updateCompositeCondition <composite_conditon_name>`

You can only update the description and parameters using the update command.

Enable a composite condition
`–enableCompositeCondition <composite_conditon_name>`

Disable a composite condition
`–disableCompositeCondition <composite_conditon_name>`

Delete a composition condition
`–deleteCompositeCondition <composite_conditon_name>`

Add an action to a composite condition
`-addActionToCompositeCondition <composite_condition_name> <action_name>`

Remove an action to a composite condition
`-removeActionFromCompositeCondition <composite_condition_name><action_name>`

Base conditions

Standard

To trigger alerts based on a single data source and log record, you define a base condition.

Creating a base condition

To create a base condition, you must specify the following details:

```
{ "name": "<base_condition_name>",
"description": "<base_condition_description>",
"baseConditionTemplateName": "<template_name>",
"datasourceName": "<datasource_name>",
"parameterValues": { "<parameter_1>" : "<value_1>", "<parameter_2>" : "<value_2>"},
"actions": ["<action_1>", "<action_2>", "<action_3>"]
}
```

- *<base_condition_name>* is the name.
- *<base_condition_description>* is the description.
- *<template_name>* is the template that you want to use as the basis for the condition instance.
- *<datasource_name>* is the name of the data source that the condition monitors.
- *<parameter_1>* and *<value_1>* are the parameters and values that you want to monitor.
- *<action_1>*, *<action_2>*, and *<action_3>* are the actions that are triggered when the condition is met.

For example:

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

Search query base condition template

You can use the built-in query base condition template to trigger an event based on a search query that is specified in the Apache Solr query syntax. Most of the common query syntaxes such as keyword, wildcard, regular expression, and Boolean queries are supported. You must specify index configuration field names in these queries.

The following base condition triggers three actions when a WebSphere Application Server log record with an error or warning severity level is ingested for a WAS datasource *datasource1*:

```
{
"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-base-condition", "email-user1", "index"]
}
```

Composite conditions

Standard

To trigger alerts that are based on time windows, the frequency of base conditions, or multiple data sources, define a composite condition.

Creating a composite condition

To create a composite condition, you must specify the following details:

```
{
"name": "<composite_condition_name>",
"description": "<composite_condition_description>",
"compositeConditionTemplateName": "<composite_condition_template_name>",
"parameterValues": { "<parameter_1>" : "<value_1>", "<parameter_2>" : "<value_2>"},
"inputConditions": ["<condition_1>", "<condition_2>", "<condition_3>"],
"actions": ["<action_1>", "<action_2>", "<action_3>"
]
}
```

- *<composite_condition_name>* is the name of the composite condition.
- *<composite_condition_description>* is the description of the composite condition.
- *<composite_condition_template_name>* is the name of the composite condition template that you want to base this instance on.
- *<inputConditions>* are the names of base conditions that are involved in the composite condition.
- *<parameter_1>* and *<value_1>* are the parameters and values that you want to monitor.
- *<action_1>*, *<action_2>*, and *<action_3>* are the actions that are triggered when the condition is met.

Single base condition counts template

You can use the `single-condition-count` template to create a composite condition that triggers an action if a certain base condition occurs a certain number of times over a specific time period.

The following example triggers two actions if the condition defined in the `datasource1- severity-base-condition` is met four times in 1 minute:

```
{
"name": "datasource1-severity-condition-count",
"description": "Condition on count for datasource1 severity values",
"compositeConditionTemplateName": "single-     condition-count",
"inputConditions": ["datasource1-severity-base-condition"],
"parameterValues": { "windowDuration" : "60s", "threshold" : "4"},
"actions": [ "index", "log-composite-condition"]
}
```

Single base condition deduplication template

You can use the `single-condition-dedup` template to create a composite condition that deduplicates multiple instances of a base condition in a certain time window. The following composite condition triggers alerts a maximum of once within any 60-second period for multiple occurrences of the conditions that are defined in the `datasource1-severity-base-condition`:

```
{
"name": "datasource1-severity-condition-count",
"description": "Condition on count for datasource1 severity values",
"compositeConditionTemplateName": "single-condition-dedup",
"inputConditions": ["datasource1-severity-base-condition"],
"parameterValues": { "windowDuration" : "60s"},
"actions": [ "index", "log-composite-condition"]
}
```

Multiple base condition window

You can use the `multi-condition-window` template to create a composite condition that triggers alerts when multiple base conditions occur within a defined time period. The following composite condition triggers alerts if both the `datasource1-severity-base-condition` and the `datasource2-severity-base-condition` base conditions occur within the same 60-second time frame:

```
{
"name": "severity-co-occurence-count",
"description": "Condition on co-occurrence of severity conditions for datasource1 and datasource2"
"compositeConditionTemplateName": "multi-condition-window",
"inputConditions": ["datasource1-severity-base-condition","datasource2-severity-base-condition"],
"parameterValues": { "windowDuration" : "60s"},
"actions": ["index", "log-composite-condition"]
}
```

Alert actions

Standard

Alert actions are triggered when a base or composite condition is satisfied.

You can use one of the four built-in templates to create alert actions or you can use the Alerting API to create custom alert action templates and actions.

Creating an alert action

To create an alert action, you need to specify the following details:

```
{
"name": "<alert_action_name>",
"description": "<alert_action_description>",
"alertActionTemplateName": "<template_name>",
"parameterValues": {"<parameter_1>" : "<value_1>",
"<parameter_2>" : "<value_2>"},
}
```

Index alert action template

You can use the index alert action template to index any triggered alerts. You cannot specify any parameters when you use this template.

This index alert action is recorded in a data source called **_alerts**. The **_alerts** contains the following indexed fields:

`conditionName`
The name of the condition that triggered the alert.

`conditionRecord`
The raw log record that caused the condition to trigger. This is only applicable to base conditions.

`conditionsDataSource`
The list of the data sources that triggered the condition.

`timeStamp`
The time stamp of the log record that triggered the condition.

You can use the Search API or the Search UI to search the **_alerts** data source. However, you cannot specify base conditions based on this data source.

For example, you can search the **_alerts** data source for log records that triggered a condition in the last hour or day. You can also use the **_alerts** data source create custom apps and dashboards.

Email alert action template

You can use the `email` template to send an email when a condition is met. The template uses the JavaMail API to facilitate the action. One session is created for each email that is sent.

The template contains the following parameters:

smtpMailServer
The host name of the SMTP mail server.

mailServerUser
The mail server user. This is optional.

mailServerPassword
The mail server user's password. This is optional.

secure This is set to false by default and is optional. If you set this to true, IBM Operations Analytics - Log Analysis uses STMPS to communicate with the mail server.

from The email address of the sender.

to The recipients of the mail.

cc The recipients who are copied on the mail. This is optional.

bcc The recipients who are copied on the mail but who are not visible to other recipients. This is optional.

subjectPrefix
The prefix that is displayed in the email text. This is optional.

header The text that is displayed in the first line of the email. This is optional.

footer The text that is displayed in the last line of the email. This is optional.

attachLogRecordsAnnotation
This is set to false by default and is optional. If set to true, all the annotations for the log records are attached to the email. You can only use this feature for base conditions.

Here is an example email alert action:

```
{ "name": "email-user1",
"description": "Email alert action",
"alertActionTemplateName": "email",
"parameterValues": { "smtpMailServer" : "123456.ibm.com",
"from": "user1@ibm.com", "to":["user2@email.com"],
"bcc": ["user3@email.com"],
"header": "Dear User,", "footer": "*** This is a system generated email.
Do not reply to this email****" }
}
```

Log alert action template

You can use the `log` template to log alerts in a file that you can use to debug alert implementations. The template has 1 parameter, `fileName`, that you use to specify the name of the file where the alert data is stored. The template buffers new alerts and adds the data to the file every 10 seconds.

Here is an example of a log alert action template. This template stores the alerts data to the `/tmp/base-conditions.log` file:

```
{
"name": "log-base-condition",
"description":"Log triggered base conditions",
"alertActionTemplateName": "log",
"parameterValues": {"fileName" : "/tmp/base-conditions.log"},
}
```

Script alert action template

You can use a `script` template to run an external script when a condition is met.

The template has three parameters:

`scriptPath`
The path to an executable that needs to be invoked.

`commandLineParameters`
Any parameters that must be passed on the command line while invoking the script. This is optional.

`working Directory`
The working directory to be used during script invocation. This is optional.

Here is an example of a `script alert action` template that invokes the script `<HOME>/unity/custom-script.pl` from the working directory `<HOME>/unity`.

```
{
"name": "<template_name>",
"description": "<template_description>",
"implLanguage": "JAVA",
"className": "alert.def.AlertTemplate1",
"implArtifact": base-64-encoded-jar-contents",
"parameters": { "parameter1": {"description": "parameter1",
"optional": false, "multivalued": false},
"parameter2": {"description": "parameter2",
"optional": true, "multivalued": true}
}
}
```

Alert action output structure

The alert action data is output as follows:

```
{
"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>
}
```

- *<condition_name>* is the name of the base or composite condition that triggered the action.
- *<base or composite>* indicates whether the condition is a base or composite condition.
- *<data_sources>* are the data sources that are specified in the conditions that triggered the alert.

- *<annotated-record-after-index-config-translation>* is the annotated log record that triggered the condition. This is only available for base conditions.
- *<long_timestamp_value>* is the long version of the timestamp of the log record.
- *<human_readable_ISO-8601_timestamp>* is a version of the timestamp that can be read by humans.
- *<condition_specific_alert_output>* is the output that is specified in the condition that triggered the output.

JSON templates for alerts

Standard

You can use the templates in the `<HOME>/IBM/LogAnalysis/utilities/alerts/` folder to help you to implement alerts with the command line utility.

Read about the templates, their uses and the required parameters.

Alert template

Standard

Use the `alertTemplate.json` file as a template for creating alert actions.

Template

```
{
  "name": "Template name",
  "description": "Template description",
  "implLanguage": "JAVA",
  "implArtifact": "/path/to/jar-file",
  "className": "class-name",
  "parameters": {
    "parameter-name1": {"description": "parameter1-description", "optional": true, "multivalued": fal
    "parameter-name2": {"description": "parameter2-description", "optional": true, "multivalued": fal
  }
}
```

Fields

Table 14. Fields in the alertTemplate.json file

Field	**Description**
`"name"`	Enter a name for the alert template.
`"description"`	Enter a description.
`"implLanguage"`	This indicates the language used. It is always `"JAVA"`.
`"implArtifact"`	Specify the path to the folder that contains the Java Archive (JAR) file that you want to use.
`"className"`	Enter a name for the class.
`"parameters"`	Enter 1 or more parameters that you want to monitor.

Parameters

Table 15. Parameters in the alertTemplate.json file

Parameter	Description
"parameter-name1"	Enter the name of the parameter that you want to monitor.
"description"	Enter a description for the parameter.
"optional"	True or false. If this is required to trigger the alert, specify true. If not, specify false.
"multivalued"	True or false. If the parameter contains multiple values, specify true. If not, specify false.
"type"	Specify the type of parameter. For example, if it is a string, specify STRING.

Anomaly Base Condition template

Standard

Use the anomalyBaseCondition.json file as a template for creating base conditions for alerts. You can also use it to create a base condition for anomaly alerts.

Template

```
{
  "name": "Base condition name",
  "description": "Base condition description",
  "baseConditionTemplateName": "anomaly",
  "datasourceName": "datasource-name",
  "parameterValues": { "fieldName" : "field-name", "resetPeriod": "1h"},
  "actions": ["index"]
}
```

Fields

Table 16.

Field	Description
"name"	Enter a name.
"description"	Enter a description.
"baseConditionTemplateName"	Enter the name of the base condition template that you want to use for this specific instance. This is configured to use the anomaly template by default.
"datasourceName"	Enter the name of the data source that is to be monitored.
"parameterValues"	Enter the parameter values.
"actions"	Enter the name of the action that you want to happen when the condition is met.

Parameters

Table 17.

Parameter	Description
`"fieldName"`	Enter the name of the field in the data source that you want to monitor.
`"resetPeriod"`	Enter how often the field needs to be checked for a changed value. Enter in seconds, minutes or hours. For example, to check for new values every 20 minutes, enter `20m`.

Email Alert Action template

Standard

Use the `emailAlertAction.json` file as a template for creating email alert actions.

Template

```
{
  "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,
  }
}
```

Fields

Table 18.

Field	Description
`"name"`	Enter a name for the action.
`"description"`	Enter a description.
`"alertActionTemplateName"`	Enter the alert action template used for this action.
`"parameterValues"`	Enter the information required for the email.

Parameters

Table 19.

Field	Description
`smtpMailServer`	Enter the host name of the SMTP mail server.
`mailServerUser`	Enter the mail server user. This is optional.

Table 19. (continued)

Field	Description
mailServerPassword	Enter the mail server user's password. This is optional.
secure	This is set to false by default and is optional. If you set this to true, Log Analysis uses STMPS to communicate with the mail server.
from	Enter the email address of the sender.
to	Enter the recipients of the mail.
cc	Enter the recipients who are copied on the mail. This is optional.
bcc	Enter the recipients who are copied on the mail but who are not visible to other recipients. This is optional.
subjectPrefix	Enter the prefix that is displayed in the email text. This is optional.
header	Enter the text that is displayed in the first line of the email. This is optional.
footer	Enter the text that is displayed in the last line of the email. This is optional.
attachLogRecordsAnnotation	This is set to false by default and is optional. If set to true, all the annotations for the log records are attached to the email. You can only use this feature for base conditions.

Log Alert Action template

Standard

Use the `logAlertAction.json` file as a template for creating log alert actions. These actions are saved to the `alert.log` file where you can use them to help you to debug your alerts.

Template

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

Fields

Table 20. Fields in the logAlertAction.json file

Field	Description
"name"	Enter a name.
"description"	Enter a description.
"alertActionTemplateName"	Enter the template that you used as the basis for this instance.
"parameterValues"	Enter the path to the directory where you want to save the `alert.log` file.

Multiple Condition Window template

Standard

Use the multiConditionWindow.json file as a template to help you to create a condition that uses multiple base conditions.

Template

```
{
  "name": "Condition name",
  "description": "Condition description",
  "compositeConditionTemplateName": "multi-condition-window",
  "inputConditions": ["input-condition-name1", "input-condition-name2"],
  "parameterValues": { "windowDuration" : "60s"},
  "actions": ["index"]
}
```

Fields

Table 21. Fields in the multiConditionWindow.json file

Field	Description
"name"	Enter a name.
"description"	Enter a description.
"compositeConditionTemplateName"	The name of the template that this instance is based on.
"inputConditions"	Enter the name of the base conditions that are used by this composite condition.
"parameterValues"	Enter the parameters that you want to monitor.
"actions"	Enter the name of the alert action that is triggered when the condition is met.

Query Base Condition template

Standard

Use the queryBaseCondition.json file as a template to help you create base conditions that are based on search queries.

Template

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

Fields

Table 22. Fields in the queryBaseCondition.json file

Field	Description
"name"	Enter a name.
"description"	Enter a description.

Table 22. Fields in the `queryBaseCondition.json` file (continued)

Field	Description
"baseConditionTemplateName"	Enter the template that this instance is based on.
"datasourceName"	Enter the name of the data source that you want to monitor.
"parameterValues"	Enter the search query that you want to use to find the field that you want to monitor.
"actions"	Enter the actions that are triggered when the condition is met.

Single Condition Count template

Standard

Use the `singleConditionCount.json` as a template to help you to create a condition that triggers an action when a certain base condition occurs a specified amount of time in a specified period.

Template

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

Fields

Table 23. Fields in the Single Condition Count template

Field	Header
"name"	Enter a name.
"description"	Enter a description.
"compositeConditionTemplateName"	Enter the name of the composite condition template that this instance is based on.
"inputConditions"	Enter the name of the data source that you want to monitor.
"parameterValues"	Enter the name of the field that you want to monitor, the value, and the amount of times the value needs to occur to trigger the action.
"actions"	Enter the action that you want to occur when the threshold is exceeded.

Single Condition Deduplication template

Standard

Use the `singleConditionDedup.json` template to help you to create a condition that triggers an action when a certain base condition occurs a specified amount of time in a specified period while eliminating duplicates.

Template

```
{
  "name": "Condition name",
  "description": "Condition description",
  "compositeConditionTemplateName": "single-condition-dedup",
  "inputConditions": ["input-condition-name"],
  "parameterValues": { "windowDuration" : "60s"},
  "actions": ["index"]
}
```

Fields

Table 24. Fields in the Single Condition Deduplication template

Field	Description
"name"	Enter a name.
"description"	Enter a description.
"compositeConditionTemplateName"	Enter the name of the composite condition template that this instance is based on.
"inputConditions"	Enter the name of the data source that you want to monitor.
"parameterValues"	Enter the name of the field that you want to monitor, the value, and the amount of times the value needs to occur to trigger the action.
"actions"	Enter the action that you want to occur when the threshold is exceeded.

Alert Action Script template

Standard

Use the `scriptAlertAction.json` file as a template to help you to create alert actions that run a script when the alert is triggered.

Template

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

Fields

Table 25. Fields in the alert action script template

Field	Description
"name"	Enter a name.
"description"	Enter a description.
"alertActionTemplateName"	Enter the name of the alert action template that this is based on.

Parameters

Table 26. Parameters in the alert action script template

Parameter	Description
"scriptPath"	Enter the directory where the script is stored.
"commandLineParameters"	Enter any command line parameters that you want to pass to the script when it is run.
"workingDirectory"	Enter the directory where the script is run.

Configuring alerts with the REST API

Standard

To help automate alert monitoring, you can use the REST API to create alerts, alert actions, and conditions.

For more information about the REST API for alerts, see "Alerting REST API" on page 391.

Configuring auditing

You can use the auditing features of IBM Operations Analytics - Log Analysis to track activities to provide fact analysis and actively monitor user activities.

The IBM Operations Analytics - Log Analysis auditing feature is installed and enabled by default.

The audit data is stored in the `audit.log` file in the <HOME>/IBM/LogAnalysis/logs directory, and in the Apache Solr index.

IBM Operations Analytics - Log Analysis

supports auditing of the following processes:
- Access (login/logout)
- Authorization (user/role/permission management)
- Ingestion
- Search
- Alerts

Audit parameters

The auditing feature is enabled by default. The administrator can modify the default parameters after installation if required.

The `unitysetup.properties` file is in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` folder. The administrator can edit values for the parameters in table 1, if required.

Table 27. Audit `unitysetup.properties`

Parameters	Value
`AUDIT_ACTIONS=`	Specifies where the audit data is stored. The default value is `LOG,INDEX`. These values are the only supported values and are enabled by default.
`AUDIT_INTERVAL=`	Defines how frequently the audit data is written. The default value is `120000` milliseconds.

The audit data is written in JSON format to the `<HOME>/IBM/LogAnalysis/logs/audit.log` file. The audit file is a rolling file, supporting up to 20, 50-MB files by default.

The default file properties are in the `log4j.properties` file in `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/` directory.

Note: If you change any of the audit parameters, you must save your changes and restart IBM Operations Analytics - Log Analysis.

Viewing the audit file

The audit file is a rolling file that can be accessed and searched at any time in the `audit.log` file or searched on the Solr console.

About this task

The audit feature of IBM Operations Analytics - Log Analysis is enabled by default, and records audit data at specified intervals. For information about the modifying the default parameters, see the *Audit parameters* topic in the *Configuring Auditing* section of the *Configuration* guide.

Procedure

To view the audit data, open the `audit.log` file in the `<HOME>/IBM/LogAnalysis/logs` directory.
The audit data is written in JSON format to the `<HOME>/IBM/LogAnalysis/logs/audit.log` file.

Configuring Expert Advice

You can use the Expert Advice app to search a knowledge source for terms that occur in log entries.

You can use the default version of the Expert Advice custom app. This version searches the IBM Support portal for the terms that are contained in selected columns. You can customize this version of the app to suit your needs. See “Customizing the default Expert Advice custom app” on page 97.

You can also implement your own version of the Expert Advice app. You can also customize this version so that it searches a knowledge base other than the IBM support portal.

For more information, see “Configuring a custom Expert Advice app” on page 98.

Customizing the default Expert Advice custom app

You can use the default implementation of the Expert Advice custom app to search the IBM Support portal for the terms that are contained in selected columns. You can also configure the custom app settings to suits your needs.

The custom app that is used to provide expert advice is in the `<HOME>/AppFramework/Apps/` directory where `<HOME>` is the directory in which IBM Operations Analytics - Log Analysis is installed. The custom app is named `IBMSupportPortal-ExpertAdvice.app`.

Displaying more results

By default, the 10 most relevant results are displayed. To change this value, open the custom app file and edit the number of displayed items. This setting is determined by the `value` parameter. To edit this parameter, open the custom app file and edit the `value` parameter:

```
{
"name": "__MAX_RESULTS",
"value" : "10",
"type" : "String"
},
```

Note: Increasing this value might affect the performance of the custom app as the additional data must be collected before it can be displayed.

Increasing the number of search terms

You can configure the number of unique terms that can be accepted for a search. By default, the number of unique terms is set to 7. To edit this parameter, open the custom app file and edit the `value` parameter:

```
{
"name": "__TERM_LIMIT",
"value" : "7",
"type" : "String"
},
```

Enhancing search strings

To ensure that a search return results, the custom app removes content from the search term that is unlikely to return results. For example, a log message that contains the search string `unable to access jarfile /myMachine/foo/bar/foobar.jar` is not likely to return a specific match as the server path is likely to be specific to a user. To ensure better search results, the custom app abbreviates this to `unable to access jarfile`.

The custom app uses a set of criteria to amend the search string. The following values that are removed:

- URL
- File name
- File path
- IP address
- Number
- Punctuation

These values are specified as regular expression patterns in JavaScript Object Notation (JSON) format in the GeneralizerPatterns_1.0.json file. This file is the dictionary of pattern definitions. To add more patterns or to edit the existing patterns, edit this file.

The set of patterns that are applied to the input data are specified in the custom app file. To apply the new pattern definitions that you created in the GeneralizerPatterns_1.0.json file, or to remove some of the patterns that are being applied, edit:

```
{
"name": "__DEFAULT_GENERALIZATION",
"value" : ["URL", "FILENAME","FILEPATH", "IP", "NUMBER", "PUNCTUATION" ],
"type" : "StringArray"
},
```

The order in which the patterns are specified in the __DEFAULT_GENERALIZATION parameter is the order in which they are applied to the data. Ensure that the order is correct. For example, IP must be matched before the NUMBER value. Otherwise, the IP pattern is not matched.

Debugging

The custom app generates a number of debug messages that provide information about the execution flow and custom app status. These messages are not displayed by default. To enable displaying these messages, set the following parameter to true:

```
{
"name": "__DEBUG",
"value" : "false",
"type" : "String"
},
```

The custom app supports two levels of debugging. Level 1 displays only the most important messages and Level 2 displays all messages. To set this level, edit the value parameter in this section of the file:

```
{
"name": "__LOGLEVEL",
"value" : "2",
"type" : "String"
}
```

Configuring a custom Expert Advice app

To implement a custom version of the Expert Advice app that uses a knowledge source other than the IBM support portal, you must create a new searcher.

Procedure

1. Create a search wrapper code for your customized knowledge source. The wrapper code must contain implementations of the Searcher, SearchPage, and Hit interfaces. For details about each interface, see “Searcher interface” on page 99, “SearchPage interface” on page 100, and “Hit interface” on page 101.
2. Save the search wrapper code as a Java Archive file.
3. Specify the path to this Java Archive file in the __SEARCH_CONNECTOR_JAR parameter in the Expert Advice app file. For example:

```
{
"name": "__SEARCH_CONNECTOR_JAR",
"value" : "ExampleConnector.jar",
"type" : "String"
},
```

4. The search wrapper code must implement a searcher interface. Specify the class that implements the interface in `__SEARCH_CONNECTOR_CLASS` parameter in the Expert Advice app file. For example:

   ```
   {
   "name": "__SEARCH_CONNECTOR_CLASS",
   "value" : "com.foo.bar.connector.example.ExampleSearcher",
   "type" : "String"
   },
   ```

   For more information about the Searcher interface, see “`Searcher` interface.”

5. If you want to pass arguments to the search wrapper code, specify the arguments in the `__SEARCH_CONNECTOR_ARGS` parameter in the Expert Advice app file. For example:

   ```
   {
   "name": "__SEARCH_CONNECTOR_ARGS",
   "value" : ["foo", "bar"],
   "type" : "String"
   },
   ```

6. If your search wrapper requires more Java Archive files, you must add the paths for these files to the Java class path in the `ExpertAdvice.sh` file.

`Searcher` interface

The `Searcher` interface is the main interface that the search wrapper must implement.

The basic function of the wrapper as implemented by the `Searcher` interface can be summarized as follows:

1. Accepts a query.
2. Run the query against the specified knowledge source.
3. Return the query results.

The interface is defined as follows:

```
package com.ibm.tivoli.unity.loganalytics.knowledgeSource;
public interface Searcher {
  public void init(String[] args) throws Exception;
  public SearchPage issue(String query) throws Exception;
  public SearchPage issue(QueryLevel<String> query) throws Exception;
}
```

Methods

The interface contains the following methods:

`init(String[] args)`

Expert advice dynamically loads the Java Archive file that is specified in the app file. It then creates an instance of the class that implements the `Searcher` interface that is also specified in the app file. You must ensure that an empty constructor exists for this class. After the Expert Advice app creates an instance of the class, it calls the `init()` method of that instance along with the arguments that are specified in the app file. You can use the `init()` method to specify any initialization that requires external arguments.

issue(String query)

Use this method to run a simple query against the knowledge source. Billing error is an example of a simple input query. The output object must be a class that implements the SearchPage interface. For more information about the SearchPage interface, see "SearchPage interface"

issue(QueryLevel<*String*> query)

Use this method to run a structured query against the knowledge source. The QueryLevel structured query object encapsulates the query levels or order. The output object must be a class that implements the SearchPage interface.

SearchPage interface

The SearchPage interface represents the set of ordered results that are returned by the query from the knowledge source.

This interface is defined as:

```
package com.ibm.tivoli.unity.loganalytics.knowledgeSource;
public interface SearchPage {
  public String getUserQuery();
  public String getProcessingTime();
  public long getTotalResults();
  public long getStartIndex();
  public long getNumResults();
  public List<Hit> getHits();
}
```

Methods

This interface has the following methods:

getProcessingTime()

The processing time as reported by the search system. This method is not parsed into any numeric format and is for display purposes only.

getTotalResults()

The total number of available results as reported by the search system. It is only for display purposes.

getStartIndex()

The start index value for the first result in the search page. For example, if the query returns 100 results and the wrapper reads 10 pages at a time, the method returns a value of 0 and the page contains results 0 to 9. The next search page, the method returns a value of 10 and the page contains results 10 to 19.

getNumResults()

The number of results that are available for the search page.

getHits()

The ranked list of results for the search page. Each result entry must be a class that implements the Hit interface. For more information about the Hit interface, see "Hit interface" on page 101.

Hit interface

The Hit interface represents a single result object.

This interface is defined as:

```
package com.foo.bar.example.loganalytics.knowledgeSource;

public interface Hit {
  public int getOrdinalNum();
  public String getTitle();
  public String getUrl();
  public String getSummary();
}
```

Methods

This interface contains the following methods:

getOrdinalNumber()
: The ordinal number of this result that is used on the parent search page.

getTitle()
: The title of the result page. The title is used as the anchor text when results are displayed on the Expert Advice search results page.

getUrl()
: The URL of the result page. The url is used to generate the links that are displayed on the Expert Advice search results page.

Additional classes

When it interacts with the Expert Advice code, the wrapper code can refer to two classes, QueryLevel and ItemSet.

QueryLevel

The app generates a series of queries that are based on the set of columns that the user specifies. The precision of the queries varies and the queries are ordered to reflect the precision of the results. The order is such that a query at a higher-level yields more accurate results that a lower-level query. The QueryLevel class captures the ordering of the queries. Some queries can yield results with the same level of precision without yielding the same results. The QueryLevel class can contain more than one query for this reason.

This class is defined as:

```
package foo.bar.example.loganalytics.artifacts;
public class QueryLevel<T> extends ArrayList<ItemSet<T>>
```

Each ItemSet object that is contained in a QueryLevel object represents a group of terms that must all occur on the same page. You must use an AND query for these terms.

If there is more than one ItemSet object in a QueryLevel object, you must use an OR query to separate the individual AND queries. For example:

```
(ItemSet_1_Term_1 AND ItemSet_1_Term_2) OR (ItemSet_2_Term_1 AND ItemSet_2_Term_2)
```

The syntax that specifies the AND and OR logic is specific to the knowledge source and is handled by the wrapper code.

ItemSet

The ItemSet class represents a single group of terms that must all occur in the same results page. You must use an AND query for this class.

This class is defined as:

```
package foo.bar.example.loganalytics.partialorder;
public class ItemSet<T> extends HashSet<T>
```

Configuring launch in context

You can launch IBM Operations Analytics - Log Analysis in context from within an approved IBM product with the Search **UI** or custom app.

Search UI launch-in-context

You can use the Search **UI** to launch IBM Operations Analytics - Log Analysis in context from within other products.

To start IBM Operations Analytics - Log Analysis in context using the Search **UI**, you must specify a URL in the following format:

```
https://<hostname>:<port>/Unity/SearchUI?queryString=<q>&timefilter=<t>
&dataSources=<ds>
```

where:

hostname
The host name that corresponds to the data source.

port The port that is used for communication with the IBM Operations Analytics - Log Analysis web application.

q The value of the search string with a valid velocity query syntax.

t A JSON format file filter to specify absolute or relative time.

For example, absolute time filters include "startTime": "24/206/2013 05:30:00" and "endTime": "25/06/2013 05:30:00". Relative time filters include "granularity": "Day" and "lastnum": "7".

ds A JSON file format to specify single or group data sources to be queried.

In this example, the user uses the Search **UI** to launch IBM Operations Analytics - Log Analysis.

```
https://0.000.00.00:1111/Unity/SearchUI?queryString=severity:==
"Critical"&timefilter={"type":"relative","lastnum":"7","granularity": "Day"}
&dataSources=[{ "type": "datasource", "name": <omnibusEvents>}]
```

Custom app launch-in-context

You can launch IBM Operations Analytics - Log Analysis custom apps in context from within other products.

To start IBM Operations Analytics - Log Analysis custom app in context, use the following URL:

```
https://<ip_address>:<port>:/Unity/CustomAppsUI?name=<name>&appParameters=<params>
```

where:

url The URL format that you specify must be in the format:

```
https://<ip_address>:<port>:/Unity/CustomAppsUI?name=<name>&appParameters=<params>
```

ip_address
The IP address of the server on which IBM Operations Analytics - Log Analysis is installed.

port The port that is used for communication with the IBM Operations Analytics - Log Analysis web application.

name Specify the name of the application file. This is the name of the `.app` file that displays in the Custom Apps pane in the Search workspace.

params
(Optional) Specify a custom app parameter JSON.

In this example, the user launches the *Day Trader App* custom app in context without a custom app parameter.

```
https://0.000.00.00:1111/Unity/CustomAppsUI?name=Day%20Trader&20App
```

Configuring the DSV toolkit

The DSV toolkit is used to create Insight Packs that allow you to load Delimiter Separated Value (DSV) data into IBM Operations Analytics - Log Analysis. The DSV toolkit contains python scripts that take input from a properties file that describes a DSV log type and produces as output an installable Insight Pack.

IBM Operations Analytics - Log Analysis provides a semi-structured data analytics solution. Use IBM Operations Analytics - Log Analysis to identify problems and propose solutions by searching and indexing large volumes of unstructured data from a variety of sources. IBM Operations Analytics - Log Analysis allows you to reduce problem diagnosis and resolution time and more effectively manage your infrastructure and applications.

Before you can perform a search on log or other data, you must first load the data into IBM Operations Analytics - Log Analysis. An Insight Pack is a set of artifacts packaged together to allow IBM Operations Analytics - Log Analysis to ingest the data that is loaded. An Insight Pack contains a complete set of artifacts required to process a data source. You can install, uninstall, or upgrade an Insight Pack as a stand-alone package.

Each Insight Pack defines:

- The type of log data that is to be consumed.
- How data is annotated. The data is annotated to highlight relevant information.
- How the annotated data is indexed. The indexing process allows you to manipulate search results for better problem determination and diagnostics.
- Optionally, how to render data in an app chart or visualization.

What's new

The DSV toolkit was updated to improve ingestion performance and update property files.

- This DSVToolkit provides improved ingestion performance, with the ability to ingest large DSV records. The adjustment to the Java stack size for files with long log records is no longer required.
- Users can continue to use Insight Packs that were generated by older versions of the DSVToolkit. The improved performance of the updated DSVToolkit does not apply to these Insight Packs.

- The aqlModuleName property was renamed as moduleName. Generated property files that contain the aqlModuleName property continue to function correctly but a warning message, indicating that the aqlModuleName property is deprecated, is displayed.
- Use the newly added quoteChar property to specify the quotation character that you want to use to enclose fields that contain delimiters and line-breaks.

  Note: If the quoteChar property is not specified, the double quotation mark (") is added by default during processing.

  Property files that were created before the quoteChar property was added work as before as the double quotation mark was implicit in previous DSVToolkit versions.
- The totalColumns property is no longer required.

Create an Insight Pack using the DSV toolkit

This topic outlines how to install the DSV toolkit and use it to create an Insight Pack. If you are using IBM Operations Analytics - Log Analysis 1.2 or later, the DSV Toolkit is installed by default, and therefore does not need to be installed separately.

Before you begin

Copy the DSVToolkit_v1.1.0.4.zip from <HOME>/SmartCloudAnalyticsLogAnalysisContent/tooling to the <HOME>/unity_content directory and extract the files. The files are extracted to the <HOME>/unity_content/DSVToolkit_v1.1.0.4 directory. Ensure that you, and any additional users that require the DSV toolkit, have write permissions to the DSVToolkit_v1.1.0.4 directory.

Procedure

1. Run the primeProps.py script to create a new properties file, or to update an existing properties file.
2. Edit the properties file to meet the requirements of your DSV log file format. For information about the requirements of each section of the properties file, see the *Specifying properties for a log file type* section of this document.
3. Run the devGen.py script to generate, and where required deploy, your Insight Pack.

Specifying properties for a log file type

This topic outlines the properties that describe the contents of your DSV file. Each type of DSV file that you wish to ingest requires a properties file.

The properties file must conform to a specific format. For example, section headers must be enclosed in brackets and each section is made up of items that must be in the format key: value or key=value.

Note: When specifying items in a key: value format, a space is required after the colon (:).

For more information, see http://docs.python.org/2/library/configparser.html

SCALA_server

Specify the details for your IBM Operations Analytics - Log Analysis server in the SCALA_server section.

The following parameter must be defined:

scalaHome
Specify the path to the home directory of the IBM Operations Analytics - Log Analysis server.

DSV_file

The DSV file section specifies parameters that apply to the entire log file and the entire Insight Pack.

DSV file parameters

delimiter
Specify the column separator that is used in the DSV log.

version
Specify the version number of the Insight Pack. The version must be a four digit number with a period separating each number. For example, 1.2.3.4.

moduleName
Specify the name of the Insight Pack and the underlying IOL-LA artifacts. The module name must:

- Start with a letter or an underscore (_). The letters can be upper or lowercase.
- Subsequent characters can be upper or lowercase letters, underscores, or digits (0-9).

quoteChar
Specify the quotation mark character that is used to enclose fields that contain delimiters and line-breaks.

Updates to the DSV file parameters

- The aqlModuleName property was renamed as moduleName. Generated property files that contain the aqlModuleName property continue to function correctly but a warning message, indicating that the aqlModuleName property is deprecated, is displayed.
- Use the newly added quoteChar property to specify the quotation character that you want to use to enclose fields that contain delimiters and line-breaks.
- The totalColumns property is no longer required.

field*_indexConfig

Define values for index configuration in the field*_indexConfig section.

You can create a section in your properties file for each column of your DSV log file type. For each column that you want to add to the properties file, replace the * in field*_indexConfig with the column number in your DSV log file. For example, field3_indexConfig corresponds to column 3 of the DSV log file. Add fields as required. For more information, see the *Index Configuration* topics in the *Administering IBM Operations Analytics - Log Analysis* section of the Information Center.

name Specify the name of the field in the index configuration. The field name is displayed in the Search workspace.

dataType
Specify the type of data contained in the log column. The valid values are TEXT, DATE, LONG, or DOUBLE.

dateFormat
This field is required if you have specified DATE as the dataType value. Specify the format of the timestamp used by the DSV log file. This format must conform with the Java 7 SimpleDateFormat class specification.

retrievable
(Optional) Specify the value for the retrievable field in the index configuration. The default value is TRUE.

retrieveByDefault
(Optional) Specify the value for the retrieveByDefault field in the index configuration. The default value is TRUE.

sortable
(Optional) Specify the value for the sortable field in the index configuration. The default value is FALSE. For the timestamp field, this value must be set to TRUE.

filterable
(Optional) Specify the value for the filterable field in the index configuration. The default value is FALSE. For the timestamp field, this value must be set to TRUE.

searchable
(Optional) Specify the value for the searchable field in the index configuration. The default value is TRUE.

path_* (Optional) Specify the additional paths that are added to the list of paths in the index configuration. Replace the asterisk (*) with an integer. Start at 1 and increment by 1 for each additional path. One path is generated dynamically and placed at the end of the list.

A path is a JSON path that points to the data that is displayed in the Search workspace. The path must have the form:

```
path_1: annotations.aqlModuleName_viewName.fieldName
```

where the dynamically generated paths are created using the following substitutions:

aqlModuleName
The aqlModuleName in the properties file.

viewName
The name item in an indexConfig section with the word Final appended to it.

fieldName
The name in the indexConfig section. For an example of fieldName, see the *Excluding and combining columns* topic.

combine
This field is required if a path is specified. Specify the method used to merge the contents of multiple paths. The valid values are FIRST and ALL. The default value is FIRST. When the combine value is set to FIRST, paths that you have defined are checked before the dynamically generated path.

Defining a timestamp section:

The properties file must contain a section that corresponds with the timestamp of the log record. The name item must be timestamp, the dataType must be DATE, a dateFormat must be specified, and sortable, filterable, and searchable must all be set to true.

For an example of a correctly completed timestamp section, see the *Example properties file* topic. For more information about supported date formats, see the *Supported formats* section.

field0_indexConfig

Define the index configuration for the whole log record.

The name, dataType, path, and combine values are shown in the *Example properties file* topic. These values must appear unchanged in every properties file. The field number in the section name must be 0.

Example properties file with edited index configuration fields

This is a sample properties file. All of the required values have been specified.

```
[SCALA_server]
scalaHome: $HOME/IBM/LogAnalysis

[DSV_file]
delimiter: ,
aqlModuleName: csv3Column
version: 1.0.0.0
totalColumns: 3

[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
retrievable: true
retrieveByDefault: true
sortable: true
filterable: true
searchable: true
dataType: DATE
dateFormat: yyyy-MM-dd'T'HH:mm:ss.SSSX

[field2_indexConfig]
name: severity
retrievable: true
retrieveByDefault: true
sortable: false
filterable: true
searchable: true
dataType: TEXT

[field3_indexConfig]
name: message
retrievable: true
retrieveByDefault: true
```

```
sortable: false
filterable: false
searchable: true
dataType: TEXT
```

This properties file creates an Index configuration that processes log records similar to this example:

```
2013-04-25T12:30:49.456-02:00, Warning,
Heap utilization patterns indicate that you may have a memory leak
```

Excluding and combining columns

This topic outlines how you can configure your properties file to exclude and combine columns from your DSV log file when it is displayed in IBM Operations Analytics - Log Analysis.

Excluding columns

If you do not want to display a column that is included in a DSV log file, do not specify that column when you add `indexConfig` sections to the properties file. For example, if you do not want to display columns 2, 3, and 4 of a 5 column log file, only specify the `field1_indexConfig` and `field5_indexConfig` property sections.

Combining columns

You can combine multiple columns from the DSV log file into one column in the Search workspace by specifying multiple paths in one `indexConfig` section. The section with multiple paths must be the one with the highest column number to ensure that the correct annotation is applied to the DSV log file. Fields that are part of the combined column, but are otherwise unimportant, can have all `true/false` index configuration fields set to `false` to ensure that data that is not required is not indexed.

The sample properties file combines the 2nd and 4th columns of the DSV log file into one column when it is displayed in the IBM Operations Analytics - Log Analysis Search workspace.

```
[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: shortMessage
combine: ALL
```

Generate a properties file

Use the primeProps.py to generate a template properties file. Default values are added where appropriate. Update the template properties file before running the dsvGen script.

Syntax

```
python primeProps.py pathToProps numNewSections [options]
```

Parameters

These parameters are defined for this script:

pathToProps
The path to the properties file that you want to create.

numNewSections
Specify the number of indexConfig sections that you want to add to the properties file. Each indexConfig section corresponds to a column in your DSV file.

Options

These additional options are defined for this script:

-o Overwrites the existing properties file. The default value for this property is false.

-h Displays the help screen for this script.

-f Add this option and specify the path to a DSV file containing a header. The header is parsed and the name item for each generated section uses the header name instead of a default name.

After running this command, open and review the properties file to ensure that the name of each section complies with the requirements of AQL and the DSV specifications. For example, the timestamp must be lower case and the name cannot contain spaces.

Example primeProps.py output

This example displays the output generated by the primeProps.py script. It shows a template with default values for the DSV_file, SCALA_server, and field0_indexConfig sections The command python primeProps.py dsvProperties.properties 1 results in this output:

```
[SCALA_server]
scalaHome: $HOME/IBM/LogAnalysis

[DSV_file]
delimiter: ,
aqlModuleName: dsv1Column
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: field1
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
```

Next steps

To ensure that the properties file contains sufficient detail to generate an Insight Pack, complete these steps:

- Verify the default username and password.
- Verify the default delimiter.
- Verify that the `indexConfig` section of the properties file contains a field named timestamp. You must edit the relevant field name.
- Edit a section so that the name is `timestamp`, the `dataType` is set to `DATE`, the `dateFormat` value is appropriate, and that the sortable and filterable values are set to `TRUE`.
- Verify the default `scalaHome` location.

Generate an Insight Pack

Use the `dsvGen.py` script to generate and, where required deploy, an Insight Pack. You can also use the `pkg_mgmt.sh` command to install the Insight Pack. After the Insight Pack has been generated you can use the Log Analysis Insight Pack Tooling to make any additional changes that you require.

Syntax

The syntax required to run the `dsvGen.py` script is:

```
python dsvGen.py pathToProps [options]
```

Parameters

These parameters are defined for this script:

pathToProps
The path to the properties file that you want to use to generate your Insight Pack.

Options

These additional options are defined for this script:

-o Overwrites the existing Insight Pack archive file. The default value for this property is `false`.

-h Displays the help screen for this script.

-d Deploy the Insight Pack using the `pkg_mgmt.sh` command. The `install` and `deploylfa` options are used. The default value for this property is `false`.

If you specify both the -d and -o options, any Insight Pack of the same name is removed, using the `pkg_mgmt.sh` uninstall option, before the new Insight Pack is installed.

-f
Applies the `-f` option to the `pkg_mgmt.sh` command. This eliminates all user prompts. The `-d` option must be used when using `-f`.

-l
Creates a Log Source for the generated Insight Pack. The Log Source hostname corresponds to the short hostname of the current server, and the Log Source log path points to a default file in the `<HOME>/logsources/` directory. The `-d`, `-u`, and `-p` options must be used when you use the `-l` option, even if the default `unityadmin` credentials exist.

-u
Specify a username to pass to `pkg_mgmt.sh`. The default `unityadmin` credentials will be used if nothing is supplied.

-p
Specify a password to pass to `pkg_mgmt.sh`. The default `unityadmin` credentials will be used if nothing is supplied.

Example

Executing this command results in the output described:

```
python dsvGen.py dsv5.properties -o -d
```

In this example:

- The `aqlModuleName` item is set to `dsv5Column`.
- In the `DSVToolkit_v1.1.0.4/build` directory, the `dsv5ColumnInsightPack_v1.0.0.0` directory is deleted.
- In the `DSVToolkit_v1.1.0.4/dist` directory, the archive `dsv5ColumnInsightPack_v1.0.0.0.zip` is deleted.
- The Insight Pack archive is created in the `DSVToolkit_v1.1.0.4/dist` directory and is named `dsv5ColumnInsightPack_v1.0.0.0.zip`.
- The Insight Pack archive is copied to the `<HOME>/unity_content/DSV` directory.
- The `pkg_mgmt.sh` command removes the old Insight Pack, if it exists, and re-installs it using the new archive.

Troubleshooting

There are several commonly encountered problems when using the DSV Toolkit. This is a description of the symptoms encountered and suggested solutions for resolving the problems.

The number of successfully ingested records is 0

Problem:
The number of successfully ingested records is 0, and there are no errors.

Resolution:
The number of columns in the DSV file does not match the number specified in the properties file. Verify that the properties file is correct and that none of the records in the DSV file have an abnormal number of fields.

The dsvGen.py script is displaying an error

Problem:
The dsvGen.py script is displaying an error like "The timestamp field must be sortable and filterable" or "A field of type DATE was specified, but no dateFormat item was provided".

Resolution:
The timestamp field you define has a specific set of requirements. See the "Defining a timestamp section" on page 107 for the detailed requirement.

Supported formats

This section outlines the formats that are supported for DSV log files.

Timestamp formats

The timestamp format from the DSV log file is based on the timestamp formats supported by the Java 7 `SimpleDateFormat` class. Any date format not supported by the `SimpleDateFormat` class cannot be processed from a DSV log file.

If a time-only timestamp is contained in the log, `SimpleDateFormat` assigns a default date of Jan 1, 1970 to the timestamp. Any double quotes that surround a timestamp are removed. Other characters must be included as literals in the dateFormat item. For example, a timestamp surrounded by brackets must be specified in the format:

```
dateFormat '['yyyy-MM-dd HH:mm:ss']'
```

For more information on valid timestamp formats for SimpleDateFormat, see http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html.

Delimiter formats

The follow delimiters are supported:

- comma (,)
- colon (:)
- semicolon (;)
- pipe (|)
- dash (-)
- slash (/)
- backslash (\)
- tab (\t)
- tilde (~)

Quotation characters

The follow quotation characters are supported:

- Double quotation mark (")
- Single quotation mark (')

Note: If the `quoteChar` property is not specified, the double quotation mark (") is added by default during processing.

Property files that were created before the `quoteChar` property was added work as before as the double quotation mark was implicit in previous DSVToolkit versions.

DSV formats

The Insight Packs generated by the DSV toolkit support DSV log file types that meet these requirements. In each case, a comma is used as a delimiter:

- Each log record is on a separate line. A line break is used to delimit the log records. CRLF denotes a line break. For example:
  ```
   aaa,bbb,ccc CRLF
   zzz,yyy,xxx CRLF
  ```
- The last record in the file might, or might not, end with a line break. For example:
  ```
  aaa,bbb,ccc CRLF
  zzz,yyy,xxx
  ```
- A header line might be present as first line of the file. This header line has same format as all standard record lines and contains names that correspond to the fields in the file. The header line also contains the same number of fields as the records in the rest of the file. For example:
  ```
  field_name,field_name,field_name CRLF
  aaa,bbb,ccc CRLF
  zzz,yyy,xxx CRLF
  ```
- Within the header and each record, there might be one or more fields that are separated by delimiters. Each line contains the same number of fields. Spaces are considered as part of a field and are ignored. The last field in the record must not be followed by a delimiter. For example:
  ```
    aaa,bbb,ccc
  ```
- Each field might or might not be enclosed in quotation characters. If fields are not enclosed in quotation characters, quotation characters might not appear inside the fields. The following examples show fields that are enclosed by single and double quotation characters:
  ```
      "aaa","bbb","ccc" CRLF
      'zzz','yyy','xxx'
  ```
- Fields containing quotation characters, delimiters, and line breaks must be enclosed in quotation characters. The following examples show fields that are enclosed by single and double quotation characters:
  ```
  "aaa","b,bb","ccc" CRLF
   'zzz','y,yy','xxx'
  ```
- If quotation characters are used to enclose fields, a quotation character that appears inside a field must be escaped by preceding it with another quotation character. The following examples show single and double quotation characters:
  ```
      "aaa","b""bb","ccc"
     'zzz','y''yy','xxx'
  ```

Configuring Insight Packs

Before you can use any of the Insight Packs that are available out of the box with IBM Operations Analytics - Log Analysis , you must configure the Insight Packs.

The Insight Pack defines:

- The type of data that is to be consumed.
- How data is annotated. The data is annotated to highlight relevant information.
- How the annotated data is indexed. The indexing process allows you to manipulate search results for better problem determination and diagnostics.
- How to render the data in a chart.

The following Insight Packs are now installed with the product:

WASInsightPack

The WebSphere Application Server Insight Pack includes support for ingesting and performing metadata searches against WebSphere Application Server V7 and V8 log files. Updates to WAS index configuration will improve indexing performance. The field `logsourceHostname` has been changed to `datasourceHostname`.

WASAppInsightPack

The Websphere Application Server (WAS) Applications Insight Pack provides troubleshooting dashboards for WebSphere Application Server Logs. A new authentication mechanism eliminates the need to specify userid and password in the application script. The field `logsourceHostname` has been changed to `datasourceHostname`.

DB2InsightPack

The DB2 Insight Pack includes support for ingesting and performing metadata searches against DB2 version 9.7 and 10.1 `db2diag.log` files. The field `logsourceHostname` has been changed to `datasourceHostname`.

DB2AppInsightPack

The DB2 Applications Insight Pack provides troubleshooting dashboards for DB2 Logs. A new authentication mechanism eliminates the need to specify userid and password in the application script. The field `logsourceHostname` has been changed to `datasourceHostname`.

Syslog Insight Pack

The Syslog Insight Pack includes support for ingesting and performing metadata searches against syslog data logging. The field `logsourceHostname` has been changed to `datasourceHostname`.

WebAccessLogInsightPack

The Web Access Logs Insight Pack provides the capability to ingest and perform metadata searches against Web Access Logs such as Apache IHS, JBoss, Apache Tomcat. The pack now includes a Web Health Check Dashboard example that provides summaries of key metrics.

WindowsOSEventsInsightPack

You can use the Windows OS Event Insight pack and the IBM Tivoli Monitoring Log File Agent to load and search Windows OS events. New support for data collection using logstash provides an alternative to the IBM Tivoli Monitoring Log File Agent.

JavacoreInsightPack

The Java Core Insight Pack provides the capability to ingest and search metadata that originates in Java Core files in IBM Operations Analytics - Log Analysis. The field `logsourceHostname` has been changed to `datasourceHostname`.

GAInsightPack

The Generic Annotation Insight Pack is not specific to any particular log data type. It can be used to analyze log files for which a log-specific Insight Pack is not available

DB2 Insight Pack

A DB2® Insight Pack is provided with IBM Operations Analytics - Log Analysis.

The Insight Pack includes support for ingesting and performing metadata searches against the DB2 version 9.7 and 10.1 `db2diag.log` files.

This document describes the version of the DB2 Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the DB2 Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack and updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html.

Best practices for db2diag files

DB2 creates multiple `db2diag.log` files based on the database configuration. Follow these best practices to handle multiple `db2diag logs` for the rotating and multiple database partition scenarios.

Scenario 1: Database spread across partitions and members

If the database is spread across multiple partitions and members, then a `db2diag` file is created in multiple directories according to the `DIAGPATH` value. It can be difficult to interpret the `DIAGPATH` and `db2nodes.cfg` to find all the log files for each member and host. The best practice recommendation is to use the **`db2diag`** tool, which consolidates all the log records in to a single file.

For databases spread among multiple partitions and members, consolidate the records from all `db2diag log` files on all your database partitions, run the following command:

```
db2diag -global -output filename.log
```

Where *filename*.log is the name of the consolidated log file, such as `db2diag_myglobalrecords.log`.

Scenario 2: Rotating log files

If DB2 is configured for rotating log files, the files are named dynamically as db2diag.<*number*>.log. The best practice is to change the default log agent configuration files to recognize the rotating logs.

If you choose to consolidate the rotating logs with the `db2diag` tool, follow the *DB2 data loading best practice* about loading the consolidated file.

Using the Data Collector client to load the consolidated rotating logs can improve data ingestion performance. However, you must filter the logs based on timestamps to avoid ingesting duplicate logs each time you consolidate the logs.

Optional: Filtering `db2diag log` file records can reduce the time required to locate the records needed when troubleshooting problems. If the `db2diag` tool is used to filter records, only those records included in the filter result will be annotated by the IBM Operations Analytics tools - all other records are excluded. The **`db2diag`** tool includes other options for filtering and formatting the log records.

If you choose to consolidate the rotating logs with the `db2diag` tool, use the Data Collector as described in the *DB2 data loading best practice* about loading the consolidated file. More information about the **`db2diag`** utility can be found at the DB2Information Center here:

For version 10.1:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

For version 9.7:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

Note: To find the locations of the individual logs, use the DB2 command:

```
GET DATABASE MANAGER CONFIGURATION
```

On **UNIX** systems, The `DIAGPATH` parameter shows the diagnostic data directory path that contains the log files. The default directory is:

`INSTHOME``/sqllib/db2dump`,

where *`INSTHOME`* is the home directory of the DB2 instance.

On **Windows** systems, the default path depends on the operating system. To find the default path on Windows, use the command:

```
DB2SET DB2INSTPROF
```

Configuration artifacts

The Insight Pack supports the DB2 timestamp format `YYYY-MM-DD-hh.mm.ss Z`. You can customize artifacts in the index configuration file.

Note: Data sources are not predefined. A user with administrator privileges must define at least one DB2 data source before the application can be used.

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 28. Insight Pack configuration artifacts

Artifact	**Name for the `db2diag.log`**
Splitter rule set	`DB2Diag-Split`
Annotator rule set	`DB2Diag-Annotate`
Source type	`DB2Diag`
Collection	`DB2Diag-Collection1`

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the `<HOME>/IBM-LFA-6.30/config/lo` directory:

- `DB2InsightPack-lfadb2.conf`: Configuration file for the DB2 log file agent.
- `DB2InsightPack-lfadb2.fmt`: Matches records for the db2diag log files.

Splitting and annotation AQL modules

Splitting and annotation are handled by the following Annotation Query Language (AQL) modules.

Table 29. Insight Pack AQL modules

AQL Module	Description
common	Common code module that is used across multiple log files (for example, to recognize timestamp formats).
commonDB2	Common annotations module that is used across splitter and annotator modules.
annotatorDB2Diag	Annotator module for db2diag log files.
splitterDB2Diag	Splitter module for db2diag log files.

Log file formats

The basic formats of the DB2 log files are described here as a reference for users.

The basic format of db2diag.log file is:

```
timestamp recordId LEVEL: level(source)
PID : pid TID : tid PROC : procName
INSTANCE: instance NODE : node DB : database
APPHDL : appHandle APPID: appID
AUTHID : authID
EDUID : eduID EDUNAME: engine dispatchable unit name
FUNCTION: prodName, compName, funcName, probe: probeNum
MESSAGE : messageID msgText
CALLED : prodName, compName, funcName OSERR: errorName (errno)
RETCODE : type=retCode errorDesc
ARG #N : typeTitle, typeName, size bytes
... argument ...
DATA #N : typeTitle, typeName, size bytes
... data ...
```

Only these fields are present in all log records.

```
timestamp
timezone
recordID
level
pid
tid
FUNCTION
```

For more information about DB2 log file formats, see the following topic on the DB2 information centers:

The format for DB2 10.1 log files is documented in:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

The format for DB2 v9.7 log files is documented in:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

Log file splitters

The splitters provided with the Insight Pack are described here as a reference for users.

DB2diaglog splitter

The db2diaglog splitter uses timestamps to define the beginning and end of each log record.

Log file annotations

The annotations that are defined by the log file index configurations are described here.

The following table lists the index configuration files that are included in the Insight Pack.

Table 30. Index configuration files

Log file	Index configuration file
db2diag.log	Included in the sourcetypes.json file

The following sections describe the fields that are defined in the index configuration file. These fields, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records. Fields are extracted from the fields of a log record or collected from metadata around the log file. Each table gives the names of the fields (these names correspond to fields in the IBM Operations Analytics - Log Analysis Search workspace), descriptions of how the related annotations are made, and the index configuration attributes assigned to the fields.

Log record annotations

The following table lists the index configuration fields that relate to log record annotations. Each field corresponds to part of a db2diag log record. The fields are listed in the order in which they appear in a log record.

Table 31. Log record index configuration fields

Field	Description	Attributes

timestamp	The timestamp of the log record, which is located at the beginning of a line.	dataType = DATE
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

recordID	The record identifier of the log record that follows the timestamp. The recordID of the log files specifies the file offset at which the current message is being logged (for example, 27204 and the message length (for example, 655) for the platform where the log was created.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

diagnosticLevel	A diagnostic level of the log record that follows the label LEVEL:. It is the diagnostic level that is associated with an error message. For example, Info, Warning, Error, Severe, or Event." Not all log records have a diagnostic level.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true

Table 31. Log record index configuration fields (continued)

Field	Description	Attributes

processID	The process identifier in the log record that follows the PID: label. For example, 1988.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

threadID	The thread identifier (TID) for the process in the log record that follows the TID: label. For example, 1988	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

processName	The name of the process in the log record that follows the PROC:label. For example, db2iCacheFunction.exe	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

instance	The DB2 instance that generated the message in the log record that follows the INSTANCE:label. For example, DB2.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

node	The node identifier that generated the message for a multi-partition system. Otherwise it is 000. It follows the NODE:label. For example, 001.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true

databaseName	If present, the database name in the log record, that follows the DB:label. For example, DB2.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

applicationHandle	If present, the application handle in the log record, that follows the APPHDL: label. For example, 0-2772.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

applicationID	If present, the application identifier in the log record, that follows the APPID: label. For example, *LOCAL.DB2.130226175838.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

authorizationID	If present the authorization user identifier in the log record, that follows the AUTHID: label. For example, adminuser1.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

Table 31. Log record index configuration fields (continued)

Field	Description	Attributes

eduID	The engine dispatchable unit identifier in the log record that follows the EDUID: label. For example, 2004.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

eduName	The name of the engine dispatchable unit in the log record that follows the EDUNAME: label. For example, db2agent (instance).	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

functionProductName	The product name that wrote the log record. It follows the FUNCTION: label, which also includes the component name, function name, and function probe point. For example, DB2.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

functionComponentName	The name of the component that wrote the log record. It follows the product name in the FUNCTION: label, which also includes the product name, function name, and function probe point. For example, UDB	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

functionFunctionName	The name of the function that wrote the log record. It follows the component name in the FUNCTION: label, which also includes the product name, component name, and function probe point. For example, Self tuning memory manager	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

functionInfo	The information that is returned by the function in the log record. It follows the FUNCTION: label. It includes all information after the FUNCTION entry. For example, DATA #1 : unsigned integer, 8 bytes	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

functionProbePoint	The probe point within the function that wrote the log record. It follows the probe: label in the FUNCTION: label, which also includes the product name, component name, and function information. For example, probe:1008	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

message	The message that follows the MESSAGE: label. It is optional data that a function can provide in the log record. For example, New STMM log file (C:\ProgramData\IBM\DB2\DB2COPY1\DB2\stmmlog\stmm.0.log) created automatically.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

sqlcode	The SQL code is optional data that is provided by the function in the log record. It is preceded by the text SQLCODE or sqlcode.	dataType = LONG
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true

Table 31. Log record index configuration fields (continued)

Field	Description	Attributes

msgClassifier	The message ID if it exists in a message (which follows MESSAGE: label). It starts with 3 or 4 letters, followed by 4 numbers, followed by I, E, or W such as ADM0506I.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true

DB2Hostname	The hostname following the HOSTNAME: label where the DB2 log record was generated. If there is only one DB2 server, there is no hostname in the log record. For example, mydb2host.tiv.pok.ibm.com	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true

start	This is the message following the label START provided by the function. It is an indication of the start of an event. For example, Starting FCM Session Manager.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

stop	This is the message follows the label STOP provided by the function. It is an indication of the end of an event. For example, DATABASE: DTW : DEACTIVATED: NO.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

Metadata annotations

The following table lists the index configuration fields that relate to metadata annotations.

Table 32. Metadata index configuration fields

Field	Description	Annotation attributes
application	The application name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
hostname	The host name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
logRecord	The entire log record output by the splitter.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
datasourceHostname	The host name specified in the data source.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true

Table 32. Metadata index configuration fields (continued)

Field	Description	Annotation attributes
middleware	The middleware name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
service	The service name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Searching DB2 log files for a return code example

You can search the log files for keywords. Search results are displayed in a timeline and a table format. You can use the search function to find fields that are not indexed.

Before you begin

To search a log file, you must first define a data source as the db2diag.log file.

This example shows how to search for RETCODE. For more information about other keywords you can search for in the FUNCTION information, see the DB2information center for your version here:

For version 10.1:http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

For version 9.7:http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

Procedure

1. From the Search workspace, click **Add Search** to create a tab that contains your search criteria.
2. Optional: To limit the extent of the search to specific data sources and any descendant data sources, select a leaf node from the data source tree.
3. In the **Time Filter** pane, click the **Time Filter** list and select the time period for which you want to search. Select **Custom** to specify a start time and date, and an end time and date for your search.
4. In the **Search** field, type the return code string that you want to search for in the log files.

   For example, to search for return codes related to missing files include the string FILE_DOESNT_EXIST. The full return code for this example is:

   ```
   RETCODE : ECF=0x9000001A=-1879048166=ECF_FILE_DOESNT_EXIST
   ```

   To search for strings that related to missing files, type FILE_DOESNT_EXIST in the **Search** field.
5. Click **Search**.

   For more information about searching logs, see the section *Searching log files* in the IBM Operations Analytics - Log Analysis Knowledge Center

Results

A graph that shows the distribution of matching events in the log is displayed.

DB2 data loading best practice

Best practice recommendation for DB2 Insight Pack data loading.

There are different data loading recommendations depending on whether you used the rotating or consolidated DB2 `db2diag.log` files and how many servers you need to monitor. To set up the data loading, you need to consider:

DB2 configuration

Is DB2 configured to produce a single log file for a single server, rotating log files for a single server, or multiple logs for multiple servers?

You can use the use the **`db2diag`** tool that is included with DB2 to merge and consolidate the log files. More information about the **`db2diag`** utility can be found in the DB2 documentation at:

For version 10.1:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

For version 9.7:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

Data loading

Determine if you want to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote DB2 server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data. In some scenarios, you must use the Data Collector client as described in scenario 3.

Logfile agent configuration

Use the logfile agent configuration files to specify the log files you want to monitor. The `DB2InsightPack-lfadb2.conf` and `DB2InsightPack-lfadb2.fmt` files are located in the directory:

```
<HOME>/IBM-LFA-6.30/config/lo
```

The log file scenarios here describe the specific settings for these files.

Scenario 1 - Individual log file on one DB2 Server

For a single log file on one DB2 server follow these best practices.

DB2 Configuration

DB2 is configured for a single log file (non-rotating), `db2diag.log` on one server

Data Loading Method

The recommended method for loading data is to use the LFA installed on the remote DB2 server to push data or to use the LFA installed on the local IBM Operations Analytics server to pull data.

Logfile Agent Configuration - `DB2InsightPack-lfadb2.conf` file

In the `DB2InsightPack-lfadb2.conf` file, specify the following parameters to monitor the log files.

```
LogSources=<db2 log directory to monitor>/db2diag.log
#FileComparisonMode
```

The FileComparisonMode parameter should be commented out since it only applies when using wildcards in a LogSources parameter

Logfile Agent Configuration - DB2InsightPack-lfadb2.fmt file

Use the default DB2InsightPack-lfadb2.fmt file.

```
// Matches records for any Log file:
//

REGEX AllRecords
(.*)
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

Scenario 2 - log file rotation on one DB2 server

For rotated log files on a single DB2 server follow these best practices.

DB2 configuration

DB2 is configured for rotating log files using the DIAGSIZE configuration option. The db2diag.log files are named dynamically as db2diag.*<n>*.log.

Data Loading Method

The recommended method for loading data is to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote DB2 server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data.

Logfile Agent Configuration - DB2InsightPack-lfadb2.conf file

In the DB2InsightPack-lfadb2.conf file, specify the following parameters to monitor the rotating log files:

```
LogSources=<db2 log directory to monitor>/db2diag.*.log
FileComparisonMode=CompareByAllMatches
```

Logfile Agent Configuration - DB2InsightPack-lfadb2.fmt file

Use the following DB2InsightPack-lfadb2.fmt file to specify a fixed log file name. Otherwise you must define multiple logsources in the IBM Operations Analytics - Log Analysis Administrative Settings page because the rotating log file name changes. The fmt file allows a fixed file name in the logpath.

```
// Matches records for any Log file:
//

REGEX AllRecords
(.*)
hostname LABEL
-file db2diag.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

Scenario 3 - Consolidating log files from multiple DB2 servers

If you consolidate log files from multiple DB2 servers follow these best practices.

DB2 Configuration

If the database is spread across multiple partitions and members, then a db2diag.log file is created in multiple directories according to the DIAGPATH value. It can be difficult to interpret the DIAGPATH and db2nodes.cfg to find all the log files for each member and host. The best practice recommendation is to use the db2diag tool, which will bring the information from all the members together to create a consolidated db2diag.log. The db2diag utility allows you to filter based on timestamp and this should be done to include only new log entries in the consolidated logs. Information on this filter can be found here:

For version 9.7:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0011728.html

For version 10.1

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0011728.html

Data Loading Method

The recommended method for loading data is to use the Data Collector client. Remove the previous consolidated db2diag.log file before creating or copying a new version into the directory from which you load data.

Generic Annotation Insight Pack

A Generic Annotation Insight Pack is installed when you install IBM Operations Analytics - Log Analysis. This Insight Pack is not specific to any particular log data type. It can be used to analyze log files for which a log-specificInsight Pack is not available.

The Insight Pack facilitates data ingestion and metadata searches of logs files where a date and time stamp can be identified within the log records.

This document describes the version of the Generic Annotation Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Generic Annotation Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html

Generic Annotation installation

Instructions on how to install the Generic Annotation Insight Pack.

Procedure

1. Upload the Generic Annotation Insight Pack archive file, GenericAnnotationInsightPack_<version>.zip, to the system where IBM Operations Analytics - Log Analysis is installed.

   Where <version> is the version of the Generic Annotation Insight Pack.
2. Install the Generic Annotation Insight Pack with the pkg_mgmt.sh command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install <path>/
   GenericAnnotationInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Generic Annotation Insight Pack.
3. Deploy the log file agent with the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa <path>/
   GenericAnnotationInsightPack_<version>.zip
   ```

Related tasks:

Importing an Insight Pack
Import an existing Insight Pack into the Eclipse workspace.

Upgrading an Insight Pack

You can upgrade an Insight Pack that you have previously installed. This topic outlines how to upgrade an existing Insight Pack.

About this task

If the Insight Pack that you want to upgrade is not installed, you can choose to complete a full installation of the Insight Pack. In addition to upgrading existing artifacts and installing any artifacts added to the Insight Pack, this command removes unused artifacts that have been excluded from the upgraded Insight Pack.

Upgrade an Insight Pack by completing these steps:

Procedure

1. Download the Insight Pack archive and copy it to the `<HOME>/IBM/LogAnalysis/unity_content` directory on your IBM Operations Analytics - Log Analysis system.
2. Execute the command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -upgrade insight_pack.zip
   -U username -P password -f
   ```

   where `insight_pack` is the path to the Insight Pack that you want to upgrade. These additional parameters are also defined:

   -U (Optional) The username for a user with administrative access rights. This parameter is not necessary if you have not changed the default `unityadmin` password.

   -P (Optional) The password for the username that you have specified. This parameter is not necessary if you have not changed the default `unityadmin` password.

   -f (Optional) This parameter can also be used to install the Insight Pack, if it is not already installed.
3. (Optional) If the Insight Pack is not installed and you have not specified the `-f` parameter, a message is displayed indicating that the Insight Pack is not installed. If you want to proceed, enter `Y`.

Configuration artifacts

You must create at least one data source.

Data sources are not defined by default. Create at least one data source before the logs can be ingested.

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 33. Configuration Artifacts

Insight Pack	Configuration artifact
Splitter Rule Set	`dateTime-Split`
Splitter Rule Set	`timeOnly-Split`
Annotator Rule Set	`Generic-Annotate`
Source Type	`Generic`

Table 33. Configuration Artifacts (continued)

Insight Pack	Configuration artifact
Collection	Generic-Coll

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file. The Insight Pack supports a variety of timestamp formats. See the section “Timestamp formats” on page 131 for a complete list of the available formats.

Table 34. Configuration Artifacts. Table 2 lists the configuration artifacts that are provided with the Insight Pack.

Splitter rule set	Annotator rule set	Source type	Collection
Generic-dateTime-Split	Generic-Annotate	Generic	Generic-Collection1
Generic-timeOnly-Split			
NormalizedMonthFirst-Split		Generic-NormalizedMonthFirst	Normalized-MonthFirst
NormalizedDayFirst-Split		Generic-NormalizedDayFirst	Normalized-DayFirst
NormalizedYearFirst-Split		Generic-NormalizedYearFirst	Normalized-YearFirst

Log file formats

The Generic Annotation Insight Pack annotates all log files irrespective of their format.

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the `Log_Analytics_install_dir/IBM-LFA-6.30/config/lo` directory:

Log file splitter

The Generic Annotation Insight Pack contains Rule Sets that can be used to split incoming log files.

These are:

- `Generic-dateTime-Split` (default)
- `Generic-timeOnly-Split`

Each Rule Set splits a log based on each line having either a time stamp or a date and time stamp.

The `Generic-dateTime-split` splitter splits log records using the date and time stamp of the log file. If the log file does not have year format that the splitter can interpret in the log records, the splitter adds a year value based on the IBM Operations Analytics - Log Analysis server system time. The Index Configuration must be updated to reflect this action.

The `Generic-timeOnly-split` splitter splits log records using only the time stamp in the log record. Where the log file does not have a date in the log records that can be interpreted by splitter, the current date value set for the IBM Operations

Analytics - Log Analysis serveris used. The format `MM/dd/yyyy` is inserted before the format of the time. The Index Configuration must be updated to reflect this action.

The splitters provided with the Insight Pack are described here as a reference for users.

DateTime splitter
The dateTime splitter recognizes all supported timestamp formats. The timestamp must have a date and a time. If the year is missing from the date, the current year will be appended to the front of the timestamp. You must modify the index configuration with the proper timestamp format for the splitter to function properly.

TimeOnly splitter
The timeOnly splitter recognizes all supported time formats. The timestamp must have a time and must not have a date. The splitter will append the current date to the front of the timestamp in the format MM/dd/yyyy. You must modify the index configuration with the proper timestamp format for the splitter to function properly.

NormalizedMonthFirst splitter
The splitter assumes a purely numeric date (for example, 07/08/09) is in the format MM/dd/yy. The timestamp must have a time, and may have an optional date. The date may have an optional year. If the date or year is missing, the current date or year is substituted. The NormalizedMonthFirst splitter outputs the timestamp in a normalized format. As a result, the index configuration does not need to be modified with the timestamp format.

NormalizedDayFirst splitter
The splitter assumes a purely numeric date (for example, 07/08/09) is in the format dd/MM/yy. The timestamp must have a time, and may have an optional date. The date may have an optional year. If the date or year is missing, the current date or year is substituted. The NormalizedDayFirst splitter outputs the timestamp in a normalized format. As a result, the index configuration does not need to be modified with the timestamp format.

NormalizedYearFirst splitter
The splitter assumes a purely numeric date (for example, 07/08/09) is in the format yy/MM/dd. The timestamp must have a time, and may have an optional date. The date may have an optional year. If the date or year is missing, the current date or year is substituted. The NormalizedYearFirst splitter splitter outputs the timestamp in a normalized format. As a result, the index configuration does not need to be modified with the timestamp format.

Log file annotations

The Generic annotator allows you to search and analyze log files for which a specific annotator is not available. There are two types of annotations created by the Generic annotator. Those are Concepts and Key-Value pairs. This section outlines the purpose, scope, and use of the IBM Operations Analytics - Log Analysis Generic annotator.

Included concept tokens

A concept is a piece of text that represents a real world entity such as an IP address or a hostname. These concepts are useful for searches as they provide information that can assist you in diagnosing issues. The Generic annotator includes support for these annotation tokens:

Hostname
Names given to devices that connect to a network and that are referenced in the log record.

IP Address
Numeric labels given to devices that are connected to a network and that are referenced in the log record

Severity Level
The indicator of the severity of an event in a log record. The Generic annotator provides annotation for these severity levels:

- SUCCESS
- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF
- CRITICAL
- CRITICAL_ERROR
- SEVERE
- IGNORE
- WARNING
- CONFIG
- FINE
- FINER
- FINEST
- ALL

URL Web addresses listed in the log record.

Identifier
Patterns intended to capture names of constants that might repeat within the log record and that signify the occurrence of some event. For example, `ECH_PING_FAIL_BCKP`. The Generic annotator assumes that an identifier is a sequence of alphanumeric characters in capitals that may be separated by underscores.

Excluded concept tokens

The Generic annotator assumes that these tokens are noise and they are ignored:

Date and time
For the purposes of analysis date and time are not useful.

Number
The Generic annotator ignores both whole and decimal numbers.

Hexadecimal numbers
The Generic annotator ignores hexadecimal numbers such as 7AB87F.

Stop words
A list of stop words have been defined for the Generic annotator. This is to allow the Generic annotator to ignore common words that might appear frequently, but offer no value in an analysis of the log records.

Key-value pairs

A Key-value annotation extracts data from a log record if it is in the format `<key> = <value>`. For example, `ERROR-CODE = 4499`. These Key-value pairs can be used to list the values for each key. These limitations apply to Key-value pair annotations:

Colon separator
Key-value pairs that are separated by a colon are excluded. For example, `Label: ECH_PING_FAIL_BCKP`.

Hyphen prefix
Key-value pairs where the value begin with a hyphen are excluded. For example, `ERRORCODE = -4499`.

Numbers with commas
Key-value pairs where the value includes a comma are excluded. For example, `ERRORCODE = 4,499`.

Forward and backward slash characters
Key-value pairs where the value contains a forward or backward slash are excluded. For example, `path = /opt/IBM/`.

Quotes
Key-value pairs where the value is contained within quotation marks. For example, `language = "English"`.

Delimiter characters
Some limitations exist where the value in a Key-value pair contains a delimiter. However, these depend on the whether the value contains a token that can be annotated based on the list of included tokens. For example, `Time = Thu Nov 22 06:28:48 EST 2012` is delimited by a space after `Thu`and therefore the Key-value pair is assumed to be `Key = Time`, `Value = Thu`. However, a Date and Time annotator can annotate the full value to give a value of `Key = Time`, `Value = Thu Nov 22 06:28:48 EST 2012`.

Key-value pairs

A Key-value annotation extracts data from a log record if it is in the format <key>=<value>. For example, ERROR-CODE = 4499. These Key-value pairs are used to list the values for each key in the Discovered Patterns section of the Search UI.

There are two categories of KVP annotations. The first is Key-value pairs that are separated by an equal sign (=). The second category is those separated by a colon (:). Each category has a different set of rules to determine what is a valid annotation.

Key-value pairs separated by an equal sign, '='

- Both the key and value must be one token
- The key can contain upper and lower case letters, dashes (-), underscores (_), and periods (.)
- The key must begin with a letter

- The value can contain upper and lower case letters, numbers, dashes (-), underscores (_), periods (.), at signs (@), and colons (:)
- The value can be surround by matching brackets [ ], parentheses ( ), angle-brackets < >, single quotes ' ', or double quotes " "
- The value must being with a letter or a number, and may have an optional dash (-) at the beginning
- A single whitespace character may be on one or both sides of the equal sign
- The single token rule for the value is disregarded when a concept is found for the value. For example, if a multi token date is identified as the value, the whole date, not just the first token, will be annotated.
- Users may add custom regular expressions to the dictionary located at `Log_Analytics_install_dir/unity_content/GA/GAInsightPack_<version>/extractors/ruleset/GA_common/dicts/userSpecifiedStrings.dict`. Matches to these regular expressions will be used when checking if the value is part of a larger concept.

Key-value pairs separated by a colon, ':'

- Both the key and value must be between 1 and 3 tokens
- The tokens can contain any character except whitespace or colons.
- Tokens must be separated by spaces or tabs
- The colon may have one or more spaces or tabs to the left and must have at least one space or tab to the right. There may be more than one space or tab to right of the colon
- The entire string must be on a line by itself

Timestamp formats

IBM Operations Analytics - Log Analysis is capable of annotating many commonly used timestamp formats. This appendix lists the supported timestamp formats.

Timestamp annotations, also called date-time annotations, are constructed from two base formats: a date format and a time format. Date-time annotation works as follows:

1. An annotator identifies date patterns in a text and annotates them with date annotations.
2. Another annotator identifies time patterns in the text and annotates them with time annotations.
3. The timestamp annotator then identifies and annotates specific patterns in which date and time annotations occur contiguously in the text.

Date annotation formats:

The date annotation formats are specified in the `Date_BI.aql` file.

View: `DateOutput`

File name: `Date_BI.aql`

The following date annotation formats are available.

Table 35. Date annotation formats

Format name	Pattern	Examples
D1	A date interval, where: • the month is a supported month format • the month can precede or follow the interval • the year is optional • commas (,) are optional	9 - 12 December 2012 December 9 – 12, 2012 9 - 12 december DEC 9 - 12
D2	A date that contains the suffixes th, st, nd, or rd, and where: • the word “of” is optional • the year is optional • commas (,) are optional • the month is a supported month format • the month can precede or follow the day	3rd December 2012 4th DEC Dec, 1st 2nd of December
D3	A date that contains the day of the week, and where: • the word “the” is optional • the suffixes th, st, nd, and rd are optional • the year is optional • commas (,) are optional • the month is a supported month format • the month can precede or follow the day	Sunday, the 3rd of December, 2012 Wed, 1st DEC
D4	A date that contains forward-slash characters (/), in the format Day/Month/Year, and where: • the month is a supported month format • the month follows the day • the year has four digits • the suffixes th, st, nd, and rd are optional	3/December/2012 1st/Dec/2012
D5	A date in the format Year-Month-Day or Year.Month.Day, where: • the year has four digits • the month has two digits • the day has two digits Because this pattern comprises only digits, leading zeros (0) might be required.	2012-01-30 2012.12.03
D6	A date in the format Day-Month-Year or Day/Month/Year, where: • the year can have two or four digits • the month and the day do not require leading zeros (0), even if they are single digits	30-1-12 3/12/2012
D7	A date in the format Month-Day-Year or Month/Day/Year, where: • the year can have two or four digits • the month and the day do not require leading zeros (0), even if they are single digits	1-30-12 12/3/2012

Time annotation formats:

The time annotation formats are specified in the `Time_BI.aql` file.

View: `TimeOutput`

File name: `Time_BI.aql`

The following time annotation formats are available.

Table 36. Time annotation formats

Format name	Pattern	Examples
T1	A time in the format Hour:Minute, where: • the hour need not be padded with leading zeros (0) if it is a single digit • seconds and milliseconds are optional • the minute is two digits • the second, if present, is two digits and preceded by a colon (:) • the millisecond, if present, is three digits and preceded by a colon (:) or a period (.)	2:05 20:30 02:05 23:11:59 23:11:59:120 23:11:59.120
T2	A time in T1 format, plus the year, where the year is four digits.	2:05 2012 23:11:59.120 2012
T3	A time in the format Hour:Minute:Seconds, where: • milliseconds are optional • the time zone is optional • the minute is two digits • the second is two digits and preceded by a colon (:) • the millisecond, if present, is three digits and preceded by a colon (:) or a period (.) • the time zone, if present, is: – preceded by a plus (+) or minus (-) sign – has no space preceding the plus (+) or minus (-) sign – has the format Hour:Minute, where both the hour and minute are two digits – contains no intervening spaces	3:11:59+05:30 2:05:59.120-01:05
T4	A time in T1 format, plus a 12-hr clock designation.	2:05 PM 11:11:59 a.m.
T5	A time in T1 format, plus a supported time zone format.	2:05 IST 11:11:59 PST

Table 36. Time annotation formats (continued)

Format name	Pattern	Examples
T6	A time in T1, T4, or T5 format, plus the time zone and year, where: • the time zone is optional • the year is optional • the time zone, if present: – is in digits – is preceded by a plus (+) or minus (-) sign – contains an optional colon (:) between the hour and the minute • the year, if present, is four digits	11:11:59 a.m. +05:30 11:11:59 PST -0030 2012

Date-time annotation formats:

The date-time (timestamp) annotation formats are specified in the `DateTime-consolidation_BI.aql` file.

View: `DateTimeOutput`

File name: `DateTime-consolidation_BI.aql`

The following date-time annotation formats are available.

Table 37. Date-time annotation formats

Format name	Pattern	Examples
DT1	A timestamp in the format DT, where: • D is the D1, D2, D3, D4, D5, D6, or D7 date format • T is the T1, T2, T3, T4, T5, or T6 time format	Sunday, the 3rd of December, 2012 23:11:59.120 IST 3/12/2012 2:11:59+05:30 6/10/12 2:48:28:381 MDT Thu Nov 22 06:28:48 EST 2012
DT2	A timestamp in the format D4:T3.	3/December/2012:2:00:00.000 1st/Dec/2012:23:11:59+05:30
DT3	A timestamp in the format D4:Hour:Minute:Seconds Z, where Z is an RFC 822-four-digit time zone format that conforms with the Java `SimpleDateFormat` class.	3/December/2012:02:00:00 -0030 1st/Dec/2012:23:11:59 +0530

For more information about the `SimpleDateFormat` class, see:

http://docs.oracle.com/javase/1.4.2/docs/api/java/text/SimpleDateFormat.html#rfc822timezone.

Other formats

The view `DateTimeOutput` also supports timestamp formats in these ISO date formats:

- yyyy-MM-ddTHH:mm:ss.SSSZ. For example, 2013-02-27T13:57:21.836+0000
- yyyy.MM.ddTHH:mm:ss.SSSZ. For example, 2013.02.27T13:57:21.836+0000

Variations of these formats are also supported:

- yyyy/MM/dd-HH:mm:ss.SSSZ. For example, 2013/02/27-13:57:21.123+0000
- yyyy/MM/dd-HH:mm:ss.SSS. For example, 2013/02/27-13:57:21.123
- yyyy-MM-dd-HH:mm:ss. For example, 2013-02-27-13:57:21

Date-time stamps with no year value

Some applications write logs with no year in the date/time stamp. For example, the UNIX messages log, such as shown here:

```
Apr 24 09:41:16 bluewashmachine symcfgd: subscriber 2 has left -- closed
 0 remaining handles

Apr 24 09:41:20 bluewashmachine rtvscand: New virus definition file loaded.
 Version: 150423c.

Apr 24 09:41:38 bluewashmachine kernel: type=1400 audit(1366792898.697:52164):
 avc:  denied
  { module_request } for  pid=18827 comm="smtpd" kmod="net-pf-10"
 scontext=system_u:system_r:postfix_smtpd_t:s0
 tcontext=system_u:system_r:kernel_t:s0 tclass=system

Apr 24 09:41:38 bluewashmachine kernel: type=1400 audit(1366792898.822:52165):
 avc:  denied
  { module_request } for  pid=18833 comm="proxymap" kmod="net-pf-10"
 scontext=system_u:system_r:postfix_master_t:s0
 tcontext=system_u:system_r:kernel_t:s0 tclass=system
```

In this case, the Generic-dateTime-Split splitter identifies the string Apr 24 09:38:58 as a valid date-time stamp. To meet the required date formats of IBM Operations Analytics, a valid year must be associated with the date-time string. The generic-dateTime-split splitter address this problem by placing a yyyy value at the beginning of the identified date-time stamp format. As a result, the timestamp now reads 2013 Apr 24 09:38:58.

You must update the timestamp format for files this type in the indexConfig. For example, if you want IBM Operations Analytics to ingest log records with the timestamp format MMM dd HH:mm:ss, the dateFormat must be specified as shown here.

```
 "timestamp": {
           "searchable": true,
           "filterable": true,
           "retrievable": true,
           "dataType": "DATE",
           "tokenizer": "literal",
           "sortable": true,
           "source": {
               "dateFormats": [
                   "yyyy MMM dd HH:mm:ss"
               ],
               "paths": [
                   "metadata.timestamp"
               ],
               "combine": "FIRST"
           },
           "retrieveByDefault": true
       },
```

The supported date formats without a year are:

- Apr 16 (MMM dd)
- 16 Apr (dd MMM)
- 16 April (dd MMM)
- April 16 (MMM dd)

Year end scenario: The `generic-dateTime-split` splitter applies the current year to any timestamp that is ingested where no year can be discerned from the log record. The exception is when the log record is ingested where current system time of the IBM Operations Analytics server identifies the month as January, but the incoming date/timestamp is December. In such situations, the year value that is applied is the current year minus 1.

Logs with no date

Some data sources that are ingested by IBM Operations Analytics do not support a date within the timestamp. For example, some lines of a log display the characteristic shown here in bold:

```
00:11:35.103 INFO  [main] - Server accepting connections on rmi://9.11.222.333:1099/

09:34:33.071 INFO  [main] - Server accepting connections on tcp://9.11.222.333:3035/
```

To ingest such a data source, IBM Operations Analytics provides a splitter rule set `Generic-timeOnly-Split`, which you can use along with the Generic-Annotator to ingest such a log. The splitter prepends a date to the identified time from each record of the log.

You must update the timestamp format for files of this type in the `indexConfig`. For example, in order for IBM Operations Analytics to ingest a log with records with timestamp such as `09:34:33.071`, the `dateFormat` must be specified as here.

```
 "timestamp": {
           "searchable": true,
           "filterable": true,
           "retrievable": true,
           "dataType": "DATE",
           "tokenizer": "literal",
           "sortable": true,
           "source": {
               "dateFormats": [
                   "MM/dd/yyyy HH:mm:ss.SSS"
               ],
               "paths": [
                   "metadata.timestamp"
               ],
               "combine": "FIRST"
           },
           "retrieveByDefault": true
       },
```

Only the date portion of the `dateFormat` is fixed. The time portion must reflect the format found in the incoming `logSource`.

The Generic annotator Insight Pack defines the Annotation Query Language (AQL) that supports log files with no date in the `timeOnly` splitter. The `timeOnly` splitter is defined in the file:

```
GAInsightPack_<version>/extractors/ruleset/timeOnlySplitter
```

Supported weekday formats:

The supported weekday formats are specified in the `wkday.dict` dictionary file.

Dictionary file: `dicts/wkday.dict`

The following long and short forms of the days of the week are supported. Both lower and upper cases, and upper case for first character, are supported.

Table 38. Supported weekday formats

Long form	Short form
monday	mon
tuesday	tue tues
wednesday	wed
thursday	thu thur thurs
friday	fri
saturday	sat
sunday	sun

Supported month formats:

The supported month formats are specified in the `month.dict` dictionary file.

Dictionary file: `dicts/month.dict`

The following long and short forms of the months of the year are supported. Both lower and upper cases, and upper case for first character, are supported.

Table 39. Supported month formats

Long form	Short form
january	jan
february	feb
march	mar
april	apr
may	No short form.
june	jun
july	jul
august	aug
september	sept sep
october	oct
november	nov
december	dec

Supported time zone formats:

The supported time zone formats are specified in the `timeZone.dict` dictionary file.

Dictionary file: `dicts/timeZone.dict`

The following 12-hour clock formats, in both lower and upper case, are supported:

- a.m.
- p.m.
- AM
- PM

The following page on SMC lists the supported time zone formats: What are the supported time zones?

Note: If a supplied timezone is not supported, the system behaviour is to default to reading the system timezone and normalizing to that.

Javacore Insight Pack

The Javacore Insight Pack provides the capability to ingest and perform metadata searches against javacore files in IBM Operations Analytics - Log Analysis.

Support

This document describes the version of the Javacore Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Javacore Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html.

The Javacore Insight Pack supports the ingestion of Linux javacore files produced by the IBM JRE versions 6.0 and 7.0.

The Javacore Insight Pack can be run on Linux.

Installing the Javacore Insight Pack

Instructions on how to install the Javacore Insight Pack

About this task

The Javacore Insight Pack is installed using the `pkg_mgmt` utility.

Procedure

1. Upload the Javacore Insight Pack archive file, `JavacoreInsightPack_<`*`version`*`>.zip`, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Javacore Insight Pack with the `pkg_mgmt.sh` command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install
   <path>/JavacoreInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Javacore Insight Pack.

Javacore configuration artifacts

The Javacore Insight Pack configuration artifacts.

Splitter file set:
Javacore-Split

Annotator file set:
Javacore-Annotate

Source type:
Javacore

Collection:
Javacore-Collection1

The Javacore Insight Pack does not provide any Log File Agent (LFA) configuration files (.conf and .fmt). Javacore files are ingested using the Data Collector Client.

Javacore log file splitter

The Javacore Splitter uses a filter to reduce the size of the data that is grouped into each record.

Javacore files contain a single timestamp and so are processed as one log record. The Javacore splitter ensures that the contents of the javacore file are grouped into a single log record. Javacore files contain a lot of information on the state of the JVM at the time of the javacore dump. As a result javacore files can be large in size. However, not all of this data needs to be indexed and annotated. The Javacore Splitter uses a filter to reduce the size of this data. The following entries in a javacore file are filtered by the splitter:

- `1TISIGINFO`
- `1TIDATETIME`
- `1TIFILENAME`
- `2XHOSLEVEL`
- `3XHCPUARCH`
- `3XHNUMCPUS`
- `1CIJAVAVERSION`
- `1CIVMVERSION`
- `1CIJITVERSION`
- `1CIGCVERSION`
- `1CIRUNNINGAS`
- `1CICMDLINE`
- `1CIJAVAHOMEDIR`

The entire thread stack trace is also filtered. All other data is blocked by the filter.

Turning off the filter

The filter can be configured to be on or off. It is on by default. To turn off the filter, follow these steps:

1. Create a file called javacore_insightpack.config
2. Add the following key/value to the file: `splitter.filter.on=false`
3. Save the file and copy it to your home directory on the IBM Operations Analytics - Log Analysis system

Note: Turning off the filter will pass the entire javacore file contents into IBM Operations Analytics - Log Analysis. This will affect the performance of searches on the IBM Operations Analytics - Log Analysis Search workspace as the entire javacore file contents will be contained in the logRecord annotation.

Javacore log annotations

The fields that are defined in the index configuration file, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records.

Fields are extracted from the fields of a log record or collected from metadata around the log file.

Table 40. Log record annotations

Field	Attributes

timestamp	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = true
	filterable = true
	searchable = true

SignalInfo	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

FileName	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

OSLevel	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

CPUArchitecture	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

NumCPUs	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

JavaVersion	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

Table 40. Log record annotations (continued)

Field	Attributes

VMVersion	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

JITVersion	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

GCVersion	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

RunningAs	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

CommandLineArgs	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

JavaHomeDir	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = true
	searchable = true

NumThreadsRunning	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

NumThreadsSuspended	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

NumThreadsBlocked	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

Table 40. Log record annotations (continued)

Field	Attributes

NumThreadsParked	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

NumThreadsConditionWaiting	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

CurrentThreadStacktrace	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

AllThreadStacktraces	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

Table 41. Metadata annotations

Field	Attributes

application	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

middleware	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

datasourceHostname	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

hostname	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

service	dataType = TEXT
	retrievable = true
	retrieveByDefault = true
	sortable = false
	filterable = false
	searchable = true

Ingesting Java core log data

About this task

To ingest a javacore file perform the following steps:

Note: This assumes that the logsource for the Javacore Insight Pack has already been created.

Procedure

1. Edit the file: `<HOME>/utilities/datacollector-client/javaDatacollector.properties`:
   a. Edit the **`logFile`** value to correspond to the name (including path) of the javacore file to be ingested.
   b. Edit the **`logpath`** value to be the same as the Javacore Logsource log path value.
2. Run the following command: `<HOME>/ibm-java/bin/java -jar datacollector-client.jar`

Syslog Insight Pack

The Syslog Insight Pack extends IBM Operations Analytics - Log Analysis functionality so it can ingest and perform metadata searches against syslog data logging.

This document describes the version of the Syslog Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Syslog Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html.

The formatted log includes specific property values in a name/value pair format to aid data ingestion.

Syslog is a standard for recording events to track system activity and to diagnose problems. It separates the software that generates messages from the system that stores them and the software that reports and analyzes them. Implementations are available for many operating systems. Specific configuration permits the direction of messages to various devices (console), files (`/var/log/`) or remote syslog servers. rsyslog is an open source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network. It implements the basic syslog protocol.

Supported versions

The Syslog Insight Pack can be installed with IBM Operations Analytics - Log Analysis (SCALA) 1.1.0.0 and higher.

rsyslog version 3 is included as the default syslog tool for RHEL 5.2, and this is the minimum version supported by IBM Operations Analytics - Log Analysis. IBM Operations Analytics - Log Analysis supports rsyslog version 3, 5, 6 and 7. IBM Operations Analytics - Log Analysis supports the rsyslog list format, which is recommended by rsyslog, for version 7 and higher of rsyslog.

Syslog installation

Instructions on how to install the Syslog Insight Pack.

Procedure

1. Create a directory called `<HOME>/IBM/LogAnalysis//unity_content/Syslog` on the system where IBM Operations Analytics - Log Analysis is installed and upload the Syslog Insight Pack archive file, `SyslogInsightPack_<version>.zip`, to that directory.
2. Install the Syslog Insight Pack with the `pkg_mgmt.sh` command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install <path>
   /SysLogInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Syslog Insight Pack.
3. Deploy the log file agent with the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa <path>
   /SysLogInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Syslog Insight Pack.

Related tasks:

Installing an Insight Pack
You can download an Insight Pack to extend the capabilities of IBM Operations Analytics - Log Analysis from Service Management Connect. This topic outlines how to install an Insight Pack.

Syslog configuration

Configuration of the Insight Pack are described here as a reference for users.

rsyslog requirements

Before ingesting rsyslog log files, both rsyslog and IBM Operations Analytics - Log Analysis must be configured to ensure that rsyslog log files are output in a format that can be processed by IBM Operations Analytics - Log Analysis Syslog Insight Pack.

Add the scalaLogFormat template to rsyslog:

- For rsyslog 7 and higher, which support the list format:
  1. Open the `/etc/rsyslog.conf` for edit.
  2. Add the following template:

     ```
     template(name="scalaLogFormat" type="list") {
       property(name="timestamp" dateFormat="rfc3339")
       constant(value=" host=")
       property(name="hostname")
       constant(value=", relayHost=")
       property(name="fromhost")
       constant(value=", tag=")
       property(name="syslogtag")
       constant(value=", programName=")
       property(name="programname")
       constant(value=", procid=")
       property(name="procid")
       constant(value=", facility=")
       property(name="syslogfacility-text")
       constant(value=", sev=")
       property(name="syslogseverity-text")
       constant(value=", appName=")
       property(name="app-name")
     ```

```
  constant(value=", msg=")
  property(name="msg" )
  constant(value="\n")
        }
```

   The generated log record is formatted as

   ```
   2013-07-15T21:30:37.997295-04:00 host=co052065, relayHost=co052065,
   tag=rhnsd[12171]:, programName=rhnsd, procid=12171, facility=daemon,
   sev=debug, appName=rhnsd, msg= running program /usr/sbin/rhn_check
   ```

3. Associate the scalaLogFormat template with the log files to be ingested by IBM Operations Analytics - Log Analysis. It will log all log entries to <filename> in addition to any other associations in the configuration file. For example:

   ```
   *.*          /var/log/<filename>.log;scalaLogFormat
   ```

4. Restart the rsyslog daemon.

   Refer to the rsyslog documentation http://rsyslog.com/doc for more information.

5. Ensure that the output file created for IBM Operations Analytics - Log Analysis can be read by your IBM Operations Analytics - Log Analysis user, and write that file directly to the monitored logsource directories.

   For example:

   ```
   *.* <HOME>/logsources/SyslogInsightPack
   SCALA.log;scalaLogFormat
   ```

- For versions of rsyslog older than version 7:
  1. Open the `/etc/rsyslog.conf` for edit.
  2. Add the following template (legacy format):

     ```
     $template scalaLogFormat,"%TIMESTAMP:::date-rfc3339% host=%HOSTNAME%,
     relayHost=%FROMHOST%, tag=%syslogtag%, programName=%programname%,
     procid=%PROCID%, facility=%syslogfacility-text%, sev=%syslogseverity-text%,
     appName=%APP-NAME%, msg=%msg%\n"
     ```

     The generated log record is formatted as

     ```
     2013-07-15T21:30:37.997295-04:00 host=co052065, relayHost=co052065,
     tag=rhnsd[12171]:, programName=rhnsd, procid=12171, facility=daemon,
     sev=debug, appName=rhnsd, msg= running program /usr/sbin/rhn_check
     ```

  3. Associate the scalaLogFormat template with the log files to be ingested by IBM Operations Analytics - Log Analysis. It will log all log entries to <filename> in addition to any other associations in the configuration file. For example:

     ```
     *.*          /var/log/<filename>.log;scalaLogFormat
     ```

  4. Restart the rsyslog daemon.

     Refer to the rsyslog documentation http://rsyslog.com/doc for more information.

  5. Ensure that the output file created for IBM Operations Analytics - Log Analysis can be read by your IBM Operations Analytics - Log Analysis user, and write that file directly to the monitored logsource directories.

     For example:

     ```
     *.* <HOME>/logsources/SyslogInsightPack SCALA.log;scalaLogFormat
     ```

For more information about rsyslog configuration files, see: http://www.rsyslog.com/doc/rsyslog_conf.html

Configuration artifacts

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 42. Insight Pack configuration artifacts

Artifact	Name for the `syslog` log
Splitter rule set	`Syslog-Split`
Annotator rule set	`Syslog-Annotate`
Source type	`Syslog`
Collection	`Syslog-Collection1`

Note: Data sources are not predefined. A user with administrator privileges must define at least one syslog data source type and collection before the application can be used.

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the `<HOME>/IBM-LFA-6.30/config/lo` directory (where <HOME> is the install location of IBM Operations Analytics - Log Analysis):

- `SyslogInsightPack-lfasyslog.conf`: Configuration file for the syslog log file agent.
- `SyslogInsightPack-lfasyslog.fmt`: Matches records for the syslog log files.

Splitting and annotation AQL modules

Splitting and annotation are handled by the following Annotation Query Language (AQL) modules.

Table 43. Insight Pack AQL modules

AQL Module	Description
`common`	Common code module that is used across multiple insight packs (for example, to recognize timestamp formats).
`dateTimeSplitter newlineSplitter`	Splitter modules for syslog log files.
`annotatorSyslog`	Annotator module for syslog log files.

Log file splitters

The splitters provided with the Insight Pack are described here as a reference for users.

The Insight Pack supports the ISO 8061 timestamp, yyyy-mm-ddTHH:mm:ss.SSSSSSX where X is the GMT offset. Each log record begins with an ISO-formatted timestamp and is split across timestamp boundaries. An example of the ISO-formatted timestamp generated by rsyslog is:

```
2013-06-26T12:21:29.471400-04:00
```

The IBM Operations Analytics - Log Analysis index function is limited to milliseconds in the date format. The Syslog Insight Pack will annotate the timestamp and round up the microseconds. The sample ISO-formatted timestamp will be indexed with the following format for the index configuration:

```
yyyy-mm-ddTHH:mm:ss.SSSX
```

and rendered in the IBM Operations Analytics - Log Analysis search UI as:

```
06/26/2013 16:21:29.471-04:00
```

Log file annotations

The annotations that are defined by the log file index configurations are described here.

The index configuration file is included in the Insight Pack in the `sourcetypes.json` file (found at `<path>/SyslogInsightPack_<version>/metadata`), where <path> is the path where you saved the Syslog Insight Pack.

You can customize the artifacts in the index configuration file by creating another source type and modifying a copy of the Syslog index configuration.

The following sections describe the fields that are defined in the index configuration file. These fields, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records. Fields are extracted from the fields of a log record or collected from metadata around the log file. Each table gives the names of the fields (these names correspond to fields in the IBM Operations Analytics - Log Analysis Search workspace), descriptions of how the related annotations are made, and the index configuration attributes assigned to the fields.

Log record annotations

The following table lists the index configuration fields that relate to log record annotations. Each field corresponds to part of a syslog log record.

Table 44. Log record index configuration fields

Field	Description	Attributes

syslogHostname	The hostname from the message.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

syslogRelayHostname	The hostname of the system the message was received from (in a relay chain, this is the system immediately in front, and not necessarily the original sender).	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

tag	The TAG from the message.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true

Table 44. Log record index configuration fields (continued)

Field	Description	Attributes

programName	The static part of the tag as defined by BSD syslogd. For example, when TAG is "named[12345]", programName is "named".	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

processID	The contents of the PROCID field.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true

facility	The facility from the message.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

severity	The severity from the message (in text form).	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

syslogAppName	The APP-NAME from the message.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

message	The MSG (the message) in the log record.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true

Metadata annotations

The following table lists the index configuration fields that relate to metadata annotations.

Table 45. Metadata annotation index configuration fields

Field	Description	Annotation attributes
datasourceHostname	The host name that is specified in the data source.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true

timestamp	The timestamp from the log record.	dataType = DATE
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true

Table 45. Metadata annotation index configuration fields (continued)

Field	Description	Annotation attributes
application	The application name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
middleware	The middleware name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
hostname	The host name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
service	The service name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
logRecord	The entire log record output by the splitter.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Log file considerations

This section covers log rotation and the streaming of logs to a centralized server.

Log rotation

Linux provides logrotate to configure rotation. The global options are specified in /etc/logrotate.conf. Options for specific files (which can over ride the global options) are in /etc/logrotate.d for each log file.

Note: For more information on the logrotate command, look up the UNIX command documentation online.

When the logs are rotated, the log file is renamed with a .1 extension (assuming the dateext option is not included) and truncated to zero length. The rotation configuration also determines how often old logs are archived, that is, old *.n are removed or archived. The log locations are defined in /etc/rsyslog.conf (which is by default /var/log).

The recommended method for loading data is to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote system where rsyslogd is executing to push data, or use LFA installed where IBM Operations Analytics - Log Analysis is installed to pull data. Create .conf and .fmt files specific to each log file, the destination of which is specified in /etc/rsyslog.conf. The data source definition should specify <filename>.1. This ensures all log records are processed. However, they will only be sent when the rotation occurs. The user can configure rotation to occur more frequently to minimize the time lag from the current log. Alternatively, the user can monitor the <filename>. When you monitor <filename>, there is a window where log entries will not be forwarded if log entries are rotated before LFA has polled the data. If the user has configured daily or hourly rotation, they can monitor the `*.1` file name to avoid a window where log entries are not forwarded.

The best practice is to rotate the logs frequently so <filename>.1 has recent data for IBM Operations Analytics - Log Analysis to ingest. The default log rotation is weekly. You can change the rotation for syslog in `/etc/logrotate.d/syslog`. To change it to daily, add the **`daily`** option in `/etc/logrotate.d/syslog` configuration file. If the logs are large, you can rotate them based on a size with the **`size`** option.

The following two sections describe configuration changes to `SyslogInsightPack-lfasyslog.conf` and `SyslogInsightPack-lfasyslog.fmt` only when dealing with rotating log files.

Logfile Agent Configuration - SyslogInsightPack-lfasyslog.conf

The following parameters should be specified to monitor the rotating files:

```
LogSources=<syslog directory to monitor>/<logfilename>.*
FileComparisonMode=CompareByAllMatches
```

Logfile Agent Configuration - SyslogInsightPack-lfasyslog.fmt

Use the following specification to avoid defining multiple data sources because of the file name changes when the log rotates. This allows a fixed file name in the log path specification.

```
// Matches records for any Log file:
//

REGEX AllRecords
(.*)
hostname LABEL
-file <logfilename>.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

Centralized logging

If you are streaming logs to a central server, the best practice is to stream to one consolidated log for ingestion by IBM Operations Analytics - Log Analysis. The same best practices are applicable to the consolidated file as for the logs in the non-server scenario. The logs should be rotated frequently so <filename>.1 has recent data, and the Log File Agent should be used to pull or push the data to the IBM Operations Analytics - Log Analysis server.

To configure rsyslog to stream logs to a central server (for example, 192.168.1.1), do the following:

1. Add the following to each client (or edge systems) `/etc/rsyslog.conf` file to stream to the central server:

   ```
   $ModLoad imuxsock

   $ModLoad imklog
   ```

```
 # Provides UDP forwarding. The IP is the server's IP address
 *.* @192.168.1.1:514
 # Provides TCP forwarding. But the current server runs on UDP
 # *.* @@192.168.1.1:514
```

2. On the central server (for example, with IP address 192.168.1.1) add the following to `rsyslog.conf`:

```
 $ModLoad imuxsock

 # provides kernel logging support (previously done by rklogd)
 $ModLoad imklog

 # Select the syslog reception of UDP or TCP. For TCP, load imtcp by
uncommenting $ModLoad imtcp.
#$ModLoad imudp
#$ModLoad imtcp

# Select the syslog reception port.  For TCP, uncomment InputServerRun 514
#$UDPServerRun 514
#$InputTCPServerRun 514
# This FILENAME template generates the log filename dynamically.
# You can replace the specification with variables applicable to
# your site.
# The scalaLogFormat template formats the message required
# for ingestion by SCALA.
 $template FILENAME,"/var/log/scala-syslog.log"
 $template scalaLogFormat,"%TIMESTAMP:::date-rfc3339% host=%HOSTNAME%,
 relayHost=%FROMHOST%, tag=%syslogtag%, programName=%programname%,
procid=%P  ROCID%,  facility=%syslogfacility-text%,
sev=%syslogseverity-text%, appName=%APP-NAM E%, msg=%msg%\n"


 # Log all messages to the dynamically formed file.
 *.* ?FILENAME;scalaLogFormat
```

3. Decide whether you are going to use either the UDP or TCP configuration and comment out the other. For example, to use TCP update the code section as follows:

```
# Select the syslog reception of UDP or TCP. For TCP, load imtcp
# by uncommenting $ModLoad imtcp.
#$ModLoad imudp
$ModLoad imtcp

# Select the syslog reception port.  For TCP, uncomment
# InputServerRun 514
#$UDPServerRun 514
$InputTCPServerRun 514
```

Web Access Logs Insight Pack

The Web Access Logs Insight Pack provides the capability to ingest and perform metadata searches against Web Access Logs (Apache IHS, JBoss, Apache Tomcat) in IBM Operations Analytics - Log Analysis.

This document describes the version of the Web Access Logs Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Web Access Logs Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html.

The Web Access Log Insight Pack ingests records in the web server access log. Server access logs record information about all requests handled by a web server. This can include information about the IP address of the client making the request, userid of the person making the request (determined by the HTTP authentication), timestamp when the request was received, the request line, etc. The access log is highly configurable, and the LogFormat or pattern is used to define the contents and format of the access log.

Note: The access log is different from the web server log, `server.log`.

By using the LogFormat directive or pattern to create a delimiter-separated value (DSV) access log file, the Web Access Log Insight Pack can annotate and index access logs for ingestion, annotation, and indexing into IBM Operations Analytics - Log Analysis. The Web Access Logs Insight Pack supports the following web servers (and any others which enable the LogFormat specification required by this insight pack):

- Apache/IBM HTTP Server 8.5.5.0
- Apache Tomcat 7.0.42, 6.0.37
- JBoss v7

The Web Access Logs Insight Pack can be installed with IBM Operations Analytics - Log Analysis 1.1.0.2 and higher.

Installing the Web Access Logs Insight Pack

Instructions on how to install the Web Access Logs Insight Pack

Before you begin

The prerequisites of the Web Access Logs Insight Pack are:

- IBM Operations Analytics - Log Analysis v1.1.0.2
- DSV Toolkit v1.1.0.1 or higher

  Note: The DSV is only needed if you are generating a new insight pack to support other access log formats.

About this task

The Web Access Logs Insight Pack is installed using the `pkg_mgmt` utility.

Procedure

1. Upload the Web Access Logs Insight Pack archive file, `WebAccessLogInsightPack_<`*`version`*`>.zip`, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Web Access Logs Insight Pack uisng the `pkg_mgmt.sh` command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install
   <path>/WebAccessLogInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Web Access Logs Insight Pack.
3. (Optional) If you are using the Log File Agent to load the data into IBM Operations Analytics - Log Analysis, deploy the log file agent configuration files with the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa
   <path>/WebAccessLogInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Web Access Logs Insight Pack.

Configuring the Web Access Logs Insight Pack

Instructions on how to configure the Web Access Logs Insight Pack.

Procedure

1. In the IBM Operations Analytics - Log Analysis Administrative Settings workspace, create a new log source for the log file to be monitored. The source type should be `WebAccessLog`.
2. On the web server, customize the access log format to a delimiter-separated value output (DSV) that can be consumed by the Web Access Log Insight Pack and IBM Operations Analytics - Log Analysis. The syntax to customize the log format is different for each web server, but the generated log will be the same. Following is the log format directive for the supported web servers:

   For Apache/IHS

   a. Edit `<ServerRoot>/conf/httpd.conf` file, where `<ServerRoot>` is the root installation path.

      1) Add the following log format directive:

      ```
      LogFormat "Apache/IHS,%h,%l,%u,%t,%m,\"%r\",%>s,%b,%D,
      \"%{Referer}i\",\"%{User-Agent}i\"" scalaAccessLog
      ```

      2) Update the access log directory specification to use the `LogFormat` directive:

      ```
      CustomLog logs/access_log scalaAccessLog
      ```

      3) Comment out the following line by prefixing it with #:

      ```
      CustomLog logs/access_log common
      ```

   b. Restart the web server.

   c. The generated access files are at `<ServerRoot>/logs`.

   For JBoss

   a. Edit the file `<JBOSS_HOME>/jboss-eap-6.1/standalone/configuration/standalone.xml`

   b. Find the XML element `subsystem xmlns="urn:jboss:domain:web:1.4"` and add the following `<access_log>` element:

   ```
   <subsystem xmlns="urn:jboss:domain:web:1.4"
    default-virtual-server="default-host" native="false>
     <connector name="http" protocol="HTTP/1.1" scheme="http"
      socket-binding="http"/>
     <virtual-server name="default-host" enable-welcome-root="true">
       <alias name="localhost"/>
       <alias name="example.com"/>
       <access-log prefix="access-log." pattern="JBoss,%h,%l,%u,%t,
        %m,&quot; %r&quot;,%s,%b,%D,&quot;%{Referer}i&quot;
        ,&quot;%{User-Agent}i&quot;">
         <directory path="." relative-to="jboss.server.log.dir"/>
       </access-log>
     </virtual-server>
   </subsystem>
   ```

   c. Restart the JBoss App Server

   d. Look for the access log file in `<JBOSS_HOME>/standalone/log`

      Where `<JBOSS_HOME>` is the directory where you installed JBoss

   For Apache Tomcat

   a. Edit the file `<tomcat-dir>/conf/server.xml` where `<tomcat-dir>` is the installation root and add the following log format:

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
 directory="logs"
 prefix="localhost_access_log." suffix=".txt"
 pattern="Tomcat,%h,%l,%u,%t,%m,&quot;%r&quot;,%s,%b,%D,
 &quot;%{Referer}i&quot;,&quot;%{User-Agent}i&quot;"
/>
```

b. Restart the web server using the scripts in `<tomcat-dir>/bin`

c. The log files are written on `<tomcat-dir>/logs/localhost_access_log.<date>.txt`

3. (Optional) Configure the Log File Agent to monitor rotated logs. This step is only required if your web server is configured to rotate log files and you are using the Log File Agent to ingest the log files.

   Note: Access logs are rotated by default for Apache Tomcat and JBoss. Access Logs are not rotated by default for Apache/IHS. For instructions on how to configure log rotation for Apache/IHS, see "Web Access Logs Best Practices" on page 161.

   Each web server has different syntax on how to specify rotation and the generated filename. By default, a rotated log has a timestamp or a number in the filename. Specify the log filename pattern in the `WebAccessLogInsightPack-lfadsv.conf` file that is applicable to your web server.

   a. In `WebAccessLogInsightPack-lfadsv.conf`, update LogSources to monitor all the files in the directory:

   ```
   LogSources=<web server log directory to monitor>/
   <access_log_filename_without_timestamp>*
   FileComparisonMode=CompareByAllMatches
   ```

   b. Update `WebAccessLogInsightPack-lfadsv.fmt` to specify a fixed filename so you can use the same fixed name in the path of the IBM Operations Analytics - Log Analysis logsource configuration. You only need to define one logsource with this path, and LFA will monitor all the files in the directory because you specified wildcard file naming in the `WebAccessLogInsightPack-lfadsv.conf` specification.

   ```
   // Matches records for any Log file:
   // REGEX AllRecords
   (.*) hostname LABEL
   -file web_server_access.log
   RemoteHost DEFAULT logpath PRINTF("%s",file)
   text $1
   END
   ```

   LFA will monitor all the log records in the directory (as specified by the LogSources value). This ensures no data will be lost as logs are rotated. However, LFA is allocating resources to monitor each file. This results in unnecessary resources since the rotated logs will not be updated again. It is a best practice to periodically archive old logs so LFA can release resources monitoring static files. For Unix, you can use tools like logrotate and cron to schedule archiving of old logs.

4. If you want to collect logs from multiple web servers, or want to ingest an archive of rotated logs, the recommended method for loading data is to use the Data Collector client.

Web Access Logs splitter rules

Splitting describes how IBM Operations Analytics - Log Analysis separates physical log file records into logical records using a logical boundary such as time stamp or a new line.

The Web Access Log Insight Pack will split log records on new line boundaries.

Web Access Logs annotation rules

After the log records are split, the logical records are sent to the annotation engine. The engine uses rules to extract important pieces of information that are sent to the indexing engine.

According to the required configuration, the format of a web access log file is:

```
<webServerType>,<clientIP>,<ident>,<auth>,<timestamp>,<verb>,
"<request>",<response>,<bytes>,<responseTime>,"<referrer>",
"<agent>"
```

For example:

```
Apache/IHS,119.63.193.107,-,-,[22/Jul/2013:18:12:37 +0200],GET,
"GET / HTTP/1.1",200,3759,324,"-",
"Baiduspider+(+http://www.baidu.jp/spider/)"
```

Where the following formatting applies:

Table 46. Web Access Logs formatting

Field	Format String	Description
`webServerType`	`Apache/IHS, JBoss, Tomcat`	Web server that generated this log
`clientIP`	`%h`	Remote hostname or IP address
`ident`	`%l`	Remote logname
`auth`	`%u`	Remote user if the request was authenticated
`timestamp`	`%t`	Time the request was received, in the common log format [dd/MMM/yyyy:HH:mm:ss Z]
`verb`	`%m`	Request method (GET, POST, etc)
`request`	`%r`	First line of request (method and request URI)

`response`	`%s`
	`%>s (Apache/IHS)`
		HTTP status code of the response

`bytes`	`%b`	Bytes sent (excluding HTTP header), or "-" if 0
`responseTime`	`%D`	Time taken to process request in millis (Tomcat, JBoss) or microseconds (Apache/IHS)
`referrer`	`%{Referer}i`	Referrer on the request
`agent`	`%{User-Agent}i`	User agent on the request

If you have a mixed web server environment, and you are tracking the **responseTime** (perhaps in a custom app), you may need to normalize the data. The **webServerType** can be used to know if the **responseTime** is in millisecond or microsecond.

Log File Agent configuration

You can use the IBM Tivoli Log File Agent to load data into the IBM Operations Analytics - Log Analysis.

The following Log File Agent configuration files will be installed in `<HOME>/IBM-LFA-6.30/config/lo` directory when **`pkg_mgmt`** is run with the **`-deploylfa`** option.

- `lfamb.conf` - Configuration for the web access log file agent.
- `lfamb.fmt` - Matches records for the web access log files.

Web Access Logs index configuration

To control how IBM Operations Analytics - Log Analysis indexes records from a log file, you can create indexing settings for your content Insight Pack.

The fields that are defined in the index configuration file, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records.

Fields are extracted from the fields of a log record or collected from metadata around the log file.

Table 47. Log index configuration

Field	Description	Attributes

datasourceHostname	hostname from the logsource definition	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true
		source = metadata

logRecord	complete log record text	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotate

webServerType	type of web server (Apache, JBoss, Tomcat)	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

clientIP	remote hostname or IP address	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true
		source = annotate

ident	remote logname	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

Table 47. Log index configuration (continued)

Field	Description	Attributes

auth	remote user if the request was authenticated	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

timestamp	Time the request was received. The date format is:
	dd/MMM/yyyy:HH:mm:ss Z
		dataType = DATE
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

verb	request method (GET, POST, etc)	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

request	first line of request (method and request URI)	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotate

response	HTTP status code of the response	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

bytes	bytes sent (excluding HTTP header)	dataType = LONG
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotate

responseTime	time taken to process request in milliseonds (Tomcat, JBoss) or microseconds (Apache)	dataType = LONG
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotate

referrer	referrer on the request	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotate

Table 47. Log index configuration (continued)

Field	Description	Attributes

agent	user agent on the request	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotate

Web Access Logs configuration artifacts

The Web Access Logs Insight Pack configuration artifacts.

The following artifacts are created when the Insight Pack is installed:

WebAccessLog-Split
Splitter rule set.

WebAccessLog-Annotate
Annotator rule set

WebAccessLog
Source type

WebAccessLog-Collection1
Collection

Generating an alternate Web Access Log Insight Pack

Some customers may not be able to change the log format. In this situation you may need to create an insight pack that recognizes your log format, and which reuses the files provided by the Web Access Log Insight Pack.

About this task

The following steps describe how the user creates an Insight Pack re-using the files provided by the Web Access Log Insight Pack to recognize their log format. This assumes you have knowledge of the DSV toolkit (see the readme in the DSV toolkit docs directory for more information). Use the latest DSV Toolkit (at least version 1.1.0.1) available.

Procedure

1. Make a copy of the `WebAccessLogModDSV.properties` file, and update the copied file to contain the log fields and delimiter in your log. The file is found in `<HOME>/IBM/LogAnalysis/unity_content/WebAccessLog/WebAccessLogInsightPack_v1.1.0.0/dsv`
2. In the `WebAccessLogModDSV.properties` file, change the **`aqlModulename`** which is also used to generate the Insight Pack name. For example, change **`aqlModulename`** to `WebAccessLogMod`.
3. Copy the DSV toolkit, `DSVToolkit_v1.1.0.1.zip`, to `<HOME>/IBM/LogAnalysis/unity_content` and unzip it.
4. Invoke the python script to generate a new DSV Insight Pack using the updated DSV property file:

```
python dsvGen.py
<HOME>/IBM/LogAnalysis/unity_content/WebAccessLog/WebAccessLogInsightPack_v1.1.0.0/dsv/
WebAccessLogModDSV.properties -d
```

5. Edit the generated annotation module to change the byte annotation to be the same as found in the Web Access Log annotation:

   ```
   <HOME>/IBM/LogAnalysis/unity_content/WebAccessLog/WebAccessLogInsightPack_v1.1.0.0/dsv/
   extractors/ruleset/WebAccessLog/annotations.aql
   ```

   Replace:

   ```
   create view bytesFinal as
       select stripQuotes(D.bytes) as bytes
       from DelimiterSplit D;
   ```

   with the following:

   ```
   /*-------------------------------------------------------------------*/
   /* Tweak the AQL to annotate only bytes that appear as numeric digits */
   /* Igore bytes that are written to the log as "-"                    */

   create view bytesBasic as
       select stripQuotes(D.bytes) as bytes
       from DelimiterSplit D;

   create view bytesConsolidated as
       extract regex /(\d+)/
       on D.bytes
       return
           group 0 as match
           and group 1 as bytes
       from bytesBasic D;

   create view bytesFinal as
       select GetText(D.bytes) as bytes
       from bytesConsolidated D;
   /*-------------------------------------------------------------------*/
   ```

6. To restart IBM Operations Analytics - Log Analysis run the following command from the <HOME>/IBM/LogAnalysis/utilities directory:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

   Note: Space is not a valid delimiter for the DSV Toolkit. If you use space as a delimiter, you have to update the generated annotator and splitter to recognize it.

Web Health Check custom app

The Web Health Check custom app visualizes data from the web access logs.

The Web Health Check custom app contains the following charts:

Total Web Requests
Plots the number of web requests from the last day in a line chart.

Response Time
Plots the response times from the last day in a point chart.

Abnormal Response Code
Plots all non-200 response codes over the last day in a bubble chart.

Worst Client Response Times
Plots the maximum response times of each client IP (which is not 127.0.0.1) over the last day in a bubble chart.

Response Time SLA Violations (>100)

Plots the number of web requests where the response time is > 100 over the last day in a point chart. The query can be configured by the user to match the his SLA rules.

Note: For Apache IHS web servers, the response time is given in microseconds. For Apache Tomcat and JBoss web servers, the response times are in milliseconds.

These charts also support drill down. You can double-click on any data point in the chart to open a Search workspace that is scoped to the log records that make up that data point.

Configuring the Web Health Check custom app

The available options for customizing the Web Health Check custom app

About this task

By default, the Web Health Check custom app displays data for the relative time interval **Last Day**. The time filter can be changed by editing the Chart Settings, and specifying the **Time Filter** in the **Query** tab.

Create a new Web Health Check custom app by copying the Web Health Check.appExmpl in the <HOME>/AppFramework/Apps/ WebAccessLogInsightPack_v1.1.0.2 directory.

Rename the copied file as follows: <newname>.app.

To customize the new app, complete following steps:

Procedure

1. The log source used by the Web Health Check custom app is called accessLog. Change the logsources parameter for each chart that you want to change.

   For example:

   ```
   "logsources":[
     {
      "type":"logSource",
      "name":"/accessLog"
     }
    ],
   ```

2. The relative time interval for each chart defaults to the last day. Change the **timefilters** parameter for each chart to specify a new default relative time interval. The granularity can be specified as second, minute, hour, day, week, month, or year.

   For example:

   ```
   "filter":{
      "timefilters": {
     "granularity" : "day",
     "lastnum" : 1,
     "type": "relative"
     }
    },
   ```

3. The Response Time SLA Violations charts defaults to responseTime >100. Change the query parameter to define a different SLA violation rule.

   For example:

   ```
   {
     "name":"ResponseTimeSLAViolations",
     "type":"FacetedSearchQuery",
     "start": 0,
     "results": 0,
     "value":{
      "filter":{
       "timefilters": {
   ```

```
    "granularity" : "day",
    "lastnum" : 1,
    "type": "relative"
   }
  },
  "logsources":[
   {
    "type":"logSource",
    "name":"/accessLog"
   }
  ],
  "query":"responseTime: > 100",
  "outputTimeZone":"UTC",
  "getAttributes":[
   "timestamp",
   "responseTime"
  ],
  "sortKey": [
   "-timestamp"
  ],
  "facets":{
   "timestamp":{
    "date_histogram":{
     "field":"timestamp",
     "interval":"hour",
     "outputDateFormat":"yyyy-MM-dd'T'HH:mm:ssZ",
     "outputTimeZone":"UTC"
    }
   }
  }
 }
}
```

4. After you edit the custom app file, refresh the Custom Apps pane in the Search UI in order to pick up the changes.

Web Access Logs Best Practices

Best practices for integrating your web access logs with IBM Operations Analytics - Log Analysis

Note: Access logs are rotated by default for Apache Tomcat and JBoss. Access Logs are not rotated by default for Apache/IHS.

Configuring log rotation for Apache IHS

The best practice is to configure the web server so that its access logs are rotated and then monitor all rotated access logs.

rotatelogs is a simple program for use in conjunction with Apache's piped logfile feature. It supports rotation based on a time interval or file size (in seconds). The filenames are `<logfile>.nnnn` or `<logfile>.<time>` dependent on the rotatelogs specification. For more information on rotating lots, see http://httpd.apache.org/docs/2.0/programs/rotatelogs.html.

Some users do not use the rotatelogs option because it is configured as a pipe and uses resources (that is, runs as a separate process). Another option users consider is the Unix logrotate tool. The filenames generated are `access.log`, `access.log.1`, `access.log.2`, and so on.

Example

An example of how to configure log rotation for Apache IHS:

1. Update the `httpd.conf` file:
   a. Add the following lines to the `httpd.conf` file:
      ```
      CustomLog "/root/IBM/HTTPServer/bin/rotatelogs
      /root/IBM/HTTPServer/logs/access_log.%Y-%m-%d-%H_%M_%S 86400" scalaAccessLog
      ```
   b. Replace `/root/IBM/HTTPServer` with whatever you are using as HTTPServer home variable
2. Update the log file agent configuration file:
   a. Add the following line to the `WebAccessLogInsightPack-lfadsv.conf` file:
      ```
      LogSources=<log directory>/access_log*
      ```
   b. If more than one file matches the pattern, add the line `FileComparisonMode=CompareByAllMatches` so you will monitor all the files.
3. Update the `WebAccessLogInsightPack-lfadsv.conf` file with the following code:
   ```
   // Matches records for any Log file:
   //

   REGEX AllRecords
   (.*)
   hostname LABEL
   -file ihs-access-log
   RemoteHost DEFAULT
   logpath PRINTF("%s",file)
   text $1
   END
   ```

Web Access Logs Insight Pack References

Important references for your Web Access Logs Insight Pack.

JBoss AS v7 documentation:
https://docs.jboss.org/author/display/AS71/Documentation

Tomcat documentation:
http://tomcat.apache.org/

Apache/IHS documentation:
http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.0.0/com.ibm.websphere.ihs.doc/info/ihs/ihs/welcome_ihs.html

Note: This documentation describes IBM HTTP Server in context of the WebSphere Application Server.

WebSphere Application Server Insight Pack

A WebSphere Application Server Insight Pack is provided with IBM Operations Analytics - Log Analysis.

The Insight Pack includes support for ingesting and performing metadata searches against the following WebSphere Application Server V7 and V8 log files:

- `SystemOut.log`
- `SystemErr.log`
- `trace.log`

This document describes the version of the WebSphere Application Server Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the WebSphere Application Server Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html.

Configuration artifacts

The Insight Pack supports the WebSphere Application Server timestamp format `MM/dd/yy HH:mm:ss:SSS Z`.

The timestamp format can be changed in the Data Sources workspace.

1. Open an existing WebSphere Application Server source type, then click **View Index Configuration**.
2. Select all of the text and copy it to the clipboard.
3. Create a new source type, and click **Edit Index Configuration**.
4. Paste the original index configuration into the editor, then modify the `dateFormats` field of the timestamp entry.
5. Fill in the remaining source type fields and click **OK**. Next, create a new collection that uses the source type you just created.

Note: Data sources are not predefined. A user with administrator privileges must define at least one WebSphere Application Server data source before the application can be used.

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 48. Insight Pack configuration artifacts

	SystemOut.log	**SystemErr.log**	**trace.log**
Splitter	`WASJavaSplitter`	`WASSystemErr-Split`	`WASTrace-Split`
Annotator	`WASJavaAnnotator`	`WASSystemErr-Annotate`	`WASTrace-Annotate`
Source type	`WASSystemOut`	`WASSystemErr`	`WASTrace`
Collection	`WASSystemOut-Collection1`	`WASSystemErr-Collection1`	`WASTrace-Collection1`

Note: The Splitter and Annotator for SystemOut.log use Java. The Splitters and Annotators for SystemErr.log and trace.log use the Annotation Query Language (AQL).

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the *`Log_Analytics_install_dir`*`/IBM-LFA-6.30/config/lo` directory:

- `WASInsightPack-lfawas.conf`: Configuration file for the WAS log file agent.
- `WASInsightPack-lfawas.fmt`: Matches records for the WAS log files.

Splitting and annotation AQL modules

Splitting and annotation are handled by the following Annotation Query Language (AQL) modules.

Table 49. Insight Pack AQL modules

AQL Module	Description
common	Common code module that is used across most WebSphere Application Server log files (for example, to recognize timestamp formats).
annotatorCommon	Common annotations module that is used for all WebSphere Application Server log files.
annotatorSystemOut	Annotator module for the trace.log file.
annotatorSystemErr	Annotator module for SystemErr.log files.
splitterWAS	Splitter module for SystemErr.log files.

Log file formats

The basic formats of the WebSphere Application Server log files are described here as a reference for users.

The basic format of SystemOut.log and SystemErr.log files is:

```
timestamp threadId shortname severity className methodName message
```

where the *className* and *methodName* fields are optional.

The basic format of trace.log files is:

```
timestamp threadId shortname severity className methodName message
parameter_1
parameter_2
parameter_n
```

where the *className*, *methodName*, and *parameter* fields are optional.

In all three log files, stack trace details are written in the following general format:

```
at packageName.exceptionClassName.exceptionMethodName(fileName:lineNumber)
```

For more information about basic and advanced WebSphere Application Server log file formats, see the following topic on the WebSphere Application Server information center:

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/rtrb_readmsglogs.html

Log file splitter

The splitter provided with the Insight Pack is described here as a reference for users.

WebSphere Application Server splitter

The formats of the SystemOut.log, trace.log, and SystemErr.log files are similar enough that they can share a splitter. The splitter recognizes two styles of log record. Most records begin with a timestamp enclosed in brackets and the splitter uses the timestamp to define the beginning and end of each record. However, some stacktraces have a timestamp at the beginning of every line. In these cases the splitter groups the entire stack trace into one log record.

The following example shows a stacktrace with multiple timestamps. The example is split as a single log record.

```
[1/2/13 12:26:12:166 EST] 0000005d SystemErr     R java.lang.NullPointerException
[1/2/13 12:26:12:166 EST] 0000005d SystemErr     R          at com.ibm.blah init
[1/2/13 12:26:12:166 EST] 0000005d SystemErr     R          at com.ibm.blah abcdefg
```

Log file annotations

The annotations that are defined by the log file index configurations are described here.

The following sections describe the fields that are defined in the index configuration file. These fields, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records. Fields are extracted from the fields of a log record or collected from metadata around the log file. Each table gives the names of the fields (these names correspond to fields in the IBM Operations Analytics - Log Analysis Search workspace), descriptions of how the related annotations are made, and the index configuration attributes assigned to the fields.

Log record annotations

The following table lists the index configuration fields that relate to log record annotations. Each field corresponds to part of a SystemOut, SystemErr, or trace log record. The fields are listed in the order in which they appear in a log record.

Table 50. Log record index configuration fields

Field	Description	Attributes
timestamp	The timestamp of the log record, which is located at the beginning of a line and delimited by brackets ([...]).	dataType: DATE retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
threadID	An eight-character alphanumeric (0-9, A-F) thread identifier, which is enclosed by single white space characters, that follow a timestamp.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
shortname	A sequence of characters that represents a short name, which is enclosed by single white space characters, that follow a thread identifier.	dataType: TEXT retrievable: true retrieveByDefault: false sortable: false filterable: false searchable: false
severity	A single-character event type code or severity code (A, C, D, E, F, I, O, R, W, Z, <, >, 1, 2, 3), enclosed by single white space characters, that follow a short name.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
className	If present, a sequence of characters that represents a fully qualified class name (for example, com.ibm.ws.webcontainer.servlet.ServletWrapper) that follows a severity or the string "class=".	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true

Table 50. Log record index configuration fields (continued)

Field	Description	Attributes
methodName	If present, a sequence of characters that represents a method name, which is enclosed by single white space characters, that follow a class name or the string "method=".	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: false
msgclassifier	If present, a defined sequence of characters that ends with a colon (:) and that represents a message identifier (for example, WSVR0605W).	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
message	If present, the text of the system, error, or trace message. This field is annotated only if a msgclassifier field is present.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
javaException	If present, the Java exception names that fit the following pattern: *.*Exception	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true

Stack trace annotations

The following table lists the index configuration fields that relate to log record stack trace annotations.

Table 51. Stack trace index configuration fields

Field	Description	Attributes
exceptionClassName	The class name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
exceptionMethodName	The method name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
fileName	The file name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
lineNumber	The line number in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Table 51. Stack trace index configuration fields (continued)

Field	Description	Attributes
packageName	The package name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Metadata annotations

The following table lists the index configuration fields that relate to metadata annotations.

Table 52. Metadata index configuration fields

Field	Description	Annotation attributes
application	The application name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
hostname	The host name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
logRecord	The entire log record output by the splitter.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
datasourceHostname	The host name specified in the data source.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
middleware	The middleware name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
service	The service name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

WebSphere Application Server data loading best practice

There are different data loading recommendations based on how WebSphere Application Server logging is configured.

To set up the data loading, you need to consider:

WebSphere Application Server configuration
Is WebSphere Application Server configured to produce rolling logs or single logs?

Data loading
Determine if you want to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote WebSphere Application Server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data. In some scenarios, you must use the Data Collector as described in scenario 2.

Logfile agent configuration
If you choose to use the LFA, use the logfile agent configuration files to specify the log files you want to monitor. The `WASInsightpack-lfawas.conf` and `WASInsightPack-lfawas.fmt` files are located in the directory:

```
<HOME>/IBM-LFA-6.30/config/lo
```

The log file scenarios here describe the specific settings for these files.

Scenario 1 - Log file rotation on one WebSphere Application Server server

WebSphere Application Server configuration
WebSphere Application Server is configured for rolling log files. For example, records are written to `SystemErr.log`. When it reaches a defined log file size it is renamed to `SystemErr_13.05.10_05.22.05.log` and a new `SystemErr.log` is created for more data. A similar flow occurs for `SystemOut.log` and `trace.log`.

Data Loading Method

The recommended method for loading data is to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote WebSphere Application Server server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data. Create individual .conf and `.fmt` files that are specific to each log file. The LogSources definition will specify the name of the renamed file. This approach should insure that all log records are processed (no loss of log records) but with the trade off that the log records forwarded to IBM Operations Analytics will be sent only once the rollover occurs. The log forwarding will be one log file behind real-time resulting in some time lag for search results. The amount of time is dependent on the environment, that is, what the defined log size is for rollover and how frequently this occurs based on logging volumes.

Loading logs using the LFA is a CPU bound process. If your system does not meet the minimum requirements you will need to increase the `MaxEventQueueDepth`. On some systems, altering this value may produce a noticeable impact on performance. This will buffer additional LFA events while they are waiting to be processed. The required value for `MaxEventQueueDepth` may vary depending on the size of the rolled log and the number/speed of your CPU's. If you choose not to increase this value, then older events may be replaced on the event queue by newer events and not sent to the IBM Operations Analytics server.

To minimize the chance of data loss due to CPU bottlenecks, and to reduce the latency between when a log record is written to the file and when it is loaded, we recommend that the maximum size of a WebSphere Application Server log be small enough so that you system does not fall behind while processing the logs.

An alternative method is to always monitor the current log file (for example, `SystemErr.log`) and not the renamed log file. This would result in log records being forwarded immediately by the LFA. The trade off is that this configuration may result in log entries not being forwarded if those log entries were written during the LFA polling interval (sleep time) and a rollover occurs. In this case the LFA would start processing the new logfile.

Logfile Agent Configuration - `lfawas.conf` file

You must create additional `.conf` files for each log type that you monitor. For example, if want to monitor the `SystemOut.log` and `trace.log`, then you need a `.conf` file for each log file.

Specify the following parameters to monitor the rotating `SystemErr.log` files:

```
LogSources=<was log directory to monitor>/SystemErr_*.log
FileComparisonMode=CompareByAllMatches
```

Logfile Agent Configuration - `lfawas.fmt` file

You must create additional `.fmt` files for each log type that you monitor. For example, if want to monitor the `SystemOut.log` and `trace.log`, then you need a `.fmt` file for each log file.

Use the following `.fmt` file to specify a fixed `SystemErr.log` name and avoid the need to define multiple logsources because of the rolling log file name changes. This allows a fixed file name in the `logpath`.

```
// Matches records for any Log file:
//

REGEX AllRecords
(.*)
hostname LABEL
-file SystemErr.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

The same pattern can be used to define the `.conf` and `.fmt` files for the other logs:

```
<was log directory to monitor>/SystemOut_*.log   OR
<was log directory to monitor>/trace_*.log</p>
```

Scenario 2 - Collecting Log Files from multiple WAS Servers

WebSphere Application Server Configuration

WebSphere Application Server is configured for single log files (non-rolling) on multiple servers and the server logs are collected for data loading on the IBM Operations Analytics - Log Analysis server.

Data Loading Method

The recommended method for loading data is to use the Data Collector client. Remove the previous log files before creating or copying new versions into the directory from which you will load data. The order of processing logs in this manner is important to handle any split records. When using the Data Collector client you need to set the `flushflag` so that split records can be merged. This is set in the `javaDatacollector.properties` file located in the `<HOME>/utilities/datacollector-client/` directory.

Windows OS Events Insight Pack

The Windows OS Event Insight pack allows users of IBM Operations Analytics - Log Analysis, in conjunction with the Tivoli Log File Agent or logstash, to gather and process Windows OS Events.

This document describes the version of the Windows OS Events Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Windows OS Events Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see http://www.ibm.com/developerworks/servicemanagement/downloads.html.

Two separate data gathering mechanisms are supported, the Tivoli Log File Agent and logstash.

The IBM Operations Analytics - Log Analysis Windows OS Events Insight Pack is built using the IBM Operations Analytics - Log Analysis DSV Toolkit.

For Windows events gathered by the Tivoli Log File Agent (LFA) and logstash the data is configured into a comma separated format, and indexed and annotated for analysis.

The LFA is an agent that provides a configurable log file monitoring capability using regular expressions. The LFA uses the `WINEVENTLOGS` configuration (.conf) file option to monitor events from the Windows event log. The agent monitors a comma-separated list of event logs as shown in the following example:

```
WINEVENTLOGS=System,Security,Application
```

logstash has a supported input module named `eventlog`, http://logstash.net/docs/1.2.2/inputs/eventlog, which pulls events from the Windows Events Logs. The events are then forwarded using the output module available in the logstash Integration Toolkit to the IBM Operations Analytics - Log Analysis EIF Receiver.

Installing the Windows OS Events Insight Pack

If you are using IBM Operations Analytics - Log Analysis 1.2 or later, the Windows OS Events Insight Pack is installed by default, and therefore does not need to be installed separately.

About this task

The Windows OS Events Insight Pack is installed using the `pkg_mgmt` utility.

Procedure

1. Upload the Windows OS Events Insight Pack archive file, `WindowsOSEventsInsightPack_<version>.zip`, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Windows OS Events Insight Pack with the `pkg_mgmt.sh` command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install
   <path>/WindowsOSEventsInsightPack_<version>.zip
   ```

   Where <path> is the path where you saved the Windows OS Events Insight Pack.

Uninstalling the Windows OS Events Insight Pack

Instructions on how to uninstall the Windows OS Events Insight Pack.

About this task

The Windows OS Events Insight Pack is installed and uninstalled using the `pkg_mgmt` utility.

Procedure

1. Use the **`pkg_mgmt.sh`** command to determine the location of the insight pack:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh
   ```

2. Uninstall the Windows OS Events Insight Pack with the **`pkg_mgmt.sh`** command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -uninstall
   <path>/WindowsOSEventsInsightPack_<version>
   ```

   Where <path> is the path listed by the command in step 1 showing the location of Windows OS Events Insight Pack.

Performance considerations

Ensure that you consider these limitations when using the Windows OS Events Insight Pack:

Files with long log records require adjustments to the Java stack size so that they can be ingested. This adjustment is made by adding or updating the line `-Xss<stacksize>` in the `jvm.options` file located in the `<HOME>/wlp/usr/servers/Unity` directory. `<stacksize>` is the desired stack size. By default, lines of approximately 1000 characters are supported. To support lines up to 10,000 characters, the stack size must be set to 6144 kb. To support lines up to 9,000 characters, the stack size must be set to 5120 kb. An example line is:

```
-Xss6144k
```

If you add or update the value of this parameter within the `jvm.options` file, you must restart the IBM Operations Analytics - Log Analysis system. For more information on how to restart IBM Operations Analytics - Log Analysis, see the *unity command* topic in the documentation.

Integrating the Windows OS Events Insight Pack with the Log File Agent

Configuring a Log File Agent instance on Windows allows Windows OS events to be forwarded to IBM Operations Analytics - Log Analysis.

Before you begin

Ensure that the Tivoli Log File Agent (LFA) is installed on the Windows server that is being monitored. For more information on installing the Tivoli LFA, see the "Tivoli Log File Agent User's Guide" in the IBM Tivoli Monitoring Knowledge Center.

Ensure that the Windows Server can communicate with the IBM Operations Analytics - Log Analysis server. Communication is directed to the EIF receiver port on the IBM Operations Analytics - Log Analysis server (default 5529). Ensure that any firewall restrictions are lifted.

About this task

The steps in this task outline how to use the LFA to gather and push Windows OS events to IBM Operations Analytics - Log Analysis server. The LFA can be configured to send Windows OS Events to the EIF Receiver that is deployed with IBM Operations Analytics - Log Analysis. For more details on configuring the EIF Receiver on IBM Operations Analytics - Log Analysis, see section "Configuring the EIF Receiver" in the IBM Operations Analytics - Log Analysis Knowledge Center.

Procedure

1. On the IBM Operations Analytics - Log Analysis server, copy the LFA `.conf` and `.fmt` files to the target Windows Server.

   The `.conf` and `.fmt` files are in the directory that Windows OS Events Insight Pack is installed in.

   The location of the Windows OS Events Insight Pack can be determined by using the `pkg_mgmt.sh` command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -list
   ```
2. On the target Windows Server place, both files in a directory accessible to the installation of the Tivoli LFA.
3. Edit the `lfaWinEvt.conf` file.
   a. Update the **`ServerLocation`** to the host name or IP address of the IBM Operations Analytics - Log Analysis server

   b. Update the **`ServerPort`** to the configured value on the IBM Operations Analytics - Log Analysis server.

      The default port is 5529.

      ```
      # Our EIF receiver host and port.
      # Only needed when sending events directly to OMNIbus or TEC via EIF.
      # That is configured through either the Manage Tivoli Enterprise Monitoring
      # Services GUI or the
      # "itmcmd config -A lo" command.
      ServerLocation=unityserver.ibm.com
      ServerPort=5529
      ```

      For more information on configuring the EIF Receiver on IBM Operations Analytics - Log Analysis, see section "Configuring the EIF Receiver" in the IBM Operations Analytics - Log Analysis Knowledge Center.

   The `lfaWinEvt.fmt` file formats the Windows OS events that are read by the Tivoli LFA into a CSV format for ingestion by the Windows OS Events Insight Pack.
4. The only value within this `.fmt` file you are recommended to edit is **`logpath`**. This string must match that of the configured data source on the IBM Operations Analytics - Log Analysis server.

   By default, the value of the host name is the value that is returned by executing the DOS command **`hostname`** from the command line. This string must be used as the host name value when configuring the data source on the IBM Operations Analytics - Log Analysis server.
5. Launch the **Manage Tivoli Enterprise Monitoring service** application on the Windows Server.
6. Select the **Tivoli Log File Agent** template and select **Actions** > **Configure** using defaults.
7. Enter a unique instance name when prompted.

Note: There is a limit on the length of the instance names. The internal identification of an LFA instance by ITM libraries restricts the length to 32 chars in total.

8. In the **Log File Adapter Configuration** tab, enter the location of the `.conf` and `.fmt` files, and set the **Send ITM Event** option to **No**.

   The LFA instance will now be configured and can be started from the **Manage Tivoli Enterprise Monitoring service**.

   Once started, it is possible to troubleshoot the LFA instance by:

   a. Select and right-click the LFA instance in the **Manage Tivoli Enterprise Monitoring service** dialog.

   b. Click **Advanced** > **View Trace File**.

   The `$UNITY_HOME/logs/UnityEifReceiver.log` file on IBM Operations Analytics - Log Analysis server can now be used to observe events being received from the LFA by IBM Operations Analytics - Log Analysis.

   For more information on logging the `UnityEifReceiver`, see section "Enabling console logging and changing the log level for the EIF receiver" in the IBM Operations Analytics - Log Analysis Knowledge Center.

   Note: When configuring the LFA, ensure that the **No TEMS** option is selected. For more details on configuring this option, see the known issue "Log File Agent fails to post events" in the IBM Operations Analytics - Log Analysis Knowledge Center.

Integrating the Windows OS Events Insight Pack with logstash

Configuring logstash on Windows allows Windows OS events to be forwarded to IBM Operations Analytics - Log Analysis.

Before you begin

Ensure that the logstash Integration Toolkit has been deployed on the Windows Server being monitored. For more details on configuring logstash on a Windows Server see the section *logstash Integration Toolkit* in the IBM Operations Analytics - Log Analysis Knowledge Center.

Ensure that the Windows Server can communicate with the IBM Operations Analytics - Log Analysis server. Communication will be directed to the EIF receiver port on the IBM Operations Analytics - Log Analysis server (default 5529). Ensure that any firewall restrictions are lifted.

About this task

The steps in this task outline how to configure logstash to send Windows OS Events to the EIF Receiver that is deployed with IBM Operations Analytics - Log Analysis. For more details on configuring the EIF Receiver on IBM Operations Analytics - Log Analysis see the section *Configuring the EIF Receiver* in the IBM Operations Analytics - Log Analysis Knowledge Center.

Procedure

1. On the target Windows Server ensure that logstash is not running. For information on how to stop logstash, see the section *Stopping logstash* in the IBM Operations Analytics - Log Analysis Knowledge Center.
2. Make a backup of the `<logstash Location>\lstoolkit\logstash\config\logstash-scala.conf` file.

3. On the IBM Operations Analytics - Log Analysis server, copy the `logstash-scala.conf` file to the target Windows Server.

   The `logstash-scala.conf` file is located in the directory that Windows OS Events Insight Pack is installed in.

   The location of the Windows OS Events Insight Pack can be determined by using the `pkg_mgmt.sh` command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -list
   ```
4. On the Windows Server place the `logstash-scala.conf` file in the location `<logstash Location>\lstoolkit\logstash\config`. This overwrites the existing version.
5. On the Windows server ensure that logstash eif output module is configured to send data to the IBM Operations Analytics - Log Analysis server.
6. On the Windows server check that the values of the output module in the new `logstash-scala.conf` file match that of the backed up copy. This check is needed if you have specified a non-standard location for the eif output module.
7. On the target Windows Server start logstash. For information on how to start logstash, see the section *Starting logstash* in the IBM Operations Analytics - Log Analysis Knowledge Center.

Windows OS event format generated by the Tivoli Log File Agent

Windows OS events are formatted by the Tivoli Log File Agent into a csv format.

The value of the log source's logpath must match that specified for the logpath in the `.fmt` file deployed on the LFA on the Windows server.

The Windows OS Events Insight pack has been built using the IBM Operations Analytics - Log Analysis DSV toolkit. Events are formatted by the Tivoli Log File Agent into a csv format with the following columns.

Table 53. Log file format

Number	Column Name	Description
1	`EventCategory`	Describes the sussystem of event, for example, EventLog:Application or EventLog:Security
2	`Timetsamp`	Time of event
3	`Level`	Information, Warning, Error etc
4	`User`	If a user name is associated with the event
5	`EventSource`	Source of event
6	`Keywords`	Events may have keywords associated upon generation.
7	`EventID`	Event ID
8	`Description`	Text description of event

Windows OS event format generated by logstash

The basic format of the Windows Event Log generated by logstash is described here as a reference for users.

The Windows OS Events Insight pack has been built using the IBM Operations Analytics - Log Analysis DSV toolkit. Events are formatted by logstash into a csv format with the following columns.

Table 54. Log file format

Number	Column Name	Description
1	`EventLog`	Describes the subsystem of event, for example Application or Security
2	`Timetsamp`	Time of event
3	`Level`	Information, Warning, Error etc
4	`User`	If a user name is associated with the event
5	`EventSource`	Source of event
6	`EventID`	Event ID
7	`Description`	Text description of event
8	`Hostname`	Hostname of the Windows machine
9	`EventRecordNumber`	Unique event ID
10	`Category`	Numeric category

Windows OS Events Insight Pack App

A single sample app is provided with the Windows OS Events Insight Pack. This app queries the data gathered by the application and generates four charts.

The four charts generated by the application are:

- Event Level counts per hour over the past 24 hours
- Event Log counts per hour over the past 24 hours
- Event Source Counts per hour over the past 24 hours
- Event Level per Event Source over the past 24 hours

By default, the app will query a logsource named **`WindowsOSEventsLFA`**.

If you want to run the chart on another logsource based on the **`WindowsOSEventsLFA`** source type:

1. Open the file: `<HOME>/AppFramework/Apps/WindowsOSEventsInsightPack_`*`<version>`*`/WindowsEvents.app`.
2. Update the value of the logsource name from **`WindowsOSEventsLFA`** to whatever logsource name is required.

   Note: All configured logsource names can be seen on the IBM Operations Analytics - Log Analysis Administrative settings UI under the **Log Sources** tab.

Limiting the flow of events to the Windows OS Event Log Insight Pack

An optional set of steps the purpose of which is to limit the events flowing to the Windows OS Event Log Insight Pack.

About this task

Windows OS generates a large number Information level logs that users may not wish to track. The `.fmt` file can be edited to limit what is monitored.

Procedure

1. On the Windows server edit the fmt file (See steps above for configuring the tivoli LFA) as follows. For more information about how to configure the IBM Tivoli Monitoring Log File Agent, see "Integrating the Windows OS Events Insight Pack with the Log File Agent" on page 171.

   Update the `.fmt` file from:

   ```
   // Matches records for any Log file and converst to csv format:
   //
   REGEX AllRecords
   ^([A-Z][a-z]{2} [0-9]{1,2} [0-9]{1,2}:[0-9]{2}:[0-9]{2} [0-9]{4})
   [0-9] (\S+) (\S+) (\S+) (\S+) ([0-9]+) (.*)
   hostname LABEL
   -file FILENAME
   RemoteHost DEFAULT
   logpath "WindowsOSEventsLFA"
   text PRINTF("%s,%s,%s,%s,%s,%s,%s,%s",file,$2,$3,$4,$5,$6,$7,$8)
   END
   ```

   To:

   ```
   // Matches records for any Log file and converst to csv format:
   //
   REGEX AllRecords
   ^([A-Z][a-z]{2} [0-9]{1,2} [0-9]{1,2}:[0-9]{2}:[0-9]{2} [0-9]{4})
   [0-9] (Warning|Error|Critical) (\S+) (\S+) (\S+) ([0-9]+) (.*)
   hostname LABEL
   -file FILENAME
   RemoteHost DEFAULT
   logpath "WindowsOSEventsLFA"
   text PRINTF("%s,%s,%s,%s,%s,%s,%s,%s",file,$2,$3,$4,$5,$6,$7,$8)
   END
   ```

   This will limit the events being sent to IBM Operations Analytics - Log Analysis to those of type Warning or Error or Critical. No 'Information' events will be sent to IBM Operations Analytics - Log Analysis.
2. Restart the LFA instance using the **Manage Tivoli Enterprise Monitoring service** application

IBM MB Insight Pack

The IBM MB Insight Pack facilitates data ingestion and metadata searches of IBM Message Broker and IBM Integration Bus logs files in IBM Operations Analytics - Log Analysis to enable faster problem identification.

The IBM MB Insight Pack supports searching and indexing of the following:

- `console` logs
  - UNIX/Linux: /var/mqsi/components/*broker_name*/*execution_group_uuid*\console.txt
  - Windows: *workpath*\components\broker_name\execution_group_uuid\console.txt

    where *workpath* is the Message Broker defined working directory.

    The `console.txt` log file includes information about the `Broker ID`, messages from the Message Broker, log record time stamps, Universally Unique ID for IBM Integration Bus objects, and Java exceptions.

- Syslog
  - UNIX/Linux: The syslog is the local error log. The configuration of your UNIX/Linux system determines where the syslog messages are sent.

The IBM MB Insight Pack can be installed with IBM Operations Analytics - Log Analysis 1.3.0.0 or higher.

Supported IBM Integration Bus versions

The IBM MB Insight Pack supports the following IBM Integration Bus and IBM WebSphere Message broker versions. IBM Integration Bus was known as IBM WebSphere Message broker until version 8.0.

- IBM WebSphere Message broker 8.0
- IBM Integration Bus 9.0
- IBM Integration Bus 10.0

Installing the IBM MB Insight Pack

To ingest data and perform searches of the IBM Message Broker logs files in IBM Operations Analytics - Log Analysis, you must install the IBM MB Insight Pack.

Before you begin

You must install IBM Operations Analytics - Log Analysis 1.3.0.0 or higher.

About this task

The IBM MB Insight Pack is installed by using the pkg_mgmt utility.

Procedure

1. Upload the IBM MB Insight Pack archive file, IBMMBInsightPack_*<version>*.zip, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the IBM MB Insight Pack by using the pkg_mgmt.sh command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install
   <path>/IBMMBInsightPack_<version>.zip
   ```
   Where *<path>* is the path where you saved the IBM MB Insight Pack.
3. (Optional) If you are using the Log File Agent to load the data into IBM Operations Analytics - Log Analysis, deploy the log file agent configuration files with the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa
   <path>/IBMMBInsightPack_<version>.zip
   ```
   Where *<path>* is the path where you saved the IBM MB Insight Pack.

IBM MB Insight Pack configuration artifacts:

During IBM MB Insight Pack installation, a number of artifacts are created.

The following artifacts are created when the IBM MB Insight Pack is installed:

- MsgBrokerSplitter, MsgBrokerSyslog_Splitter : Splitter file set.
- MsgBrokerAnnoator, MsgBrokerSyslog_Annoator : Annotator file set.
- Message Broker console log, Message Broker Syslog : Source type.
- Dash.app : Sample Dashboard for Syslog

Configuring the IBM MB Insight Pack

To index and search the IBM Message Broker logs files, you must configure the IBM MB Insight Pack.

Procedure

1. The default timestamp format that is supported by the `Message Broker console log` is `yyyy-MM-dd HH:mm:ss.SSS`. If the `console log` timestamp format is different, you must create a new source type. To create a new source type, complete the following steps:
   a. In the Administrative settings, open the **Data Types** tab.
   b. Select **Create New Source Type**.
   c. Enter a name for the new source type.
   d. Select **Enable Splitter**>**Fileset**. In the **Fileset** field, select **MsgBrokerSplitter**
   e. Select **Enable Annotator**>**Fileset**. In the **Fileset** field, select **MsgBrokerAnnotator**.
   f. Copy the `indexconfig` from the Message Broker console log source type and select **Edit Index Configuration** in the new source type. Paste the `indexconfig` in to the input box.
   g. Modify the **`timestamp`** parameter. For example:

```
"timestamp": {
        "dataType": "DATE",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "source": {
          "paths": [
            "metadata.timestamp"
          ],
          "dateFormats": [
            "yyyy-MM-dd HH:mm:ss.SSS"
          ]
        }
      },
```

   h. Click **OK** to complete and save the new source type.
2. The default timestamp format that is supported by the `Message Broker syslog` is `MMM dd HH:mm:ss yyyy`. If the `syslog` timestamp format is different, you must create a new source type. If the year is not included in the syslog, the IBM MB Insight Pack appends it. To create a new source type, complete the following steps:
   a. In the Administrative settings, open the **Data Types** tab.
   b. Select **Create New Source Type**.
   c. Enter a name for the new source type.
   d. Select **Enable Splitter**>**Fileset**. In the **Fileset** field, select **MsgBrokerSyslog_Splitter**
   e. Select **Enable Annotator**>**Fileset**. In the **Fileset** field, select **MsgBrokerSyslog_Annotator**.
   f. Copy the `indexconfig` from the Message Broker syslog source type and select **Edit Index Configuration** in the new source type. Paste the `indexconfig` in to the input box.
   g. Modify the **`timestamp`** parameter. For example:

```
"timestamp": {
        "dataType": "DATE",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "source": {
          "paths": [
            "metadata.timestamp"
          ],
          "dateFormats": [
            "MMM dd HH:mm:ss yyyy"
          ]
        }
      },
```

h. Click **OK** to complete and save the new source type.

3. To index `console.txt` logs, complete the following steps:
   a. Navigate to the IBM Operations Analytics - Log Analysis settings workspace.
   b. Create a log source for the log file to be monitored. The source type must be Message Broker console log or the source type that is created in step 1.
4. To index `syslog`, complete the following steps:
   a. Navigate to the IBM Operations Analytics - Log Analysis settings workspace.
   b. Create a log source for the log file to be monitored. The source type must be Message Broker Syslog or the source type that is created in step 2.

Log File Agent configuration

You can use the IBM Tivoli Log File Agent to load data into the IBM Operations Analytics - Log Analysis.

The following Log File Agent configuration files will be installed in `<HOME>/IBM-LFA-6.30/config/lo` directory when **`pkg_mgmt`** is run with the **`-deploylfa`** option.

- `lfamb.conf` - Configuration for the web access log file agent.
- `lfamb.fmt` - Matches records for the web access log files.

IBM MB Insight Pack Log splitter rules

Splitting describes how IBM Operations Analytics - Log Analysis separates physical log file records in to logical records by using a logical boundary such as time stamp or a new line.

The IBM MB Insight Pack splits log records on new line boundaries. In the following `syslog` example each line is considered a new log record.

```
Mar 22 14:23:48 ldp3147 IIB[14903]: [ID 702911 user.info] IBM Integration Bus v9002 (EAIESB1PRD1BK
Mar 22 14:23:49 ldp3147 IIB[14692]: [ID 702911 user.info] IBM Integration Bus v9002 (EAIESB2PRD1BK
```

For console logs, the IBM MB Insight Pack groups log records with multiple lines. For example, the following log record has multiple lines that belong to the exception in the first line. Therefore, the logs are grouped as one with a common time stamp that corresponds to the first record.

```
2014-12-10 22:04:48.557     18 java.lang.NullPointerException
2014-12-10 22:04:48.557     18  at au.net.api.integration.iib.common.Config.<clinit>(Config.java:2
2014-12-10 22:04:48.557     18  at java.lang.J9VMInternals.initializeImpl(Native Method)
2014-12-10 22:04:48.573     18  at java.lang.J9VMInternals.initialize(J9VMInternals.java:237)
```

IBM MB Insight Pack Log annotation rules

After the log records are split, the logical records are sent to the annotation engine. The engine uses rules to extract important pieces of information that are sent to the indexing engine.

IBM MB Insight Pack console log annotator

The IBM MB Insight Pack annotates the console log records based on the following fields:

- timestamp
- message
- UUID
- broker_name
- thread_id
- logRecord

IBM MB Insight Pack syslog annotator

The IBM MB Insight Pack annotates the syslog records based on the following fields:

- timestamp
- SystemName
- Product
- Process
- severity
- version
- Broker
- ExecutionGroup
- ThreadID
- MessageGroup
- MessageID
- MessageText
- logRecord

IBM MB Insight Pack index configuration

To control how IBM Operations Analytics - Log Analysis indexes records from a log file, you can create indexing settings for your content Insight Pack.

The fields that are defined in the index configuration file, or annotations, are displayed in the IBM Operations Analytics - Log Analysis search workspace, and can be used to filter or search the log records.

Fields are extracted from the fields of a log record or collected from metadata around the log file.

Table 55. Log index configuration for console logs

Field	Description	Attributes

timestamp	The time that the log record was written by the broker or execution group The format is:
	yyyy-MM-dd HH:mm:ss.SSS
		dataType = DATE
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = metadata

message	The message that is written by the broker or execution group.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

UUID	The Universally Unique ID of the execution group.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true
		source = annotations

broker_name	The broker name of the parent process.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

thread_id	The ID of the thread that initiated the process.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true
		source = annotations

logRecord	The entire log record.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = metadata

Table 56. Log index configuration for message broker syslog

Field	Description	Attributes

timestamp	The time that the log record was written. The format is:
	MMM dd HH:mm:ss yyyy
		dataType = DATE
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = metadata

Table 56. Log index configuration for message broker syslog (continued)

Field	Description	Attributes

SystemName	The name of the system running the IBM MB Insight Pack.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

Product	The product that generated the log record.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotations

Process	The process ID	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotations

version	The IBM Integration Bus version. For example, IBM Integration Bus v9002	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Broker	The broker name.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

ExecutionGroup	The execution group name.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

ThreadID	ID of the thread that initiated the process.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

MessageGroup	The message group of the log record. For example, Msg 1/3, Msg 2/3.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = false
		searchable = true
		source = annotations

Table 56. Log index configuration for message broker syslog (continued)

Field	Description	Attributes

MessageID	The ID of the message that is sent in the log record. For example, BIP31321, BIP2060W.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

severity	The severity of the event that caused the message. It is extracted from the last digit of the msgID. The severity levels are:
	• Info
	• Warning
	• Error
	• Severe
		dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

MessageText	The message that is written by the broker or execution group to the logs.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

logRecord	The entire log record	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

IBM MB Insight Pack dashboard

You can use the IBM MB Insight Pack dashboard example app to visualize data from Message Broker logs.

Use the IBM MB Insight Pack dashboard to find Message Broker-related problems in the shortest time and drilldown into the problem to identify the root cause.

The IBM MB Insight Pack app contains the following charts:

BIP2001I:Brk start, BIP2228E: Brk Abend
Displays broker start and end events.

BIP2208I: EG start, BIP2204I: EG Stop
Displays execution group start and end events.

HTTP Listener Start
Displays HTTP listener events.

Error, Warning and Info messages
Displays the distribution of error, warning, and information logs over time.

Configuring the IBM MB Insight Pack dashboard:

To use the IBM MB Insight Pack dashboard, you must configure the IBM MB Insight Pack dashboard app.

Procedure

To configure the IBM MB Insight Pack dashboard app, complete the following steps

1. Copy the Message Broker Dash.app in the <HOME>/IBM/LogAnalysis/AppFramework/Apps/IBMMBInsightPack_*<version>* directory. Save the copied Dash.app with a new, unique name. Save the copied Dash.app with a new, unique name. For example, MBDashboard1.app.
2. The dashboard name must be unique. To change the dashboard name, modify the **name** parameter in the MBDashboard1.app.
3. The default log source that is used by the IBM MB Insight Pack dashboard app is MBSyslog. You must change the log source for each chart. To change the log source, modify the **logsources** parameter. For example,

```
"logsources": [
                {
                  "name": "\/MBSyslog",
                  "type": "logSource"
                }
             ],
```

4. To change the time interval for each chart, modify the **filters** parameter. For example,

```
"filter": {
                 "range": {
                    "timestamp": {
                       "to": "27\/05\/2015 11:22:46.191 +0530, 11:22 AM",
                       "dateFormat": "dd\/MM\/yyyy HH:mm:ss.SSS Z",
                       "from": "27\/05\/2014 11:22:46.191 +0530, 11:22 AM"
                    }
                 }
              }
```

   To display the data for the past 2 years in the dashboard, modify the **timefilters** parameter as follows:

```
"filter": {
                 "timefilters": {
           "granularity": "year",
           "lastnum": 2,
           "type": "relative"
                 }
              }
```

5. Save the changes. To see the new dashboard, refresh the **Search Dashboards** pane on IBM Operations Analytics - Log Analysis UI.

IBM MB Insight Pack References

Important references for your IBM MB Insight Pack.

IBM Integration Bus Version 10.0 documentation:
http://www-01.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm

IBM MQ Insight Pack

An IBM MQ Insight Pack is provided with IBM Operations Analytics - Log Analysis.

The IBM MQ Insight Pack provides the capability to index and search against IBM MQ logs files in IBM Operations Analytics - Log Analysis to enable faster problem identification in MQ-related problems.

The IBM MQ Insight Pack supports logs from the following:

- IBM MQ 8.0
- MQ Series 7.5
- MQ Series 7.1
- MQ Series 7.0.1

The IBM MQ Insight Pack supports searching and indexing of the following:

AMQERR logs generated per Queue Manager

UNIX: At /var/mqm/qmgrs/<*queueManagername*>/errors

Windows: At c:\Program Files\IBM\WebSphere MQ\qmgrs\<*queueManagername*>\errors

AMQERR.log that is common for all queue managers.

UNIX: At /var/mqm/errors

Windows: At c:\Program Files\IBM\WebSphere MQ\qmgrs\errors

The error log files, named AMQERR01.LOG, AMQERR02.LOG, and AMQERR03.LOG, contain information that is logged by queue managers that can be used for problem debugging.

MQ FFST files generated by Queue Manager

UNIX: At /var/mqm/errors

Windows: At c:\ProgramData\IBM\MQ\errors

The MQ FFST files are named AMQnnnnn.mm.FDC where nnnn is the ID of the process reporting the error, and mm is the sequence number.

Note: The c:\ProgramData directory is hidden by default. To change the directory, enter **%PROGRAMDATA%** in the Windows Explorer location bar to browse it, or configure Windows Explorer to show hidden files and directories.

Installing the IBM MQ Insight Pack

To index and search IBM MQ logs, you must install the IBM MQ Insight Pack.

About this task

The IBM MQ Insight Pack is installed by using the pkg_mgmt utility that is shipped with IBM Operations Analytics - Log Analysis.

Procedure

1. Upload the IBM MQ Insight Pack archive file, IBMMQInsightPack_<*version*>.zip to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install IBM MQ Insight Pack by using the pkg_mgmt.sh command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install
   <path>/IBMMQInsightPack_<version>.zip
   ```

   Where:

   <HOME> is the directory where IBM Operations Analytics - Log Analysis is installed.

   <*path*> is the path where you saved the IBMMQInsightPack_<*version*>.zip.
3. (Optional) If you are using the Log File Agent to load the data into IBM Operations Analytics - Log Analysis, deploy the log file agent configuration files with the following command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa
<path>/IBMMQInsightPack_<version>.zip
```

Where:

<HOME> is the directory where IBM Operations Analytics - Log Analysis is installed.

<path> is the path where you saved the IBMMQInsightPack_*<version>*.zip.

IBM MQ Insight Pack configuration artifacts

The IBM MQ Insight Pack created artifacts when it is installed.

The following artifacts are created when the IBM MQ Insight Pack is installed:

WMQAMQERRSplitter, WMQFDCSplitter
Splitter file set.

WMQAMQERRAnnoator, WMQFDCAnnoator
Annotator file set

WMQ_AMQERR, WMQ_FDC
Source type

WMQAMQERR_C1, WMQFDC_C1
Collection

Configuring the IBM MQ Insight Pack

To index and search IBM MQ logs, you must configure the IBM MQ Insight Pack.

Procedure

1. (Optional) To modify the default timestamp format for AMQERR files:
   a. The default timestamp format supported by the source type installed with the IBM MQ Insight Pack is MM/dd/yyyy HH:mm:ss. If the timestamp format in the AMQERR file is different, create a new source type by copying and modifying the contents of the WMQ_AMQERR source type. To change the timestamp format, edit the **dateFormats** parameter. For example:

```
"timestamp": {
       "dataType": "DATE",
       "retrievable": true,
       "retrieveByDefault": true,
       "sortable": true,
       "filterable": true,
       "searchable": true,
       "source": {
         "paths": [
           "metadata.timestamp"
         ],
         "dateFormats": [
           "yyyy-MM-dd HH:mm:ss.SSS"
         ]
       }
     },
```

   b. Save the changes to the timestamp format.
2. (Optional) To modify the default timestamp format for FDC files:
   a. The default timestamp format supported by WMQ_FDC source type installed with the IBM MQ Insight Pack is EEE MMMMM dd yyyy HH:mm:ss zzzz. If the timestamp format in the FDC file is different, create a new source type by copying and modifying the contents of the WMQ_FDC source type. To change the timestamp format, edit the **dateFormats** parameter. For example:

```
"timestamp": {
    "dataType": "DATE",
    "retrievable": true,
    "retrieveByDefault": true,
    "sortable": true,
    "filterable": true,
    "searchable": true,
    "source": {
     "paths": [
      "metadata.timestamp"
    ],
    "dateFormats": [
     "yyyy-MM-dd HH:mm:ss.SSS"
    ]
   }
  },
```

b. Save the changes to the timestamp format.

For information on valid timestamp formats, see https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html

3. To index `AMQER` logs:
   a. Navigate to the IBM Operations Analytics - Log Analysis settings workspace.
   b. Create a log source for the log file to be monitored. The source type must be `WMQ_AMQERR` or the source type that is created in step 1a, if different.
4. To index `FDC` files:
   a. Navigate to the IBM Operations Analytics - Log Analysis settings workspace.
   b. Create a log source for the log file to be monitored. The source type must be `WMQ_FDC` or the source type that is created in step 2a, if different.
5. (Optional) To configure the IBM Tivoli Monitoring Log File Agent to stream `MQ AMQERR` logs to the IBM Operations Analytics - Log Analysis server, see the *Configuring IBM Tivoli Monitoring Log File Agents* topic in the *IBM Tivoli Monitoring Log File Agent configuration scenarios* section of *Loading and streaming data* in the *Configuring IBM Operations Analytics - Log Analysis* guide.

IBM MQ Insight Pack splitter rules

Splitting describes how IBM Operations Analytics - Log Analysis separates physical log file records in to logical records by using a logical boundary such as time stamp or a new line.

The IBM MQ Insight Pack splits log records on timestamps. For example, in the following log record the critical utility task manager has started the `LOGGER-IO` task. This task was run once. The `ACTION` is *None*.

```
4/2/2015 12:52:16 - Process(8456.3) User(user1) Program(amqzmuc0.exe)
                      Host(HOST1) Installation(Installation5)
                      VRMF(7.1.0.0) QMgr(QM_71)

AMQ5051: The queue manager task 'LOGGER-IO' has started.
EXPLANATION:
The critical utility task manager has started the LOGGER-IO task. This task has now started 1 time
ACTION:
None.
```

The IBM MQ Insight Pack splits `FDC` records on based on the timestamp.

IBM MQ Insight Pack Annotation rules and index configuration

After the log records are split, the logical records are sent to the annotation engine. The engine uses rules to extract important pieces of information that are sent to the indexing engine.

The IBM MQ Insight Pack annotates the log records based on the following fields:

Table 57. Field description and index configuration for `WMQ_AMQERR` source type

Field	Description	Attributes

`timestamp`	Time that IBM MQ generated the log record.	`dataType = DATE`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = TRUE`
		`searchable = true`
		`source = metadata`

`Process`	The Queue manager process generating the log record.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = FALSE`
		`filterable = false`
		`searchable = true`
		`source = annotations`

`User`	The Queue manager user name.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = false`
		`filterable = true`
		`searchable = true`
		`source = annotations`

`MessageID`	AMQXXXX logged by the queue manager.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = true`
		`searchable = true`
		`source = annotations`

`MessageText`	The message followed by AMQXXXX in the log record.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = false`
		`filterable = false`
		`searchable = true`
		`source = annotations`

`SystemName`	The host running the Queue manager.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = true`
		`searchable = true`
		`source = annotations`

`ProgName`	The IBM MQ internal program generating the log record	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = true`
		`searchable = true`
		`source = annotations`

Table 57. Field description and index configuration for `WMQ_AMQERR` source type (continued)

Field	Description	Attributes

`Qmgr`	The name of the Queue manager generating the log record.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = true`
		`searchable = true`
		`source = annotations`

`VRMF`	The IBM MQ version	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = false`
		`filterable = false`
		`searchable = true`
		`source = annotations`

`logRecord`	Entire log record	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = false`
		`filterable = false`
		`searchable = true`
		`source = metadata`

Table 58. Field description and index configuration for `WMQ_AMQERR` source type

Field	Description	Attributes

`timestamp`	The time in `FDC`.
	Format: `EEE MMMMM dd yyyy HH:mm:ss zzzz`
		`dataType = DATE`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = TRUE`
		`searchable = true`
		`source = metadata`

`logRecord`	Entire log record.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = false`
		`filterable = false`
		`searchable = true`
		`source = metadata`

`SystemName`	The host running the Queue manager.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = true`
		`filterable = true`
		`searchable = true`
		`source = annotations`

`VRMF`	The IBM MQ version.	`dataType = TEXT`
		`retrievable = true`
		`retrieveByDefault = true`
		`sortable = false`
		`filterable = false`
		`searchable = true`
		`source = annotations`

Table 58. Field description and index configuration for `WMQ_AMQERR` *source type (continued)*

Field	Description	Attributes

ProbeID	The probe ID of the FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

ApplicantName	The name of the application generating FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Component	The IBM MQ component generating FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

SCCS	Source Code Control System.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

LineNumber	The source code line number.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

BuildLevel	The IBM MQ build level.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

User	The Queue manager user name.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = true
		searchable = true
		source = annotations

ProcessName	The process name generating FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Table 58. Field description and index configuration for WMQ_AMQERR source type (continued)

Field	Description	Attributes

Process	The process ID generating FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Thread	The IBM MQ thread generating FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Qmgr	The name of the Queue manager generating the log record.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

MajorErrorCode	The major error code reported in FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Operating System	The operating system running Qmgr.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

MessageID	AMQXXXX logged by the queue manager.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = true
		filterable = true
		searchable = true
		source = annotations

MessageText	The message followed by AMQXXXX in the log record.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Comment1	The comment from FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

Table 58. Field description and index configuration for `WMQ_AMQERR` source type (continued)

Field	Description	Attributes

`Comment2`	The comment from FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

`MQMFunctionStack`	The function stack generating FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

`TraceHistory`	The trace history from FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

`EnvVariables`	The `EnvVariables` reported in FDC.	dataType = TEXT
		retrievable = true
		retrieveByDefault = true
		sortable = false
		filterable = false
		searchable = true
		source = annotations

IBM MQ Insight Pack dashboard

You can use the IBM MQ Insight Pack dashboard example app to visualize data from MQ logs.

Use the IBM MQ Insight Pack dashboard to find MQ-related problems in the shortest time, and drill down into the problem to identify the root cause.

The IBM MQ Insight Pack app contains the following charts:

Which are top `MessageIDs`
Plots a pie chart of `MessageIDs` from the selected log records.

Which `MessageID` occurred the most and when?
Plots a bar graph that shows how `MessageIDs` are distributed over a selected time frame.

Which `Qmgr` logged these `MessageIDs`?
Plots a bubble chart that shows which `Qmgr` logged most of the `MessageIDs`.

Which `MQ` program logged these `MessageIDs`?
Plots a bar graph that shows which `MQ` program logged the `MessageIDs`.

Configuring the IBM MQ Insight Pack dashboard app

To use the IBM MQ Insight Pack dashboard, you must configure the IBM MQ Insight Pack dashboard app.

Procedure

To configure and customize the IBM MQ Insight Pack dashboard app, complete the following steps.

1. Copy the `MQDash.app` from the `<HOME>/IBM/LogAnalysis/AppFramework/Apps/IBMMQInsightPack_<version>` directory to a new directory. For example `MQDashboard1.app`.
2. The dashboard name must be unique. To change the dashboard name, modify the **name** parameter in the `MQDashboard1.app`.
3. The default log source that is used by the IBM MQ Insight Pack dashboard app is `MQ_80`. You must change the log source for each chart. To change the log source, modify the **logsources** parameter. For example:

   ```
   "logsources":[
   {
   "type":"logSource",
   "name":"\/MQ_80.log"
   }
   ],
   ```
4. The default time interval is the last year, and the last number is 2. To change the time interval for each chart, modify the **timefilters** parameter. For example:

   ```
   "filter": {
                   "timefilters": {
                      "granularity": "year",
                      "lastnum": 2,
                      "type": "relative"
                   }
                },
   ```

   The granularity can be specified as second, minute, hour, day, week, month, or year.
5. Save the changes. Refresh the `Search Dashboards` panel on IBM Operations Analytics - Log Analysis UI to see the new dashboard.

IBM MQ Insight Pack References

Important references for your IBM MQ Insight Pack.

IBM WebSphere MQ library:
http://www-01.ibm.com/software/integration/wmq/library/

Ticket Analytics Insight Packs

The Ticket Analytics Insight Packs are provided with IBM Operations Analytics - Log Analysis.

The Ticket Analytics Insight Packs provide IBM Operations Analytics - Log Analysis users who implemented a ticketing with solutions to search and analyze tickets.

There are three Ticket Analytics Insight Packs for the three ticketing solutions. The Insight Packs and solutions are as follows:

- `TicketAnalyticsForControlDesk_InsightPack_`*<version>*
- `TicketAnalyticsForRemedy_InsightPack_`*<version>*
- `TicketAnalyticsForServiceNow_InsightPack_`*<version>*

The Insight Packs include the logstash based adapter. The adapter retrieves the tickets that are then ingested in to the IBM Operations Analytics - Log Analysis server where each ticket is split and annotated. During ingestion, the tickets are analyzed and assigned a classification. The classification of tickets ensures that analytical insights are enabled.

The Ticket Analytics Insight Packs install sample dashboards that can be used to visualize insights.

Installing the Ticket Analytics Insight Packs

To search and analyze tickets with a ticketing solution, you must install the Ticket Analytics Insight Packs.

Before you begin

You must install the Insight Pack and logstash based adapter for your ticketing solution. The Insight Packs and solutions are as follows:

- `TicketAnalyticsForControlDesk_InsightPack_`*<version>*
- `TicketAnalyticsForRemedy_InsightPack_`*<version>*
- `TicketAnalyticsForServiceNow_InsightPack_`*<version>*

Procedure

1. Upload the appropriate `TicketAnalyticsInsightPacks.zip` file to the system where IBM Operations Analytics - Log Analysis is installed.
2. Copy the `TicketAnalyticsInsightPacks.zip` to <HOME>/IBM/LogAnalysis and extract it. This creates a `ticketanalytics` folder that contains the installation files and logstash ticket adapter compressed file.
3. Edit the Ticket Analytics installation script to make it executable. For example, to use the command line to edit the script, run the following command:

   ```
   chmod +x icd_<insight_pack>_install.sh
   ```

   where *<insight_pack>* is the name of the Ticket Analytics Insight Pack.
4. Make copies of the following files:
   - <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/server.xml
   - <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityConfig.xml
5. To install a Ticket Analytics Insight Pack, run the following script:

   ```
   ./icd_<insight_pack>_install.sh -install -home <HOME>
   -U <LAadmin_user> -P <LAadmin_user_password>
   ```

   where
   - *`<insight_pack>`* is the name of the Ticket Analytics Insight Pack.
   - <HOME> is the home directory.
   - *<LAadmin_user>* is the admin user name.
   - *<LAadmin_user_password>* is the admin user password.

   Note: The IBM Operations Analytics - Log Analysis server restarts once during the installation.

Results

The selected Ticket Analytics Insight Pack, including the Ticket Dashboard applications is installed.

Example

Use the following commands to modify a Ticket Analytics Insight Pack.

-install

To install a Ticket Analytics Insight Pack. For example,

```
/icd_addon_install.sh -install -home <HOME> -U
<LAadmin_user> -P <LAadmin_user_password>
```

-uninstall

To uninstall a Ticket Analytics Insight Pack. For example,

```
/icd_addon_install.sh -uninstall -home <HOME> -U
<LAadmin_user> -P <LAadmin_user_password>
```

-list To list the installed Ticket Analytics Insight Packs. For example,

```
/icd_addon_install.sh -list -home <HOME> -U
<LAadmin_user> -P <LAadmin_user_password>
```

Uninstalling a Ticket Analytics Insight Packs

You can uninstall a Ticket Analytics Insight Packs.

Before you begin

You must manually delete all of the data from the created data sources before uninstalling a Ticket Analytics Insight Pack.

About this task

You can uninstall the following Ticket Analytics Insight Pack:

- `TicketAnalyticsForControlDesk_InsightPack_`*<version>*
- `TicketAnalyticsForRemedy_InsightPack_`*<version>*
- `TicketAnalyticsForServiceNow_InsightPack_`*<version>*

Procedure

To uninstall a Ticket Analytics Insight Pack, run the following script:

```
./icd_addon_install.sh -uninstall -home <HOME>
-U <LAadmin_user> -P <LAadmin_user_password>
```

where

- `icd_addon` is the name of the Ticket AnalyticsInsight Pack.
- <HOME> is the home directory.
- *<LAadmin_user>* is the admin user name.
- *<LAadmin_user_password>* is the admin user password.

Setting up the logstash adapter for ticketing

You must install the logstash based adapter for your ticketing solution.

About this task

The logstash adapter included with the Ticket Analytics Insight Pack for `IBM Control Desk` is connected to an `IBM Control Desk` instance. The logstash adapters included with Ticket Analytics Insight Pack for `Remedy` and Ticket Analytics Insight Pack for `ServiceNow` are file-based. The file-based adapters require each ticket instance to be separated by a line, and within each record the ticket column data is separated by a comma delimiter.

Procedure

1. Copy and extract the LogstashTicketAdapters.zip to <HOME>/IBM/LogAnalysis to create a *<plugin_home>* folder.
2. To import the IBM Operations Analytics - Log Analysis client certificate to the JVM, complete the following steps:
   a. Copy the client.crt file from <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security on the IBM Operations Analytics - Log Analysis server.
   b. Run the following command:

   ```
   keytool -import -file <path> -keystore <JAVA_HOME>/jre/lib/security/
   cacerts -storepass changeit
   ```

   where *<path>* is the location of the client.crt, and *<JAVA_HOME>* is the JAVA location.
3. To edit the logstash plug-in, complete the following steps:

   The logstash plug-in is in the *<plugin_home>* folder. For example, the IBM Control Desk ticket adapter, sccd_adapter_logstash.conf, is at *<plugin_home>*/logstash/configs/sccd_adapter_logstash.conf
   a. Input section: Edit the input path for Ticket Analytics Insight Pack for Remedy or ServiceNow to point to the ticket source.
   b. Output section: Edit the following output path configuration parameters for the IBM Operations Analytics - Log Analysis server where the tickets are ingested:

Table 59. Ticket Analytics Insight Pack configuration parameters

Parameter	Description
scala_url	The URL of the IBM Operations Analytics - Log Analysis server where the ticket data is ingested.
scala_user	The IBM Operations Analytics - Log Analysis server user name.
scala_password	The IBM Operations Analytics - Log Analysis server password.
disk_cache_path	The disk cache location on the IBM Operations Analytics - Log Analysis server.

4. If you are using IBM Control Desk, edit the following parameters in the *<plugin_home>*/logstash/configs/SCCDConfig.properties file:

Table 60. Ticket Analytics Insight Pack IBM Control Desk properties

Parameter	Description
HOSTNAME	The IBM Control Desk host name.
PORT	The IBM Control Desk server http port.
USERNAME	The IBM Control Desk user name.
PASSWORD	The IBM Control Desk user's password.
TIME_BETWEEN_FETCH	The time between fetching 2 consecutive tickets.
DATE_FORMAT	The date format for the ticket date attributes.
IS_SSL	The security protocol setting. The default setting is 0. Set to 1 if the IBM Control Desk connection uses HTTPS.

Table 60. Ticket Analytics Insight Pack IBM Control Desk properties (continued)

Parameter	Description
PROTOCOL	The security protocol. You can use SSL or TLS.

5. Edit the <JAVA_HOME> variable in the start script to point to your JVM. The start script is in the *<plugin_home>* folder.
6. Edit the start script to make it executable. For example, to use the command line to edit the script, run the following command:

```
chmod +x startICDTicketAdapter
```

7. To start the logstash adapter, run the start script.

Ticket Analytics Insight Pack data format

The Ticket Analytics Insight Packs require data to be saved or converted in to a specific format.

Data format

The Ticket Analytics Insight Packs require data in the CSV format. Individual records must be separated by a line, and within each record the ticket column data must be separated by a comma delimiter.

The logstash adapter for IBM Control Desk converts the data that it fetches from the configured IBM Control Desk instance in to the required format.

The logstash adapters for Remedy and ServiceNow read data from a file. They do not convert data. Therefore, the data must be in the required format.

Data can be sent in batches. The first line of each batch must contain the column names.

Supported columns

A set of columns is defined as a JSON file in the Ticket Analytics Insight Pack configuration folder, which is located at <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/tickets/config. The followingJSON files, specific to each Ticket Analytics Insight Pack are in the configuration folder:

- SCCDAnnotationMap.json – For IBM Control Desk Columns
- RemedyAnnotationMap.json – For Remedy
- Columns ServiceNowAnnotationMap.json – For ServiceNow Columns

The JSON files contain an entry for each column in the following format: Column_Name : Internal Column Name where the Column_Name is the name of the ingested column, and the Internal Column Name is the internal name that is defined in the index configuration.

Ticket Analytics Insight Pack configuration properties

After you complete the required Ticket Analytics Insight Pack installation to modify the configuration properties edit the ticketanalytics.properties file.

The ticketanalytics.properties file is at <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/tickets/config.

Modify only the properties in the table.

Table 61. Ticket Analytics Insight Pack configuration properties

Parameter	Description
FIRST_LINE_COLUMNS	Defines the data type. If this property is set to*yes*, the first line in the ticket data batch is defined as Column data. If this property is set to *no*, the data in the batch is considered ticket data. The default columns that are defined in the JSON files are considered in order.
SERVICENOW_TIMEFORMATS	Defines the default input time formats that are expected in the ticket data. You can specify one or more different formats. When you are parsing a date type, the formats are tried in the order they are defined.
REMEDY_TIMEFORMATS	Defines the default input time formats that are expected in the ticket data. You can specify one or more different formats. When you are parsing a date type, the formats are tried in the order they are defined.
SCCD_TIMEFORMATS	Defines the default input time formats that are expected in the ticket data. You can specify one or more different formats. When you are parsing a date type, the formats are tried in the order they are defined.

Ticket Analytics Insight Pack dashboards

There are three dashboards for each ticket type that is packaged with the Ticket Analytics Insight Packs.

The dashboards in table 1 are included with the Ticket Analytics Insight Packs for each of the ticket types, `IBM Control Desk`, `Remedy`, and `ServiceNow`. Users can create more charts and dashboards based on ingested ticket data.

Table 62. Dashboard descriptions

Dashboard	Description
Problem Origin	The charts on this dashboard illustrate the classification categories and ticket attributes that relate to the origin of a problem.
Problem Resolution	The charts on this dashboard illustrate the classification categories and ticket attributes that relate to the resolution of a problem.
Problem Trends	The charts on this dashboard illustrate how problems related to tickets trend historically.

Configuring aliases

You can use the IBM Operations Analytics - Log Analysis alias feature to specify an alias.

The IBM Operations Analytics - Log Analysis alias feature is installed and enabled by default.

The alias data is stored in the `aliasSource.json` file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF directory.

Configuring an alias

You can use the IBM Operations Analytics - Log Analysis alias feature to specify an alias, consisting of an alternative name.

About this task

Aliases are displayed in the IBM Operations Analytics - Log Analysis UI.

Procedure

1. Open the `aliasSource.json` file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF directory.
2. To apply an alias to a fieldname, edit the fieldname parameters in the `aliasSource.json` file. The following example shows the parameters to apply an alias to a hostname:

```
{
  "aliasKeyValueArray": [
    {
      "sourceType": "WASSystemOut",
      "fields": [
        {
          "fieldName": "hostname",
          "alias_field": "severity_alias",
          "translations": {
            "E": "Error",
            "W": "Warning",
            "O": "Overflow",
            "I": "Info"
          }
        }
      ]
    }
  ]
}
```

   Where:

   - **`sourceType`** specifies the source type for which the alias is created.
   - **`fieldName`** specifies the field name of the source type.
   - **`alias_field`** specifies the alias name of the field.
   - **`translations`** contains the field values and aliases.

What to do next

Verify that the aliases are correctly configured in the IBM Operations Analytics - Log Analysis UI.

Errors that are recorded during configuration are stored in the `UnityApplication.log` file in the `<HOME>/IBM/LogAnalysis/logs` directory.

Configuration reference

Read reference information about the scripts and properties, which you can configure after you install Log Analysis.

ldapRegistryHelper.properties

You can edit the ldapRegistryHelper.properties to specify LDAP server details.

The following properties are required and define the connection information for the target LDAP server.

Table 63. LDAP server connection information properties

Property	Description
ldap_hostname_property=	The LDAP hostname.
ldap_port_property=	The LDAP port.
ldap_baseDN_property=	The LDAP baseDN. For example, "dc=com" for TDS users, and "CN=Users,DC=sflab,DC=local"for AD users.

The following properties are optional and define the connection information for the target LDAP server. Where applicable, default settings are assigned.

The **bindPassword** value for AD users is encrypted in the **ldapRegistryHelper_config.xml**.

Table 64. Optional LDAP server connection information properties

Property	Description
ldap_bindDN_property=	The LDAP bindDN. For example, "CN=Administrator,CN=Users,DC=sflab, DC=local" for AD users.
ldap_bindPassword_property=	The LDAP bind password.
ldap_realm_property=	The LDAP realm. The default value is LdapRegistryRealm.
ldap_id_property=	The LDAP ID. The default value is LdapRegistryId.
ldap_ignoreCase_property=	The LDAP ignore case. The default value is true.

ldapRegistryHelper.sh command

You can use the ldapRegistryHelper.sh command to enable a basic connection for user authentication in IBM Operations Analytics - Log Analysis.

For more information about how to use the command to set up LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory, see "Configuring LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory" on page 38.

Supported integrations

This command currently supports connections to the IBM Tivoli Directory Server and Microsoft Active Directory.

Prerequisites

Before you use this command, you must update the ldapRegistryHelper.properties file in the <HOME>/IBM/LogAnalysis/utilities/

directory with the connection and configuration information for the target LDAP server.

Syntax

The `ldapRegistryHelper.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
 ldapRegistryHelper.sh    config | enable
```

Note:

To run the script, the `JAVA_HOME` variable must be set correctly for IBM Operations Analytics - Log Analysis. If the script fails, run the following command to set the `JAVA_HOME` variable:

```
JAVA_HOME=$<HOME>/IBM-java
```

Parameters

The `ldapRegistryHelper.sh` command has the following parameters:

`config` Use the `config` parameter to create an XML file that is called `ldapRegistry.xml` in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory. This file uses the connection and configuration information that is defined in the `ldapRegistryHelper.properties` file.

`enable`

Use the `enable` parameter to enable LDAP authentication that uses the information that is specified in the `ldapRegistry.xml` file. This parameter also disables the reference to the database-managed custom user registry.

`unity` command

Use the `unity` command to start, stop, and restart IBM Operations Analytics - Log Analysis. Also use this command to display the status of processes and to display version information for IBM Operations Analytics - Log Analysis.

Syntax

The `unity` command is located in the <HOME>/IBM/LogAnalysis/utilities directory and has the following syntax:

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

Parameters

The parameters for this command are:

- start Use this parameter to start IBM Operations Analytics - Log Analysis and associated services.

-stop Use this parameter to stop IBM Operations Analytics - Log Analysis and associated services.

-version
Use this parameter to determine the currently installed version of IBM Operations Analytics - Log Analysis.

-restart
Use this parameter to restart IBM Operations Analytics - Log Analysis and other associated services.

-status Use this parameter to display a list of IBM Operations Analytics - Log Analysis processes, including the SOLR nodes, and their status.

Configuration file parameters

The IBM Tivoli Monitoring Log File Agent uses the information that is specified in the configuration file to process log file information.

Table 1 explains that parameters that you can modify in this file.

Table 65. Parameter summary

Parameter	**Description**
`DataSources`	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA conf file, the directory you specify must not contain any subdirectories.
`SshAuthType`	You must set this value to either `PASSWORD` or `PUBLICKEY`. If you set this value to `PASSWORD`, IBM Operations Analytics - Log Analysis uses the value that is entered for `SshPassword` as the password for Secure Shell (SSH) authentication with all remote systems. If you set this value to `PUBLICKEY`, IBM Operations Analytics - Log Analysis uses the value that is entered for `SshPassword` as pass phrase that controls access to the private key file.
`SshHostList`	You use the `SshHostList` value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system. If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.

Table 65. Parameter summary (continued)

Parameter	Description
SshPassword	If the value of the SshAuthType parameter is PASSWORD, enter the account password for the user that is specified in the SshUserid parameter as the value for the SshPassword parameter. If the value of the SshAuthType parameter is PUBLICKEY, enter the pass phrase that decrypts the private key that is specified in the SshPrivKeyfile parameter.
SshPort	You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to 22.
SshPrivKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshPubKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

eifutil.sh command

To administer EIF Receiver instances, use the eifutil.sh command.

Syntax

The eifutil.sh command has the following syntax and is in the *<USER_HOME_REMOTE>*/DataForwarders/EIFReceivers/utilities where *<USER_HOME_REMOTE>* is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where *<Inst_ID>* is the ID for the specific EIF instance.

Parameters

-status

Displays the status for the installed instances. For example:

```
=============================================================================
COMPONENT              Instance         PID            PORT            STATUS
=============================================================================
EIF Receiver           eif_inst_1      13983           6601              UP
EIF Receiver           eif_inst_2      14475           6602              UP
EIF Receiver           eif_inst_3      14982           6603              UP
EIF Receiver           eif_inst_4      15474           6604              UP
EIF Receiver           eif_inst_5      15966           6605              UP
=============================================================================
```

-start <*Inst_id*>
Starts the specified instance.

-stop <*Inst_id*>
Stops the specified instance.

-startAll
Starts all instances.

-stopAll
Stops all instances.

-restart<*Inst_id*>
Restarts the specified instance.

-restartAll
Restarts all the instances.

lfautil.sh command

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the lfautil.sh command.

Syntax

The lfautil.sh command has the following syntax and is in the <*USER_HOME_REMOTE*>/utilities/ directory on the remote host where <*USER_HOME_REMOTE*> is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start Starts all the LFA instances on the remote host.

-stop Stops all the LFA instances on the remote host.

-status
Displays the status for the LFA instances on the remote host. For example:

```
==========================================
COMPONENT            PID            STATUS
==========================================
Log File Agent        23995             UP
==========================================
```

-restart
Restarts the LFA instances on the remote host.

Data Collector properties

Before you can use the data collector to stream data or load a batch of historic data, edit the javaDatacollector.props file.

The javaDatacollector.props file is in the <HOME>/IBM/LogAnalysis/utilitiesdatacollector-client folder.

The logFile, hostname, logpath, and keystore parameters are required.

The userid, password, and keystore parameters are automatically populated with the default values that are created during the installation. If you want, you can change these but you do not need to.

Table 66. Data Collector properties

Parameter	**Value**
logFile	The full path of the file you want to load.
servletURL	The URL of the Data Collector service.
userid	The user ID for the Data Collector service.
password	The password for the Data Collector service.
datasource	The datasource that you want to use to load data.
timestamp	The time stamp to use if a time stamp is not found in the log file.
batchsize	The number of BYTES of logs sent in one batch. The default value is 500,000.
keystore	The full path to the keystore file.
inputType	The valid input type is LOGS.
flush flag	If the default true is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to false no flush signal is sent when the end-of-file is reached.

unity_securityUtility.sh command

You can use the unity_securityUtility.sh command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Syntax

The unity_securityUtility.sh command is in the <HOME>/IBM/LogAnalysis/utilities directory and it has the following syntax:

```
unity_securityUtility.sh encode [textToEncode] [unity.ks]
```

Parameters

The unity_securityUtility.sh command has the following parameters:

encode

The encode action returns an AES encrypted version of the text that you enter as the text to encrypt.

[*textToEncode*]
Use the [*textToEncode*] parameter to enter the password that you want to encrypt. If you do not specify a password for this parameter, IBM Operations Analytics - Log Analysis prompts you for one.

[unity.ks]

The unity.ks file is the default keystore that is generated automatically during installation. It controls how the password is encrypted and decrypted.

The unity.ks file is used to encrypt and decrypt passwords for the following features:

- Java data collector client in the <HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties file.
- EIF Receiver in the <HOME>/IBM/LogAnalysis/utilities/UnityEIFReceiver/config/unity.conf file.

For an example of how to use this command, see "Changing the default EIF Receiver or Data Collector password" on page 266.

eif.conf file

The file eif.conf is a configuration file for the TEC Adapter used by the scala_custom_eif plugin to send log records as events to the IBM Operations Analytics - Log Analysis EIF Receiver.

The file eif.conf is found in the logstash/outputs directory relative to where the logstash Integration Toolkit was installed on the logstash server. The logstash Integration Toolkit installation configures eif.conf during installation with the server location, server port, cache file location, and the log file. You can modify any other properties to customize your installation. You must restart the logstash agent in order effect changes to this configuration file.

Note: Please refer to the comments in the eif.conf file for the latest details on the available parameters and their descriptions.

eif.conf file parameters

BufEvtMaxSize=<kilobytes>

Specifies the maximum size, in kilobytes, of the adapter cache file. The default value is 64. The cache file stores events on disk when the BufferEvents keyword is set to YES. The minimum size for the file is 8 KB. File sizes specified below this level are ignored, and 8 KB is used. There is no upper limit for the file size.

Note: If the cache file already exists, you must delete the file for parameter changes to take effect.

The BufEvtMaxSize parameter is optional.

BufEvtPath=<pathname>

Specifies the full path name of the adapter cache file. This is a required parameter when the BufferEvents value is set to YES.

BufferEvents=YES | MEMORY_ONLY | NO

Specifies how event buffering is enabled.

- **YES** - Stores events in the file specified by the BufEvtPath keyword.
- **MEMORY_ONLY** - Buffers events in memory.
- **NO** - Does not store or buffer events.

The value is not case-sensitive. The default value is YES. This parameter is optional.

ConnectionMode=connection_oriented | connection_less

Specifies the connection mode to use to connect to the IBM Operations Analytics - Log Analysis EIF Receiver. The default value is connection_less.

- `connection_oriented` - A connection is established at adapter initialization and is maintained for all events sent. A new connection is established only if the initial connection is lost. The connection is discarded when the adapter is stopped. This option can be abbreviated to co or CO.
- `connection_less` - A new connection is established and discarded for each event or group of events that is sent.

This parameter is optional.

LogFileName=<pathname>

Specifies the full path name of the log file for the adapter.

LogLevel=<level>

Specifies whether the Java API generates log messages or not. By default, no messages are generated. Specify ALL to generate messages. If you specify any other value or no value, the API does not generate messages. This parameter is optional.

ServerLocation=<host>

Specifies the name of the host where the IBM Operations Analytics - Log Analysis EIF Receiver resides.

`ServerPort=number` Specifies the port number on which the IBM Operations Analytics - Log Analysis EIF Receiver listens for events.

FQDomain=YES | NO | <fully.qualified.domain.suffix>

Specifies the fqhostname slot.

- `YES` - The adapter will attempt to determine the fully qualified hostname and if successful will fill in the fqhostname slot in the event.
- `NO` - The fqhostname slot will be set to a null string.
- `<fully.qualified.domain.suffix>` - The adapter will append this value to the hostname in order to set the fqhostname slot.

unity.conf file

The `unity.conf` is a configuration file for the IBM Operations Analytics - Log Analysis EIF Receiver.

The `unity.conf` file is found in the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/ directory.

unity.conf file parameters

Table 67. `unity.conf` parameters

Parameter	Description
`unity.data.collector.ip=host name`	The host name of the server on which IBM Operations Analytics - Log Analysis is installed
`unity.data.collector.port=9987`	The port that is specified during installation. The default value is 9987.
`unity.data.collector.protocol=https`	The communication protocol. The default value is https.

Table 67. `unity.conf` parameters (continued)

Parameter	Description

`unity.data.collector.uri=`
`/Unity/DataCollector`
	The `uri` used in the `REST` invocation.

`unity.data.collector.userid=unityadmin`	The user ID of the user assigned the `UnityUser` role. The default user ID is `unityAdmin`.

`unity.data.collector.password=`
`{aes}<Unique_string_of_alphanumeric_`
`characters>`
	The password that is associated with the `UnityAdmin` role. The default value is `{aes}<Unique_string_of_alphanumeric_characters>`.

`unity.data.collector.keystore=`
`/home/unity/IBM/LogAnalysis/wlp/usr/`
`server/Unity/keystore/unity.ks`
	The full path to the keystore file.

`unity.data.collector.eif.consumer.`
`num.events=1000000`
	The common queue for all of the `EIF` events. The default value is 1000000.

`unity.data.collector.event.service.`
`num.events=80000`
	Each datasource has 1 service queue. Events are buffered in this queue and placed in batches. The batches are passed to the poster queue. The default value is 80000.

`unity.data.collector.event.poster.`
`num.events=500`
	Each datasource has 1 poster queue. Batches are selected and posted to the IBM Operations Analytics - Log Analysis server from this queue. The default value is 500.

`unity.data.collector.gc.interval=2`	Determine the EIF Receiver memory cleanup interval in minutes. The default value is 2.
`logsource.buffer.wait.timeout=10`	Determine the buffer timeout in seconds. The default value is 10.
`logsource.max.buffer.size=450000`	Determine the buffer size in Bytes. The default value is 450000.

install.sh command

Use the `install.sh` command to install IBM Operations Analytics - Log Analysis or configure data collection for scalability on multiple remote nodes.Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server.

The `install.sh` command is in the <HOME>/IBM/LogAnalysis/remote_install_tool/ directory on the local installation of IBM Operations Analytics - Log Analysis.

install.sh command parameters

To install IBM Operations Analytics - Log Analysis with IBM Installation Manager, run the command:

```
./install.sh
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades, IBM Installation Manager if no other version is installed. For more information, see “Installing IBM Operations Analytics - Log Analysis with the IBM Installation Manager UI” on page 10.

To install IBM Operations Analytics - Log Analysis with the console, run the command:

```
./install.sh -c
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager, if no other version of IBM Installation Manager is installed. For more information, see "Installing with the IBM Installation Manager command-line interface" on page 11.

To silently install IBM Operations Analytics - Log Analysis, run the command:

```
./install.sh -s <HOME_DIR>/smcl_silent_install.xml
```

where *<HOME_DIR>* is your home directory. This command silently installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager Version 1.8.2. For more information, see "Silently installing IBM Operations Analytics - Log Analysis" on page 13.

To install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server, run the command:

```
./install.sh
```

For more information, see "Configuring data collection for scalability on multiple remote nodes" on page 68.

ssh_config.properties

Before you can use the remote installer utility, you must configure the Secure Shell (SSH) for the remote hosts.

The ssh_config.properties file is in the <HOME>/IBM/LogAnalysis/remote_install_tool/config directory.

Table 68. ssh_config parameters

Parameter	Value
REMOTE_HOST=	*REMOTE_HOST*
POST=	*POST*
TIME_OUT=	*60000*
USER=	*REMOTE_USER*
PASSWORD=	*password1*
USE_PASSWORD_LESS_SSH=	*true*
PATH_OF_PASSWORD_LESS_SSH_KEY=	*/home/pass/.ssh/id_rsa*
PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=	*passphrase1*

Audit parameters

The auditing feature is enabled by default. The administrator can modify the default parameters after installation if required.

The unitysetup.properties file is in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF folder. The administrator can edit values for the parameters in table 1, if required.

Table 69. Audit `unitysetup.properties`

Parameters	Value
AUDIT_ACTIONS=	Specifies where the audit data is stored. The default value is `LOG,INDEX`. These values are the only supported values and are enabled by default.
AUDIT_INTERVAL=	Defines how frequently the audit data is written. The default value is `120000` milliseconds.

The audit data is written in JSON format to the `<HOME>/IBM/LogAnalysis/logs/audit.log` file. The audit file is a rolling file, supporting up to 20, 50-MB files by default.

The default file properties are in the `log4j.properties` file in `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/` directory.

Note: If you change any of the audit parameters, you must save your changes and restart IBM Operations Analytics - Log Analysis.

Supported timezone names

IBM Operations Analytics - Log Analysis uses Coordinated Universal Time as the default timezone.

If you need to change the default timezone, you must change the setting after you install IBM Operations Analytics - Log Analysis but before you load any data, including the sample data that is provided on the **Getting Started** page. You cannot change the timezone after you load data.

You must use the full timezone name rather than the timezone abbreviation in the timezone parameter. For example, to change the timezone to Western Europe for Paris, France, edit the parameter as follows:

```
UNITY_TIME_ZONE=Europe/Paris
```

For a full list of the supported timezones see the table below.

Table 70. Supported timezone names

Pacific/Midway
Pacific/Niue
Pacific/Pago_Pago
Pacific/Samoa
US/Samoa
America/Adak
America/Atka
Pacific/Honolulu
Pacific/Johnston
Pacific/Rarotonga
Pacific/Tahiti
US/Aleutian

Table 70. Supported timezone names (continued)

US/Hawaii
Pacific/Marquesas
America/Anchorage
America/Juneau
America/Nome
America/Sitka
America/Yakutat
Pacific/Gambier
US/Alaska
America/Dawson
America/Ensenada
America/Los_Angeles
America/Metlakatla
America/Santa_Isabel
America/Tijuana
America/Vancouver
America/Whitehorse
Canada/Pacific
Canada/Yukon
Mexico/BajaNorte
Pacific/Pitcairn
US/Pacific
US/Pacific-New
America/Boise
America/Cambridge_Bay
America/Chihuahua
America/Creston
America/Dawson_Creek
America/Denver
America/Edmonton
America/Hermosillo
America/Inuvik
America/Mazatlan
America/Ojinaga
America/Phoenix
America/Shiprock
America/Yellowknife
Canada/Mountain
Mexico/BajaSur
Navajo
US/Arizona

Table 70. Supported timezone names (continued)

US/Mountain
America/Bahia_Banderas
America/Belize
America/Cancun
America/Chicago
America/Costa_Rica
America/El_Salvador
America/Guatemala
America/Indiana/Knox
America/Indiana/Tell_City
America/Knox_IN
America/Managua
America/Matamoros
America/Menominee
America/Merida
America/Mexico_City
America/Monterrey
America/North_Dakota/Beulah
America/North_Dakota/Center
America/North_Dakota/New_Salem
America/Rainy_River
America/Rankin_Inlet
America/Regina
America/Resolute
America/Swift_Current
America/Tegucigalpa
America/Winnipeg
Canada/Central
Canada/East-Saskatchewan
Canada/Saskatchewan
Chile/EasterIsland
Mexico/General
Pacific/Easter
Pacific/Galapagos
US/Central
US/Indiana-Starke
America/Atikokan
America/Bogota
America/Cayman
America/Coral_Harbour
America/Detroit

Table 70. Supported timezone names (continued)

America/Eirunepe
America/Fort_Wayne
America/Grand_Turk
America/Guayaquil
America/Havana
America/Indiana/Indianapolis
America/Indiana/Marengo
America/Indiana/Petersburg
America/Indiana/Vevay
America/Indiana/Vincennes
America/Indiana/Winamac
America/Indianapolis
America/Iqaluit
America/Jamaica
America/Kentucky/Louisville
America/Kentucky/Monticello
America/Lima
America/Louisville
America/Montreal
America/Nassau
America/New_York
America/Nipigon
America/Panama
America/Pangnirtung
America/Port-au-Prince
America/Porto_Acre
America/Rio_Branco
America/Thunder_Bay
America/Toronto
Brazil/Acre
Canada/Eastern
Cuba
Jamaica
US/East-Indiana
US/Eastern
US/Michigan
America/Caracas
America/Anguilla
America/Antigua
America/Aruba
America/Asuncion

Table 70. Supported timezone names (continued)

America/Barbados
America/Blanc-Sablon
America/Boa_Vista
America/Campo_Grande
America/Cuiaba
America/Curacao
America/Dominica
America/Glace_Bay
America/Goose_Bay
America/Grenada
America/Guadeloupe
America/Guyana
America/Halifax
America/Kralendijk
America/La_Paz
America/Lower_Princes
America/Manaus
America/Marigot
America/Martinique
America/Moncton
America/Montserrat
America/Port_of_Spain
America/Porto_Velho
America/Puerto_Rico
America/Santiago
America/Santo_Domingo
America/St_Barthelemy
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Thule
America/Tortola
America/Virgin
Antarctica/Palmer
Atlantic/Bermuda
Brazil/West
Canada/Atlantic
Chile/Continental
America/St_Johns
Canada/Newfoundland

Table 70. Supported timezone names (continued)

America/Araguaina
America/Argentina/Buenos_Aires
America/Argentina/Catamarca
America/Argentina/ComodRivadavia
America/Argentina/Cordoba
America/Argentina/Jujuy
America/Argentina/La_Rioja
America/Argentina/Mendoza
America/Argentina/Rio_Gallegos
America/Argentina/Salta
America/Argentina/San_Juan
America/Argentina/San_Luis
America/Argentina/Tucuman
America/Argentina/Ushuaia
America/Bahia
America/Belem
America/Buenos_Aires
America/Catamarca
America/Cayenne
America/Cordoba
America/Fortaleza
America/Godthab
America/Jujuy
America/Maceio
America/Mendoza
America/Miquelon
America/Montevideo
America/Paramaribo
America/Recife
America/Rosario
America/Santarem
America/Sao_Paulo
Antarctica/Rothera
Atlantic/Stanley
Brazil/East
America/Noronha
Atlantic/South_Georgia
Brazil/DeNoronha
America/Scoresbysund
Atlantic/Azores
Atlantic/Cape_Verde

Table 70. Supported timezone names (continued)

Africa/Abidjan
Africa/Accra
Africa/Bamako
Africa/Banjul
Africa/Bissau
Africa/Casablanca
Africa/Conakry
Africa/Dakar
Africa/El_Aaiun
Africa/Freetown
Africa/Lome
Africa/Monrovia
Africa/Nouakchott
Africa/Ouagadougou
Africa/Sao_Tome
Africa/Timbuktu
America/Danmarkshavn
Antarctica/Troll
Atlantic/Canary
Atlantic/Faeroe
Atlantic/Faroe
Atlantic/Madeira
Atlantic/Reykjavik
Atlantic/St_Helena
Eire
Europe/Belfast
Europe/Dublin
Europe/Guernsey
Europe/Isle_of_Man
Europe/Jersey
Europe/Lisbon
Europe/London
GB
GB-Eire
Greenwich
Iceland
Portugal
Universal
Zulu
Africa/Algiers
Africa/Bangui

Table 70. Supported timezone names (continued)

Africa/Brazzaville
Africa/Ceuta
Africa/Douala
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Luanda
Africa/Malabo
Africa/Ndjamena
Africa/Niamey
Africa/Porto-Novo
Africa/Tunis
Africa/Windhoek
Arctic/Longyearbyen
Atlantic/Jan_Mayen
Europe/Amsterdam
Europe/Andorra
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
Europe/Brussels
Europe/Budapest
Europe/Busingen
Europe/Copenhagen
Europe/Gibraltar
Europe/Ljubljana
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Monaco
Europe/Oslo
Europe/Paris
Europe/Podgorica
Europe/Prague
Europe/Rome
Europe/San_Marino
Europe/Sarajevo
Europe/Skopje
Europe/Stockholm
Europe/Tirane
Europe/Vaduz

Table 70. Supported timezone names (continued)

Europe/Vatican
Europe/Vienna
Europe/Warsaw
Europe/Zagreb
Europe/Zurich
Poland
Africa/Blantyre
Africa/Bujumbura
Africa/Cairo
Africa/Gaborone
Africa/Harare
Africa/Johannesburg
Africa/Kigali
Africa/Lubumbashi
Africa/Lusaka
Africa/Maputo
Africa/Maseru
Africa/Mbabane
Africa/Tripoli
Asia/Amman
Asia/Beirut
Asia/Damascus
Asia/Gaza
Asia/Hebron
Asia/Istanbul
Asia/Jerusalem
Asia/Nicosia
Asia/Tel_Aviv
Egypt
Europe/Athens
Europe/Bucharest
Europe/Chisinau
Europe/Helsinki
Europe/Istanbul
Europe/Kiev
Europe/Mariehamn
Europe/Nicosia
Europe/Riga
Europe/Sofia
Europe/Tallinn
Europe/Tiraspol

Table 70. Supported timezone names (continued)

Europe/Uzhgorod
Europe/Vilnius
Europe/Zaporozhye
Israel
Libya
Turkey
Africa/Addis_Ababa
Africa/Asmara
Africa/Asmera
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Juba
Africa/Kampala
Africa/Khartoum
Africa/Mogadishu
Africa/Nairobi
Antarctica/Syowa
Asia/Aden
Asia/Baghdad
Asia/Bahrain
Asia/Kuwait
Asia/Qatar
Asia/Riyadh
Europe/Kaliningrad
Europe/Minsk
Indian/Antananarivo
Indian/Comoro
Indian/Mayotte
Asia/Riyadh87
Asia/Riyadh88
Asia/Riyadh89
Mideast/Riyadh87
Mideast/Riyadh88
Mideast/Riyadh89
Asia/Tehran
Iran
Asia/Baku
Asia/Dubai
Asia/Muscat
Asia/Tbilisi
Asia/Yerevan

Table 70. Supported timezone names (continued)

Europe/Moscow
Europe/Samara
Europe/Simferopol
Europe/Volgograd
Indian/Mahe
Indian/Mauritius
Indian/Reunion
Asia/Kabul
Antarctica/Mawson
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Dushanbe
Asia/Karachi
Asia/Oral
Asia/Samarkand
Asia/Tashkent
Indian/Kerguelen
Indian/Maldives
Asia/Calcutta
Asia/Colombo
Asia/Kolkata
Asia/Kathmandu
Asia/Katmandu
Antarctica/Vostok
Asia/Almaty
Asia/Bishkek
Asia/Dacca
Asia/Dhaka
Asia/Qyzylorda
Asia/Thimbu
Asia/Thimphu
Asia/Yekaterinburg
Indian/Chagos
Asia/Rangoon
Indian/Cocos
Antarctica/Davis
Asia/Bangkok
Asia/Ho_Chi_Minh
Asia/Hovd

Table 70. Supported timezone names (continued)

Asia/Jakarta
Asia/Novokuznetsk
Asia/Novosibirsk
Asia/Omsk
Asia/Phnom_Penh
Asia/Pontianak
Asia/Saigon
Asia/Vientiane
Indian/Christmas
Antarctica/Casey
Asia/Brunei
Asia/Choibalsan
Asia/Chongqing
Asia/Chungking
Asia/Harbin
Asia/Hong_Kong
Asia/Kashgar
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuching
Asia/Macao
Asia/Macau
Asia/Makassar
Asia/Manila
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Urumqi
Australia/Perth
Australia/West
Hongkong
Singapore
Australia/Eucla
Asia/Dili
Asia/Irkutsk
Asia/Jayapura
Asia/Pyongyang
Asia/Seoul

Table 70. Supported timezone names (continued)

Asia/Tokyo
Japan
Pacific/Palau
Australia/Adelaide
Australia/Broken_Hill
Australia/Darwin
Australia/North
Australia/South
Australia/Yancowinna
Antarctica/DumontDUrville
Asia/Khandyga
Asia/Yakutsk
Australia/ACT
Australia/Brisbane
Australia/Canberra
Australia/Currie
Australia/Hobart
Australia/Lindeman
Australia/Melbourne
Australia/NSW
Australia/Queensland
Australia/Sydney
Australia/Tasmania
Australia/Victoria
Pacific/Chuuk
Pacific/Guam
Pacific/Port_Moresby
Pacific/Saipan
Pacific/Truk
Pacific/Yap
Australia/LHI
Australia/Lord_Howe
Antarctica/Macquarie
Asia/Sakhalin
Asia/Ust-Nera
Asia/Vladivostok
Pacific/Efate
Pacific/Guadalcanal
Pacific/Kosrae
Pacific/Noumea
Pacific/Pohnpei

Table 70. Supported timezone names (continued)

Pacific/Ponape
Pacific/Norfolk
Antarctica/McMurdo
Antarctica/South_Pole
Asia/Anadyr
Asia/Kamchatka
Asia/Magadan
Kwajalein
Pacific/Auckland
Pacific/Fiji
Pacific/Funafuti
Pacific/Kwajalein
Pacific/Majuro
Pacific/Nauru
Pacific/Tarawa
Pacific/Wake
Pacific/Wallis
NZ-CHAT
Pacific/Chatham
Pacific/Apia
Pacific/Enderbury
Pacific/Fakaofo
Pacific/Tongatapu
Pacific/Kiritimati

Loading and streaming data

Before you can perform a search on log or other data, you must first load the data into IBM Operations Analytics - Log Analysis. When the file is loaded the data is indexed and is then available to be searched.

There are two main scenarios for loading data:

- Batch loading historic data. For example, you may want to ingest historic log data in a single batch for analysis or for testing.
- Streaming data from a monitored application. You may want to load data that is streamed from a local or remote server.

You can load or stream data from local or remote servers. However, each tool is designed for a particular scenario. This is explained in the *Intended uses of data loading components* table. IBM Operations Analytics - Log Analysis is installed with an internal version of the IBM Tivoli Monitoring Log File Agent. However, IBM Operations Analytics - Log Analysis can also load data from a separate installation of the IBM Tivoli Monitoring Log File Agent, known as an external IBM Tivoli Monitoring Log File Agent.

Table 71. Intended uses of data loading components

	Load batch of historic data		Stream data	
Component	**Local**	**Remote**	**Local**	**Remote**
Data Collector client	Yes	Yes	No	No
Internal IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
External IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
logstash	No	No	No	Yes
Generic Receiver	Yes	Yes	No	No

Note: You must create a Data Source before you configure data loading. For information about creating a Data Source, see the *Administering IBM Operations Analytics - Log Analysis* section of the Information Center. For an overview of the process that you must follow to configure and use IBM Operations Analytics - Log Analysis, see the *Steps to get started with IBM Operations Analytics - Log Analysis* topic in the *Overview of IBM Operations Analytics - Log Analysis* section of the Information Center.
You can load log data into IBM Operations Analytics - Log Analysis using a number of different methods:

Data Collector client

Use the Data Collector client to ingest data in batch mode. This is the easiest method if you want to ingest a large log file for historic analysis if you want to test your IBM Operations Analytics - Log Analysis configuration before attempting the more complex IBM Tivoli Monitoring Log File Agent configuration. The Data Collector client is not designed for ingesting data from remote sources. If you want to ingest a batch of historical data from a remote source, use the IBM Tivoli Monitoring Log File Agent.

For a video that demonstrates how to batch upload a WebSphere Application Server or DB2 file using the Data Collector client, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Videos. For information about batch uploading alternative log file types such as Oracle alert logs, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Batch%20uploading%20Oracle%20Alert%20logs.

IBM Tivoli Monitoring Log File Agent

Use the IBM Tivoli Monitoring Log File Agent for scenarios where you want to stream log data from your production environment or to stream data from a remote server.

For a video that demonstrates how to upload a WebSphere Application Server or DB2 file using the IBM Tivoli Monitoring Log File Agent, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Videos.

logstash

logstash can be used as a method to collect and load data into IBM Operations Analytics - Log Analysis using the logstash Integration Toolkit. For information about the logstash Integration Toolkit, including information about how to download and install it, see logstash Integration Toolkit.

Generic Receiver

Use the Generic Receiver to load data from the REST interface into IBM Operations Analytics - Log Analysis.

You cannot use IBM Operations Analytics - Log Analysis to index log records that contain non-ASCII characters. If your log records contain non-ASCII characters, the records are not added when you use the IBM Tivoli Monitoring Log File Agent or the Data Collector client. When you use the Data Collector client errors that relate to non-ASCII characters are added to the Generic Receiver log.

Example scenarios

The following table outlines a number of example scenarios to help illustrate how you use the different components for different scenarios.

Table 72. Example data loading scenarios

Example	Component
I want to load a batch of historic log data to test the environment.	Data Collector client
I want to monitor an application on a remote server.	IBM Tivoli Monitoring Log File Agent
I want to use logstash to monitor log files on a remote server.	logstash
I want to load a batch of historic log data in the JSON format.	Generic Receiver

Supported operating systems

The supported operating systems that you can install IBM Operations Analytics - Log Analysis are listed in the *Installing* Guide. In addition to these, you also need to know what operating systems are supported by the data streaming and loading scenarios. For example, if you want to use the internal to stream data from a remote source, you need to know the supported operating systems.

Table 73. Supported operating systems for data loading

Scenario	Feature	Supported operating systems
Use the Data Collector to load a batch of historic data	Data Collector	• Red Hat Enterprise Linux Server Edition Version 5 or Version 6 (64 bit) • SUSE Linux Enterprise Server 11 (64 bit)

Table 73. Supported operating systems for data loading (continued)

Scenario	Feature	Supported operating systems
Use the internal IBM Tivoli Monitoring Log File Agent to stream data	Internal IBM Tivoli Monitoring Log File Agent	See the *Requirements for the monitoring agent* topic in the documentation for your version of IBM Tivoli Monitoring at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring
Use an external IBM Tivoli Monitoring Log File Agent to stream data	External IBM Tivoli Monitoring Log File Agent	See the *Requirements for the monitoring agent* topic in the documentation for IBM Tivoli Monitoring 6.2.3.1 at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring

Configuring data streaming

Before you can stream data, you must configure the tools that you use to send the data to IBM Operations Analytics - Log Analysis.

IBM Tivoli Monitoring Log File Agent configuration scenarios

You can use the internal IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis or you can use an external IBM Tivoli Monitoring Log File Agent to stream data from local or remote servers.

You can also use the IBM Tivoli Monitoring Log File Agent to upload a batch of historic data. For more information, see “Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent” on page 265.

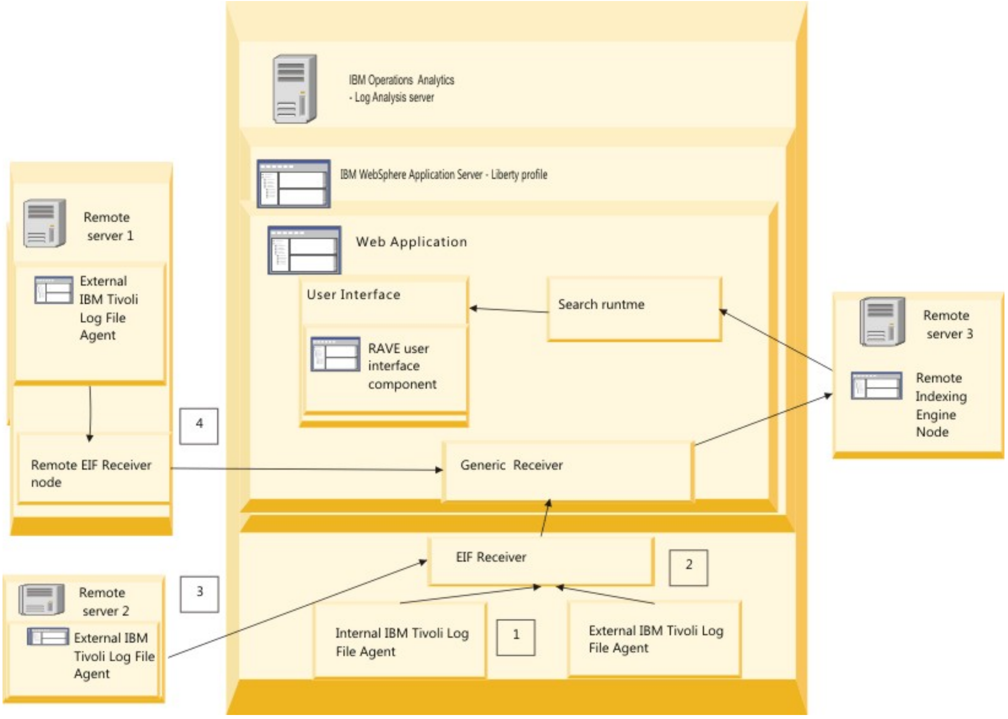
You can integrate the IBM Tivoli Monitoring Log File Agent with IBM Operations Analytics - Log Analysis in two ways.

You can use it with the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis. This is known as the internal IBM Tivoli Monitoring Log File Agent.

You can also use it with an IBM Tivoli Monitoring Log File Agent that has been installed separately as part of another installation.

You can use local and remote versions of both types of IBM Tivoli Monitoring Log File Agent.

The following graphic illustrates these possibilities:


IBM Operations Analytics - Log Analysis server
IBM WebSphere Application Server - Liberty profile
Web Application
User Interface
Search runtme
RAVE user interface component
Generic Receiver
EIF Receiver
2
Internal IBM Tivoli Log File Agent
1
External IBM Tivoli Log File Agent
Remote server 1
External IBM Tivoli Log File Agent
Remote EIF Receiver node
4
Remote server 2
External IBM Tivoli Log File Agent
3
Remote server 3
Remote Indexing Engine Node

The following possible scenarios are illustrated in the graphic:

1. Internal IBM Tivoli Monitoring Log File Agent on a local server
In this scenario, you use the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis to load data from the local installation of IBM Tivoli Monitoring to IBM Operations Analytics - Log Analysis.

2. External IBM Tivoli Monitoring Log File Agent on a local server
In this scenario, you use a version of the IBM Tivoli Monitoring Log File Agent that was not installed with IBM Operations Analytics - Log Analysis but that is installed on the same server as IBM Operations Analytics - Log Analysis.

3. External IBM Tivoli Monitoring Log File Agent on a remote server
In this scenario, you use an installation of an external IBM Tivoli Monitoring Log File Agent to push data to IBM Tivoli Monitoring Log File Agent. To facilitate this integration, you modify the properties of the IBM Tivoli Monitoring Log File Agent.

4. Remote instance of the internal IBM Tivoli Monitoring Log File Agent
In this scenario, you use a the remote installer tool to install a remote instance of the internal IBM Tivoli Monitoring Log File Agent.

The following table summarizes the different configurations required for the scenarios.

Table 74. Configuration for data streaming scenarios

Data streaming scenario	IBM Tivoli Monitoring Log File Agent type	Log file location	Required parameters in `.conf` file
1	Internal and local	Local	`Datasources`

Table 74. Configuration for data streaming scenarios (continued)

Data streaming scenario	IBM Tivoli Monitoring Log File Agent type	Log file location	Required parameters in .conf file
2	Internal and remote. You use the remote installer to create the remote instance.	Remote	Datasources, ServerLocation, ServerPort, BufEvtMaxSize.
3	Local and external	Local	Datasources
4	Remote and external	Remote	Datasources, SshAuthType, SshHostList, SshPassword, SshPort, SshPrivKeyfile, SshPubKeyfile, SshUserid.

Configuring IBM Tivoli Monitoring Log File Agents for use with IBM Operations Analytics - Log Analysis:

You can configure IBM Tivoli Monitoring Log File Agents to start IBM Operations Analytics - Log Analysis.

About this task

For more information about how configuration and format files are used, see "IBM Operations Analytics - Log Analysis configuration and format files" on page 234.

For more information about the required parameters in the configuration file, see "Configuration file parameters" on page 202.

CAUTION:

You cannot use non-ASCII characters in the installation path. The installation path cannot exceed 80 characters.

For more information, about this and about how to configure the monitoring agent in step 3 see:

http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.2/com.ibm.itm.doc_6.3fp2/install/unixconfig_ma.htm?lang=en

Procedure

1. To configure IBM Tivoli Monitoring Log File Agent, run the command:

   ```
   ./itmcmd config -A pc
   ```

   where `pc` is the product code for your agent. For example: `./itmcmd config –A lo`.
2. You are prompted to supply the following information:

   Enter instance name (default is:):
   Enter the instance name. For example, *rhelagent*.

Conf file (default is:):
Enter the configuration file path. For example, `/unity/IBM/ITM/config/lo/`.

Format File (default is:):
Enter the format file path. For example, `/unity/IBM/ITM/config/lo/`.

Note: All fields must be completed. Blank fields cause IBM Tivoli Monitoring Log File Agent to fail.

3. Where prompted, provide the monitoring agent configuration information.
4. To start the IBM Tivoli Monitoring Log File Agent, run the command

```
./itmcmd agent  -o instance name start lo
```

Configuring IBM(r) Tivoli(r) Monitoring Log File Agents:

Before you use the IBM Tivoli Monitoring Log File Agent, you may want to modify the configuration and format files.

About this task

For more information about how configuration and format files are used, see "IBM Operations Analytics - Log Analysis configuration and format files" on page 234.

For more information about the required parameters in the configuration file, see "Configuration file parameters" on page 202.

Procedure

1. Open the configuration file that you want to use.
2. Define the required parameters in the configuration file. The required parameters are different depending on the data loading scenario.
   - If you want to stream data from a local server, specify the data sources in the `DataSources` parameter.
   - If you want to push data from a remote directory, you must specify values for the `Datasources`, `ServerLocation`, `ServerPort`, and `BufEvtMaxSize` parameter.
   - If you want to use an external IBM Tivoli Monitoring Log File Agent that is not installed as part of IBM Operations Analytics - Log Analysis, you must specify values for the `Datasources`, `SshAuthType`, `SshHostList`, `SshPassword`, `SshPort`, `SshPrivKeyfile`, `SshPubKeyfile`, and `SshUserid` parameters.
3. Define the format file as required.
4. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the `.fmt` file for each rotating log type to allow for the appropriate name change. Open the `.fmt` file, and edit the line:

   ```
   -file FILENAME
   ```

   to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the `FILENAME` must read:

   ```
   -file SystemOut*.log
   ```
5. Save your changes.

Example

For example:

```
===============
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PASSWORD
SshPassword=<password>

=====================
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PUBLICKEY
SshPrivKeyfile = <SshUserid_Private_Key_File_Path>
(Or)
SshPubKeyfile = <SshUserid_Private_Key_File_Path>

======================
```

where *<password>* is the password that you want to use.

<SshUserid_Private_Key_File_Path> is the full path for the file that contains the private key of the user that is specified in the SshUserid user. For example, if you save the password to a file called `password.txt` in the `<HOME>/utilities` directory, the full path is as follows:

```
SshPrivKeyfile = <HOME>/utilities/password.txt
```

Configuring IBM Tivoli Monitoring Log File Agent subnodes:

Create an IBM Tivoli Monitoring Log File Agent subnode to group an explicit set of configurations that the IBM Tivoli Monitoring Log File Agent uses to identify and process a log event.

About this task

The subnode consists of a format (`.fmt`) file and a configuration (`.conf`) file. A single instance of the IBM Tivoli Monitoring Log File Agent can have multiple subnodes. Each subnode behaves like a single thread running in the same instance of the IBM Tivoli Monitoring Log File Agent.

You can create subnodes for the following use cases:

Improve performance by making generic format settings more specific
To improve overall performance, you can create specific configurations to replace more generic ones. For example, you can specify the same regular expression (REGEX) in a generic `.fmt` file to parse both WebSphere Application Server (WAS) and DB2 log files. However as the content of the log files differs, this is inefficient. To improve performance, replace the single `.fmt` files with 2 new files containing 2 specific REGEXs for WAS and DB2 in 2 new subnodes.

Improve performance by making generic configuration settings more specific
Similarly, you can improve performance by replacing generic configurations with more specific ones. For example, you can specify the same roll over behaviour in a generic `.conf` to process both WAS and DB2 log files. However as the roll over behaviour in the log files differs, this configuration results in some of the logs not being processed correctly and is inefficient. To improve performance, replace the single `.conf` with 2 new files in 2 new subnodes.

Improve performance for many data sources
If you use a large number of data sources to monitor the log events, you can create subnodes to spread the workload.

Remote monitoring with IBM Tivoli Monitoring Log File Agent 6.3

With IBM Tivoli Monitoring Log File Agent 6.3, you can modify the `.conf` file to monitor logs from multiple remote sources. However, the user credentials may not be the same for the remote machines. You can only maintain 1 set of user credentials in the IBM Tivoli Monitoring Log File Agent configuration file. In this case, you create multiple subnodes with different user credentials in each. This allows you to monitor multiple remote sources from a single IBM Tivoli Monitoring Log File Agent node with multiple subnodes.

There is also a limitation on the naming of subnodes. For more information, see "Character limits for IBM Tivoli Monitoring Log File Agent subnodes names" on page 232.

Procedure

1. Go to the directory where the IBM Tivoli Monitoring Log File Agent is installed. For example, if you are using the internal IBM Tivoli Monitoring Log File Agent that is delivered with IBM Operations Analytics - Log Analysis, the directory is `<Add_path>`.
2. To open IBM Tivoli Monitoring Log File Agent configuration window, run the following command:
   ```
   bin/CandleManage
   ```
3. Right click on the **Tivoli Log File Agent** service and click **Configure**.
4. Click on the instance that you want to configure and click **OK**. The **Configure Tivoli Log File Agent** window is displayed.
5. On the **Log File Adapter Configuration** tab, ensure that the **Conf file** and **Format File** fields are blank.
6. Click on the **Log File Adapter Global Settings** tab and note the directory that is specified in the **Configuration file autodiscovery directory**. This is the directory where you will save the subnode configuration files. Click **OK**.
7. In the subsequent window, you can ignore the other changes and click **Save** to save your changes.
8. Enter the root user password when prompted to implement your changes.
9. Copy the subnode configuration files to the directory that you noted in step 6. The IBM Tivoli Monitoring Log File Agent automatically detects the changes. You do not need to restart the IBM Tivoli Monitoring Log File Agent instance.

Results

The procedure describes how to configure subnodes in the IBM Tivoli Monitoring Log File Agent UI. You can also use the command line. To use the command line to configure the subnodes:

1. Go to the directory where the IBM Tivoli Monitoring Log File Agent is installed.
2. Run the following command:
   ```
   itmcmd config -A lo
   ```
3. Follow the onscreen instructions.
4. Specify the configuration file autodiscovery directory.
5. Complete the configuration.
6. Save the subnode configuration file to the directory that you specified in step 4.

Character limits for IBM Tivoli Monitoring Log File Agent subnodes names:

When you name a subnode, ensure that you are aware of the character and naming limitations.

32 character limitation

The IBM Tivoli Monitoring Log File Agent uses msn to name and identify the subnode. IBM Tivoli Monitoring limits the length of this name to 32 characters. The limit includes the identifier, the dash, and the semi-colon. This leaves 28 new characters for the host name, subnode and configuration file name.

The subnode name is specified in the following format:

```
LO:<Hostname>_<Subnode>-<Conffilename>
```

where `LO` is an identifier that is assigned to all subnodes. <*Hostname*> is the host name of the machine where the subnode is installed. <*Subnode*> is the name of the subnode.<*Conffilename*> is the name of the subnode configuration file.

For example:

```
LO:nc1234567890_WASInsightPack-lfawas
```

However, IBM Tivoli Monitoring limits the length of this name to 32 characters. The example name is 35 characters long. The limit includes the identifier, the dash, and the semi-colon, leaving 28 characters for the host name, subnode and configuration file name. To work around this limitation, IBM Tivoli Monitoring renames the subnode as:

```
LO:nc1234567890_WASInsightPack-l
```

This name is 32 characters long. The host name uses 12 characters. The subnode uses 14 characters.

This limitation can cause an issue if you use similar names for the configuration files. For example, after you name the first subnode, you create another subnode called:

```
LO:nc1234567890_WASInsightPack-lfawas2
```

IBM Tivoli Monitoring renames this subnode as:

```
LO:nc1234567890_WASInsightPack-l
```

As you can see, due to the truncation, both subnodes now have the same name, meaning that the IBM Tivoli Monitoring Log File Agent will not detect the new configuration.

Increasing the limit

The host name is used for integrations with Tivoli Endpoint Manager, where it helps you to identify subnodes. However, this is not required for IBM Operations Analytics - Log Analysis. You remove the host name from the default naming convention so that you can use all 28 characters for the configuration file name.

To change the host name setting:

1. Stop the IBM Tivoli Monitoring Log File Agent.
2. Open the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/lo_default_workload_instance.conf` file.

3. Change the default value for the following property from Y (Yes) to N (No):
   ```
   CDP_DP_USE_HOSTNAME_IN_SUBNODE_MSN='N'
   ```
4. Save the updated file.
5. Restart the IBM Tivoli Monitoring Log File Agent.

After you change this configuration setting, the subnode name no longer includes the host name. For example, the subnodes in the previous example are now named `LO:WASInsightPack-lfawas` and `LO:WASInsightPack-lfawas2`.

Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data:

If you use a IBM Tivoli Monitoring Log File Agent that is installed on a remote server to stream data to IBM Operations Analytics - Log Analysis, you must update the configuration and format files for the IBM Tivoli Monitoring Log File Agent.

Before you begin

You must create a custom data source before you configure data loading. For information about creating a data source, see "Data Source creation" on page 339.

About this task

You can use the configuration files in the `Unity_HOME/IBM-LFA-6.30/config/lo` directory as a basis for the configuration files on your remote server. However, you must ensure that the configuration files that you create:

- contain a line separator between each property that you define in the `.conf` file.
- use the `.conf` file extension and that the format file uses the `.fmt` extension.

To enable the IBM Tivoli Monitoring Log File Agent configuration, complete the following procedure:

Procedure

1. Specify a value or values for the `DataSources` property. If you have multiple locations, you can list the locations and use a comma as a separator. Ensure that you do not leave any spaces. For example, you specify the following values to represent 2 data sources:
   ```
   DataSources=/opt/IBM/WAS1/logs/SystemOut.log,/opt/IBM/WAS2/logs/SystemOut.log
   ```
   When you create a data source for a remote machine, you must enter the correct version of the host name for that machine. To find the correct host name, run the following command on the remote machine:
   ```
   uname -a
   ```
   Enter the name that is returned by this command in the host name parameter for the data source.
2. Specify the server location for the EIF receiver server. For example, for a server that is located at 111.222.333.444, specify the following value:
   ```
   ServerLocation=111.222.333.444
   ```
3. Specify the port that the EIF receiver uses. For example:
   ```
   ServerPort=5529
   ```
4. Specify the `BufEvtPath` for the LFA. The cached events are stored in this file. For example:

```
BufEvtPath=/opt/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache
```

5. Specify the maximum buffer size for the LFA. This is the maximum size that the cache is allowed to be. If the cache is full, events are dropped and performance can decline. The value that you enter here is in kilobytes. For example:

   ```
   BufEvtMaxSize=102400
   ```

6. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the `.fmt` file for each rotating log type to allow for the appropriate name change. Open the `.fmt` file, and edit the line:

   ```
   -file FILENAME
   ```

   to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the `FILENAME` must read:

   ```
   -file SystemOut*.log
   ```

7. Save your changes.

What to do next

Allow time for the log data to be ingested and then search for a value contained in your log file to validate that the configuration has succeeded.

IBM Operations Analytics - Log Analysis configuration and format files:

If you use an internal or external IBM Tivoli Monitoring Log File Agent, you can edit the configuration and property files to suit your specific installation.

The IBM Tivoli Monitoring Log File Agent configuration for a particular data source is defined in the following files:

- A `<name>.conf` file that contains the properties that are used by the IBM Tivoli Monitoring Log File Agent for processing the log files.
- A `<name>.fmt` file that contains an expression and format that is used by the agent to identify matching log file records and to identify the properties to include in the Event Integration Format (EIF) record. The EIF is sent from the agent to the receiving server. The receiving server is the server where the IBM Operations Analytics - Log Analysis server is installed. The `<name>.fmt` file uses a regular expression to determine matching records in the log file and to send each matching record to the IBM Operations Analytics - Log Analysis server in an EIF event.

If you want to use the IBM Tivoli Monitoring Log File Agent to send your log files to IBM Operations Analytics - Log Analysis server, you must customize the regular expression and define your own stanza in the `<name>.fmt` file to capture the log records that are to be sent. The event record format must include the host name, file name, log path, and text message. The IBM Operations Analytics - Log Analysis server uses these values to process the logs. For more information about the IBM Tivoli 6.3 Log File Agent and the configuration files and properties, see Tivoli Log File Agent User's Guide.

The file names must be identical for both files. For example, `WASContentPack_v1.1.0-lfawas.conf` and `WASContentPack_v1.1.0-lfawas.fmt`.

After you modify the configuration files as required, you use the IBM Tivoli Monitoring Log File Agent to load the data into IBM Operations Analytics. For a general description of how to do this, see "Using the IBM Tivoli Monitoring Log File Agent" on page 237

If you use an external instance of the IBM Tivoli Monitoring Log File Agent to load data into the IBM Operations Analytics - Log Analysis server, you must install the configuration files into the agent. This configuration ensures that the agent knows where the log files for a data source are located, how to process the records in the log file, and the server to which records are sent.

LFA configuration file examples

The following example shows the files that are installed as part of the WebSphere Insight Pack that is included as standard with IBM Operations Analytics - Log Analysis.

The `WASContentPack_v1.1.0-lfawas.conf` file contains many properties, including the following examples:

```
# Files to monitor.  The single file /tmp/regextest.log, or any file like
/tmp/foo-1.log or /tmp/foo-a.log.
      LogSources=/home/unityadm/IBM/LogAnalysis/logsources
   /WASInsightPack/*

      # Our EIF receiver host and port.
      ServerLocation=<EIF Receiver host name>
      ServerPort=5529
```

The `WASContentPack_v1.1.0-lfawas.fmt` file contains the following regular expression that matches any record within a monitored log file. In this example, the regular expression matches all the log records in the file and to the Operations Analytics server as an EIF event. The EIF event contains the host name where the agent is running, the file name of the log file, the log file path of the log file, and the log file record itself.

```
 // Matches records for any Log file:
     //

     REGEX AllRecords
     (.*)
     hostname LABEL
     -file FILENAME
     logpath PRINTF("%s",file)
     text $1
     END
```

Configuration file parameters:

The IBM Tivoli Monitoring Log File Agent uses the information that is specified in the configuration file to process log file information.

Table 1 explains that parameters that you can modify in this file.

Table 75. Parameter summary

Parameter	Description
DataSources	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA conf file, the directory you specify must not contain any subdirectories.
SshAuthType	You must set this value to either PASSWORD or PUBLICKEY. If you set this value to PASSWORD, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as the password for Secure Shell (SSH) authentication with all remote systems. If you set this value to PUBLICKEY, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as pass phrase that controls access to the private key file.
SshHostList	You use the SshHostList value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system. If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.
SshPassword	If the value of the SshAuthType parameter is PASSWORD, enter the account password for the user that is specified in the SshUserid parameter as the value for the SshPassword parameter. If the value of the SshAuthType parameter is PUBLICKEY, enter the pass phrase that decrypts the private key that is specified in the SshPrivKeyfile parameter.
SshPort	You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to 22.

Table 75. Parameter summary (continued)

Parameter	Description
SshPrivKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshPubKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

Using the IBM Tivoli Monitoring Log File Agent:

You can use the log file agent to load log file information into IBM Operations Analytics - Log Analysis.

Before you begin

Consider the size of the log files that you want to load. If a log file is in the region of 50 MB, or more, in size, increase the size of the log file agent cache. In the appropriate configuration file, set `BufEvtMaxSize=102400`. For WAS log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf`. For DB2 log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf`.

You must delete the appropriate existing cache file. For WAS log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache` and for DB2 log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache`

For very large log files, update the cache size of the EIF receiver. In the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf` file, increase the value of the `BufEvtMaxSize` property.

Lines in a log that are longer than 4096 characters are, by default, ignored by the IBM Tivoli Monitoring Log File Agent. To force it to read lines longer than 4096 characters, add the `EventMaxSize=<`*`length_of_longest_line`*`>` property to the `.conf` file that will be used while loading the log.

For WAS update `$UNITY_HOME/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf` file. DB2 update `$UNITY_HOME/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf` file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the `<HOME>/IBM/LogAnalysis/utilities` directory, run the following commands:

- `<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop`
- `<HOME>/IBM/LogAnalysis/utilities/unity.sh -start`

About this task

The IBM Tivoli Monitoring Log File Agent might be on the same server as IBM Operations Analytics - Log Analysis and monitoring a local directory. In this scenario, the installation of IBM Operations Analytics - Log Analysis completes all of the configuration required.

If the IBM Tivoli Monitoring Log File Agent is on the same server as IBM Operations Analytics - Log Analysis, but monitoring remote directories, some additional configuration is required. If you want to monitor log files on remote servers, you must make some specific settings changes. For more information about these specific settings, see the *Configuring remote monitoring that uses the predefined configuration files* topic under *IBM Tivoli Log File Agent Configuration* in the *Extending IBM Operations Analytics - Log Analysis* section.

If your configuration requires it, you can use a remote IBM Tivoli Monitoring Log File Agent. In this scenario, install and configure the IBM Tivoli Monitoring Log File Agent based on the your requirements. For more information, see the IBM Tivoli Monitoring documentation: http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/welcome.htm

Procedure

To use the log file agent to load log information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. Ensure that the log file you want to add is in the appropriate directory. For WAS logs, place the log file in the following directory:

   ```
   <HOME>/IBM/LogAnalysis/logsources/WASInsightPack
   ```

   For DB2 logs, place the log file in the following directory:

   ```
   <HOME>/IBM/LogAnalysis/logsources/DB2InsightPack
   ```

   For Generic annotator log files, place the log file in the following directory:

   ```
   $UNITY_HOME/logsources/GAInsightPack
   ```

   The log file is automatically picked up and analyzed. Depending on the size of the log file, processing it could take some time.
3. Optional: To monitor progress, check the following log files:
   - `<HOME>/IBM/LogAnalysis/logs/GenericReceiver.log`
   - `<HOME>/IBM/LogAnalysis/logs/UnityEifReceiver.log`

   When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `$UNITY_HOME/logs` directory to ensure that the data ingestion has completed correctly.

   This example illustrates the addition of a batch of log records. The result is indicated in the `RESPONSE MESSAGE` section of the log file:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO  - LogEventPoster : -
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 2078,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    ++++++++++++++++++++++++++++++++++++
2013-04-2 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    EIF event delivery to Generic Receiver -- SUCCESS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

In this log, the number of log records processed is indicated in the line:

```
{    "batchSize": 2078,    "failures": [    ],    "numFailures": 0 }
```

2078 log records were successfully ingested. The `numFailures` value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the `numFailures` value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a `TIMEOUT FLUSH` event. These events are added to the log file at the end of the session of data collection:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

2013-04-20 04:54:26,341 [pool-4-thread-1] INFO  - LogEventService :
 TIMEOUT FLUSH for logsource:nc9118041070::
  /home/example/LogAnalytics/logsources/
WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO  - LogEventPoster : ---
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 1714,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO  - LogEventService :
   POST RESULT:
{"failures":[],"batchSize":1714,"numFailures":0}

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

To calculate the number of events that have been processed, calculate the sum of all of the `batchSize` values. To calculate the number of events ingested, calculate the sum of all of the `batchSize` values and deduct the total sum of `numFailure` values.

If the ingestion fails, an error message is recorded in the `UnityEIFReceiver.log`:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster : Not Found
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
"CTGLA0401E : Missing log source ","RESPONSE_CODE":404}
```

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   FAILURE -  ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413 Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the `$UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`.

500 Internal Server Error: Displayed when there is any issue withIBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404 Not Found: Displayed when a Log Source is not found for a hostname and log path combination in the request.

409 Conflict: Displayed if the data batch is posted for a Log Source that is an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the `inputType` field in the request JSON does not match the `inputType` field in the Collection for the requested hostname and log path combination.

200 OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400 Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Results

After the task completes, the log file is indexed and can be searched using the **Search** field on the IBM Operations Analytics - Log Analysis Dashboard.

Considerations when using the IBM Tivoli Monitoring Log File Agent:

Before you configure the IBM Tivoli Monitoring Log File Agent to ingest data, update the IBM Tivoli Monitoring Log File Agent to ensure that the configuration is appropriate to the log file that you are likely to ingest.

Log file size

If your log files are likely to exceed 50 MB, increase the size of the IBM Tivoli Monitoring Log File Agent cache: In the appropriate configuration file, set `BufEvtMaxSize=102400`. For WAS log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf`. For DB2 log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf`.

You must delete the appropriate existing cache file. For WAS log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache` and for DB2 log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache`

For very large log files, update the cache size of the EIF receiver. In the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf` file, increase the value of the `BufEvtMaxSize` property.

For WAS, update <HOME>/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf file. DB2 update <HOME>/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the <HOME>/IBM/LogAnalysis/utilities directory, run the following commands:

- unity.sh -stop
- unity.sh -start

Maximum log line length

The IBM Tivoli Monitoring Log File Agent monitors each log file line. The default maximum line length that can be processed by the IBM Tivoli Monitoring Log File Agent is 4096 bytes. This is equivalent to 4096 ASCII characters. This limitation is related to the log line and not the log record. If a log record consists of multiple log lines, such as in the case of a stack trace, the limit applies to each line. This is a limitation of the IBM Tivoli Monitoring Log File Agent and does not apply if you use an alternative data collection mechanism.

Performance implications of using the IBM Tivoli Monitoring Log File Agent

Loading logs using the IBM Tivoli Monitoring Log File Agent is a CPU bound process. If your system does not meet the minimum requirements you will need to increase the MaxEventQueueDepth. On some systems, altering this value may produce a noticeable impact on performance. This will buffer additional IBM Tivoli Monitoring Log File Agent events while they are waiting to be processed. The required value for MaxEventQueueDepth may vary depending on the size of the rolled log and the number/speed of your CPU's. If you choose not to increase this value, then older events may be replaced on the event queue by newer events and not sent to the IBM Operations Analytics - Log Analysis server.

To minimize the chance of data loss due to CPU bottlenecks, and to reduce the latency between when a log record is written to the file and when it is loaded, we recommend that the maximum size of a log be small enough so that you system does not fall behind while processing the logs.

Common IBM Tivoli Monitoring Log File Agent configuration conflicts:

When you create a remote IBM Tivoli Monitoring Log File Agent (LFA) node and a custom data source and both use the same log path, you can create a conflict.

When you create a custom data source and use it monitor a directory on a remote LFA subnode and you later create another data source, like a remote data source, that monitors the same directory, you can create a conflict in the LFA configuration. These conflicts may cause errors in the Log Analysis log files and reduce the performance of Log Analysis.

The following example is provided to help you to understand this situation.

To avoid these conflicts, you need to avoid monitoring the same directory with different data sources. If you want to monitor two files in the same directory, include the file name in the **Log Path** field when you create the data source.

Example

For example, you are an administrator and you want to monitor files from an LFA that is installed on a remote server as described in the Knowledge Center documentation. See “Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data” on page 233. In this case, the LFA is not part of the Log Analysis product.

First, you must create a custom data source called `Customdatasource` to load data from remote instance of the LFA. In the Data Source creation wizard, you specify the host name and the following log path:

```
/opt/WAS/WAS_logs/myLogFile.log
```

Next, you need to create the configuration and format files for the LFA sub nodes. You create two files, `lfa1.conf` and `lfa1.fmt`. In the `lfa1.conf` file, you specify the following data source:

```
Datasources=/WAS/WAS_logs/some_dir/*
```

Logs that are subsequently generated or appended are ingested by the `Datasource1` data source.

After some time, you create another data source to load data from the same remote server. The new log file is called `newLogFile.log` and it is located in the same directory as the file that you created the `Customdatasource` data source for. You create a remote data source called `Remotedatasource` and specify the log path as:

```
/opt/WAS/WAS_logs/newLogFile.log
```

Finally, you push the log files into Log Analysis.

However, after you push the log file, you notice some strange behaviour in the Log Analysis log files. The `GenericReceiver.log` log file shows that the data is being ingested for `/opt/WAS/WAS_logs/newLogFile.log`. However, it also says that the `/opt/WAS/WAS_logs/newLogFile.log` log file is not a valid data source.

This occurs because the same log file is being monitored by both data sources. As a result, it is monitored by two different LFA sub nodes and in two different streams. The data is loaded but this can waste resources and decrease the overall performance.

To avoid this situation, you must be aware of any possible conflicts especially when you create a custom data source that monitors a directory rather than a file.

Regular expression support for the LFA:

The IBM Tivoli Monitoring Log File Agent (LFA) supports specific implementations of regular expressions.

Single-line unstructured data

If you want to use the DSV toolkit to extract and export the data in the comma-separated value (CSV) format for use with the DSV toolkit, you can use a regular expression to extract and export the data.

For example, consider the following log file record:

```
10453072 23460 E5D27197E653C548BDA744E8B407845B AOBEAI1 /EAI      I H R SACP9002
BPUSRSYS/612   23460 - XGNEA108:662:000042:06:E036977:WWS00003:7000:16:1:REV=N
Proc Time=000.03
```

You can configure Log Analysis to use a regular expression to extract and export the data in the comma-separated value (CSV) format. For example, here is an example of a regular expression that is defined in the `.fmt` file:

```
REGEX EAILOG
⌂([0-9]*)(.*)SACP9002(.*):([0-9]*):([0-9]*):([0-9]*):([a-zA-Z0-9]*):
([a-zA-Z0-9]*):([a-zA-Z0-9]*):
(.*)Proc Time=([0-9]*.[0-9]*)
timestamp $1 CustomSlot1
discard $2
SACP9002 $3
bankID $4 CustomSlot3
branchID $5 CustomSlot4
discard3 $6
tellerSID $7 CustomSlot5
workstationID $8 CustomSlot6
transactionTypeID $9 CustomSlot7
discard4 $10
responseTime $11 CustomSlot8
msg PRINTF("%s,%s,%s,%s,%s,%s,%s",timestamp,bankID,branchID,tellerSID,workstationID,
transactionTypeID,responseTime)
END
```

Manipulating date time information for the Generic Annotation Insight Pack

If you use the Generic Annotation Insight Pack or the date time rule set from the Generic Annotation Insight Pack in a custom Insight Pack, you can use some limited regular expressions that you can use to parse time and date information.

The second delimiter, which is a colon (:), is not supported. The regular expression replaces the second delimiter with a period (.), which is supported. For example, to change a date from `15/12/2014 12:12:12:088 GMT` to `15/12/2014 12:12:12.088 GMT`, you can add the following regular expression to the `.fmt` file:

```
// Matches records for any Log file:
// Log Analytics Data Source chas_access.log

REGEX nongr
([0-9][0-9])/([0-9][0-9])/([0-9][0-9]) ([0-9][0-9]):([0-9][0-9])
:([0-9][0-9]):([0-9][0-9][0-9]) ([A-Z][A-Z][A-Z])
(.*Batch Status for.*)
month $1
day $2
year $3
hour $4
minute $5
second $6
ms $7
zone $8
message $9
hostname example.com
-file /opt/la/IBM/LogAnalysis/logs/GenericReceiver.log
RemoteHost ""
logpath PRINTF("%s",file)
text PRINTF("%s/%s/%s %s:%s:%s.%s %s %s", month, day, year, hour, minute,
second, ms, zone, message)
END
```

Troubleshooting data loading:

When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `<HOME>/logs` directory to ensure that the data ingestion has completed correctly.

This example illustrates the addition of a batch of log records. The result is indicated in the `RESPONSE MESSAGE` section of the log file:

```
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO  - LogEventPoster : -----------
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 2078,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    ++++++++++++++++++++++++++++++++++++
2013-04-2 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    EIF event delivery to Generic Receiver -- SUCCESS
```

In this log, the number of log records processed is indicated in the line:

```
{    "batchSize": 2078,    "failures": [    ],    "numFailures": 0 }
```

2078 log records were successfully ingested. The `numFailures` value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the `numFailures` value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a `TIMEOUT FLUSH` event. These events are added to the log file at the end of the session of data collection:

```
2013-04-20 04:54:26,341 [pool-4-thread-1] INFO  - LogEventService :
 TIMEOUT FLUSH for datasource:nc9118041070::
  /home/yogesh/IBM/LogAnalysis/logsources/WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO  - LogEventPoster : -----------
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 1714,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO  - LogEventService :
   POST RESULT:
{"failures":[],"batchSize":1714,"numFailures":0}
```

To calculate the number of events that have been processed, calculate the sum of all of the batchSize values. To calculate the number of events ingested, calculate the sum of all of the batchSize values and deduct the total sum of numFailure values.

If the ingestion fails, an error message is recorded in the UnityEIFReceiver.log:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster : Not Found
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
"CTGLA0401E : Missing data source ","RESPONSE_CODE":404}
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   FAILURE -  ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413 Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the $UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties.

500 Internal Server Error: Displayed when there is any issue withIBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404 Not Found: Displayed when a data source is not found for a hostname and log path combination in the request.

409 Conflict: Displayed if the data batch is posted for a data source that is an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the inputType field in the request JSON does not match the inputType field in the Collection for the requested hostname and log path combination.

200 OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400 Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Configuring the EIF Receiver

How to configure remote or local installations of the Tivoli Event Integration Facility (EIF) receiver to work with IBM Operations Analytics - Log Analysis.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Configuring receiver buffer size and timeout:

When collecting data using the IBM Tivoli Monitoring Log File Agent (LFA) and Tivoli Event Integration Facility (EIF) Adapter flow, you might need to change the rate at which events are flushed to the generic receiver for indexing. Incoming events are buffered at the EIF receiver side.

About this task

To improve overall IBM Operations Analytics - Log Analysis performance, you can configure the 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. When the event rate decreases, decrease the buffer size and keep the timeout interval at the default value or increase it, depending on the event rate.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the buffer size and timeout parameters:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder used for the specific remote EIF instance.
3. Change the Timeout and Buffer Size parameters to suit your operating environment:

   ```
   #Timeout in Seconds
   logsource.buffer.wait.timeout=10
   #Buffer Size in Bytes
   logsource.max.buffer.size=250000
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see "`eifutil.sh` command" on page 70.

Results

With higher buffer sizes, notice that it takes a longer time to fill the buffer with events and for batches to be posted to the receiver.

Configuring the EIF receiver user account:

The Tivoli Event Integration Facility (EIF) receiver uses the default `unityuser` user account to access the generic receiver. You can change the user account or the default user password in the `unity.conf` configuration file.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the default EIF user or password:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder that is used for the specific remote EIF instance.
3. Change the following `userid` and `password` parameters to suit your operating environment:
   ```
   unity.data.collector.userid=unityuser

   unity.data.collector.password=password
   ```
   To encrypt the password, use the `unity_securityUtility.sh` command. For more information, see "Changing the default password for the Data Collector and EIF Receiver" on page 266.
4. Save your changes.
5. Restart IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to restart IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -restart` command to restart the instances. For more information, see "`eifutil.sh` command" on page 70.

Results

The EIF receiver uses the new credentials to access the generic receiver.

Configuring the number of events in the EIF Receiver:

You can configure the number of events that the EIF Receiver stores for each internal queue. If you intend to ingest a large quantity of data and at a high rate, configure these values to larger values. However, increasing this value also increases the memory requirements for EIF Receiver.

About this task

Ensure that you have sufficient memory to support the number of events in the queue.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change this setting:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the *<remote_deployment_location>*`/LogAnalysis/DataForwarders/EIFReceivers/`*<eif_inst_#>*`/config/unity.conf` directory. Where *<remote_deployment_location>* is the directory on the remote machine where you deployed the EIF instance. *<eif_inst_#>* is the folder used for the specific remote EIF instance.
3. Locate these lines and change the value to reflect your requirements:
   ```
   unity.data.collector.eif.consumer.num.events=1000000
   unity.data.collector.event.manager.num.events=20000
   ```
   The following settings are applicable per data source:
   ```
   unity.data.collector.event.service.num.events=20000
   unity.data.collector.event.poster.num.events=500
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

Configuring the EIF Receiver memory clean up interval:

IBM Operations Analytics - Log Analysis ensures that the memory used for data collection with the Log File Agent using a property in the `<HOME>/IBM/`

LogAnalysis/UnityEIFReceiver/config/unity.conf file. The EIF Receiver uses this value to manage the memory usage. The configuration cycle is set to a value in minutes with a default value of 2 minutes.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To configure this property:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the eifutil.sh -stop command to stop the instances. For more information, see "eifutil.sh command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the unity.conf file in the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/ directory.
   - If you use a remote installation of the EIF, the unity.conf file is in the <*remote_deployment_location*>/LogAnalysis/DataForwarders/EIFReceivers/<*eif_inst_#*>/config/unity.conf directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder that is used for the specific remote EIF instance.
3. Change the parameters to suit your operating environment:

   ```
   #gc interval is in minutes
   unity.data.collector.gc.interval=2
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the eifutil.sh -start command to start the instances. For more information, see "eifutil.sh command" on page 70.

Configuring scalable data streaming from multiple, remote sources

To facilitate dynamic data streaming that is scalable across multiple remote sources, you must configure IBM Operations Analytics - Log Analysis after you install it.

To enable data collection from remote hosts, you must complete the following steps:

1. Install Apache Solr on the remote machine.
2. Set up Secure Shell (SSH) communication.
3. Configure SSH to work with the remote installer utility.

4. Use the remote installer utility to install instances of the Event Integration Facility (EIF) or the IBM Tivoli Monitoring Log File Agent (LFA) on remote machines.
5. Configure the EIF so that it is compatible with the remote instances that your create. If you use the LFA, you do not have to configure the local installation. However, you do have to manually configure the sub nodes.

You can also maintain and administer these connections after you set them up.

As an alternative to streaming data, You can batch load data. For more information, see "Loading and streaming data" on page 223.

Installing Apache Solr on remote machines:

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

About this task

If no local instances of Apache Solr exist, then you need to install the instances on the remote machine as soon as you install IBM Operations Analytics - Log Analysis. If there is a local instance of Apache Solr, you can install the remote instances whenever you want.

You must use a non-root user to run the script.

You cannot use the installer to install Apache Solr on a local machine.

You cannot use the installer to install multiple Apache Solr nodes on a single remote machine.

To install Apache Solr on multiple remote machines, run the script separately for each remote machine. You cannot use the installer to install instances of Apache Solr simultaneously or in parallel.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:
   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```
2. To run the `remote_deploy.sh` script, enter the following command:
   ```
   ./remote_deploy_solr.sh -install
   ```
3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password-less SSH authentication is disabled. If password-less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the <HOME>/IBM/LogAnalysis/utilities/config directory. For more

information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

SSH Port
Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

Top-level Installation Directory
To use the default value, which is `<HOME>`, press enter. Alternatively, you can enter the path to the directory where you want to install the DE.

Apache Solr Search Port
To use the default value, 9989, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

Apache Solr Query Service Port
To use the default value, 7205, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

4. To start the installation, press enter. In most cases, the installation takes about 5 minutes to complete.

Results

The results of the installation are output in the log file in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs/ManageSolrnodes.log` file.

To view the status for the instances of Apache Solr that are installed remote machines, run the `unity.sh -status` command.

Example

Here is an example script output:

```
Remote Hostname in FQDN format:12345.example.com
username:unity
password:*********
SSH port: [22]
Top-level Installation Directory: [/home/unity]
Solr Search Port: [9989]
Solr Query Service Port: [7205]

Script is ready for remote installation of Solr:
Review the following inputs ....
-----------------------------------------------------------------------------
Remote Host Name: 12345.example.com
Remote User Name: unity
Remote SSH Port: 22
Top-level remote installation directory: /home/unity
Solr v9.0 - remote installation directory:
/home/unity/IBM/LogAnalysis
Solr - remote ports: 9989, 7205
------------------------------------------------------------------------
['q' - Abort]['Enter' - Install]

Sat Nov 16 03:08:38 CST 2013 Starting remote installation of Solr
, this will take couple of minutes to complete  ....
Sat Nov 16 03:08:38 CST 2013 Waiting for remote installation to complete ....
Sat Nov 16 03:11:47 CST 2013 Successfully installed Solr
Solr on remote host:12345.example.com ....
```

Removing Apache Solr instances:

Before you remove an installation of IBM Operations Analytics - Log Analysis, you must remove Apache Solr.

About this task

Note: Do not remove Apache Solr if IBM Operations Analytics - Log Analysis is still being used. IBM Operations Analytics - Log Analysis does not function properly when any instances of Apache Solr are removed. For this reason, only remove Apache Solr when you are about to uninstall IBM Operations Analytics - Log Analysis.

If you installed Apache Solr locally and remotely, remove the local instance first, then remove the remotely installed instances.

This process uses Installation Manager to remove Apache Solr instances. You can also do so silently. To run the silent removal, run following `imcl -c` command, enter `3` to modify the installation, and remove the instance.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:
   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```
2. To run the `remote_deploy.sh` uninstall script, enter the following command:
   ```
   ./remote_deploy.sh -uninstall
   ```
3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password less SSH authentication is disabled. If password less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the *`<UNITY_HOME>`*`/utilities/config` directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

   SSH Port
   Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

   Top-level Installation Directory
   To use the default value, which is `<HOME>/IBM/LogAnalysis`, press enter. Alternatively, you can enter the path to the directory where Apache Solr is installed.
4. To start the removal, press enter. You can view the logs in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs` directory.

Results

When all the remote nodes are removed, you can safely uninstall IBM Operations Analytics - Log Analysis.

Setting up Secure Shell to use key-based authentication:

Secure Shell (SSH) is a cryptographic network protocol for secure data communication between different computers. You set up key-based authentication between the IBM Operations Analytics - Log Analysis servers and the remote computers to which it connects.

About this task

Benefits of using key-based authentication:
- Data is transferred across a secure channel.
- The administrator is no longer concerned about the password changes for the remote servers.
- The passphrase is independent of the individual server password policy.
- One passphrase is used for multiple servers. Only the public key file must be copied to the client server.

For more information you can view the man pages for **`ssh-keygen`** by running this command:

```
man ssh-keygen
```

Procedure

1. To generate public and private keys, enter the following command:

   ```
   ssh-keygen -t rsa
   ```

   or either of the following commands:

   ```
   ssh-keygen
   (This command generates the same results as ssh-keygen -t rsa.)
   ```

   ```
   ssh-keygen -t dsa
   (If you specify dsa, the generated keys include _dsa in their file names.)
   ```

   The following example shows what a valid output might look like:

   ```
   bash-3.2$
   bash-3.2$ ssh-keygen -t rsa
   Generating public/private rsa key pair.
   Enter file in which you want to save the key (/home/unity/.ssh/id_rsa):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /home/unity/.ssh/id_rsa.
   Your public key has been saved in /home/unity/.ssh/id_rsa.pub.
   The key fingerprint is:
   4a:ef:d5:7a:d8:55:b3:98:a1:1f:62:be:dd:c4:60:6e unity@<variable>.example.com
   The key's randomart image is:
   +--[ RSA 2048]----+
   |                 |
   |                 |
   |                 |
   |             . ..|
   |      . S   .o+.o|
   |     . o   =o++. |
   |      . . +o+E.o |
   |       . ..o=.o  |
   |        . .o.. . |
   +-----------------+
   bash-3.2$
   ```

Enter the passphrase. (The **Enter passphrase** field can remain blank to specify an empty passphrase.)

2. To view the contents of the public key file, run the following commands:

   ```
   cd ~/.ssh
   ls -l id_rsa*
   cat id_rsa.pub
   ```

   The command output is:

   ```
   bash-3.2$
   bash-3.2$ cat .ssh/id_rsa.pub
   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDg0/GGoxGzyC7Awjbwnp0hCaeztIRt6yhAg
   GKdwM7nb7Iiv0RgwT4/48E26K1Ur9HrI1W/j0K0JHQw
   vaAFibqeLmqLdK9ctCE901ywTOPFcYeBYPUF9vp/MgaypgGxVwDbW/e0SNPb7YAtZpjRoqeUq
   oYoKzFXXspQkxdhcQfpx0RYMbQdGGg03hDCM2wr2KP
   VuTVniF2IvDu1C4fcRkUPr8aQNMiuEcJgV3VHhlau/0Uo0YpH53NXKhn/sx8xdyTVsKQ1rhW8
   g07HIVc2Tf9ZF2gYXn/HbjE509xK/APu2nztt0h+Air
   JyT5jYMi/IvSI0zbPyc0p9WijPeG8r/v unity@<variable>.in.ibm.com
   bash-3.2$
   ```

3. Create a directory called `.ssh` on the remote server. Use this to store the public key.
4. Copy the public key file (`id_rsa.pub`) to the `.ssh` directory on the remote client:

   ```
   scp /home/unity/.ssh/id_rsa.pub
   <username>@<remotehostname>:/
   <HOME>/.ssh/id_rsa.pub
   ```

   where *<hostname>* is the system host name and *<username>* is the system user name.
5. Add the content of the public key to the `authorized_keys` file on the remote host.

   ```
   bash-3.2$ ssh <username>@<remotehostname>
   bash-3.2$ cd ~/.ssh
   bash-3.2$ cat id_rsa.pub >> authorized_keys
   bash-3.2$ rm id_rsa.pub
   bash-3.2$ exit
   ```

6. Ensure that there are no duplicate keys for the same client in the authorized_keys file.
7. Log in to the remote computer to ensure that key-based SSH is working:

   ```
   ssh <username>@<hostname>
   ```

   Enter the passphrase, if prompted.

   ```
   bash-3.2$ bash-3.2$ ssh <username>@<remotehostname>
   Enter passphrase for key '/home/unity/.ssh/id_rsa':
   Last unsuccessful login: Mon Jul 15 14:22:37 2013 on ssh from <variable>.example.com
   Last login: Mon Jul 15 14:26:54 2013 on ssh from <variable>.example.com
   $
   ```

   Configuration of key-based authentication is complete.

Results

The steps may not work because different versions of SSH are supported by the operating systems that are used by the remote servers. For more information about how to solve this issue, see the *Secure Shell (SSH) configuration does not work* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Configuring secure shell (SSH) communication for multiple remote hosts:

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

Before you begin

Before you configure SSH for multiple remote hosts, you must configure SSH between IBM Operations Analytics - Log Analysis and the remote hosts. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the Information Center.

About this task

By default, the SSH properties file, `ssh-config.properties` file, is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory. If you save the file to another location, the utility requests that the user enters values for the remote host, user, and password. In this case, the utility does not use the values specified in the file.

If you save the `ssh-config.properties` file in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory, the `eif_remote_install_tool` utility uses the properties specified in the file.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

If you specify values for both the password and the private key file path, the utility uses the file to create a password-less SSH connection.

If you do not specify a value for the password or the private key file path, IBM Operations Analytics - Log Analysis cannot create a connection and instead generates an error message in the log:

```
    ERROR:
    example.unity.remote.SshConfigException:
Property file config/ssh-config.properties must contain at least one of:
PASSWORD, PATH_OF_PASSWORD_LESS_SSH_KEY
    Correct SSH configuration OR reconfigure and retry
    Installation Aborted....!
```

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory and open the `ssh-config.properties` file.
2. Specify values for the following properties for each remote host:
   - Remote host
   - Remote user ID
   - Port
   - Connection timeout in milliseconds. The default is 6000.

   For example:

```
REMOTE_HOST=<REMOTE_HOST>
PORT=<PORT>
TIME_OUT=60000
USER=<REMOTE_USER>
```

3. For password-based authentication, you also need to specify the password in the configuration file. For example:

   ```
   PASSWORD=password1
   ```

4. For public key based authentication, specify the path to the directory that contains the private key file. For example:

   ```
   PATH_OF_PASSWORD_LESS_SSH_KEY=/home/pass/.ssh/id_rsa
   ```

5. If your installation of SSH requires a passphrase, specify the passphrase. For example:

   ```
   PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=passphrase1
   ```

Configuring data collection for scalability on multiple remote nodes:

To facilitate scalable data collection on multiple remote nodes, use the `install.sh` command to install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server.

Before you begin

Before you run the command, you must configure secure shell (SSH) communication between the local installation of IBM Operations Analytics - Log Analysis and the remote host. For more information about how to do so, see "Configuring secure shell (SSH) communication for multiple remote hosts" on page 66.

About this task

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis.

You can use the remote installer in the following scenarios:

- If you have a high rate of data ingestion on multiple data sources. For example, if you have 100 or more events per second and 20 or more data sources.
- If you require improved throughput performance on the remote server.
- If the hardware resources on the remote server are restrained.
- If you want to optimize performance according to the conditions described on the Performance developer works page here: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM Log Analytics Beta/page/Performance and tuning

You can use the command to deploy up to 20 instances of the Tivoli Event Integration Facility Receiver or a single instance of the IBM Tivoli Monitoring Log File Agent on a remote node. The command deploys and configures IBM Java 1.7. The command also configures the deployed Tivoli Event Integration Facility Receiver instance to communicate with the IBM Operations Analytics - Log Analysis Data Collector interface.

However, this command does not configure the IBM Tivoli Monitoring Log File Agent subnode. You must configure this setting manually. Both the remote and local instance of the IBM Tivoli Monitoring Log File Agent can monitor remote data sources. For more information about configuring IBM Tivoli Monitoring Log File Agent, see "Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data" on page 233.

To ensure that the remote instances of the Tivoli Event Integration Facility work with the local Data Collector interface, you must create the remotely deployedTivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances as part of the same installation. This is because the encryption configuration and signature generation is done during the main installation. If you install IBM Operations Analytics - Log Analysis after you set up the remote nodes, you must install the remote Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances again. However, you can remove remote instances of the Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent without installing IBM Operations Analytics - Log Analysis again.

Note: If you use the script to install the remote instance on a server that uses the SUSE Linux Enterprise Server 11 operating system, the script fails. To resolve this issue, see the *Cannot install remote EIF instance on SUSE* topic in the *Troubleshooting* IBM Operations Analytics - Log Analysis guide.

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory and run the `install.sh` command. You are prompted for a series of inputs.
2. Enter the remote installation directory. This value must be the location where the deployed artifacts are installed on the remote host.
3. If you want to deploy the Tivoli Event Integration Facility Receiver, select it. If you do, enter the Tivoli Event Integration Facility Receiver instances that you want to deploy.
4. If you want to deploy the IBM Tivoli Monitoring Log File Agent instance on the remote node, select it.

Results

After you complete the procedure, you can now collect data from the remote hosts.

What to do next

After the initial setup, you will want to periodically change the configuration. IBM provides two commands to start and stop the instances so that you can update the configuration.

To administer Tivoli Event Integration Facility Receiver instances, use the `eifutil.sh` command.

To administer IBM Tivoli Monitoring Log File Agent instances, use the `lfautil.sh` command.

`eifutil.sh` command:

To administer EIF Receiver instances, use the `eifutil.sh` command.

Syntax

The `eifutil.sh` command has the following syntax and is in the *<USER_HOME_REMOTE>*`/DataForwarders/EIFReceivers/utilities` where *<USER_HOME_REMOTE>* is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where <*Inst_ID*> is the ID for the specific EIF instance.

Parameters

-status
Displays the status for the installed instances. For example:

```
============================================================================
COMPONENT            Instance         PID            PORT           STATUS
============================================================================
EIF Receiver         eif_inst_1      13983          6601              UP
EIF Receiver         eif_inst_2      14475          6602              UP
EIF Receiver         eif_inst_3      14982          6603              UP
EIF Receiver         eif_inst_4      15474          6604              UP
EIF Receiver         eif_inst_5      15966          6605              UP
============================================================================
```

-start <*Inst_id*>
Starts the specified instance.

-stop <*Inst_id*>
Stops the specified instance.

-startAll
Starts all instances.

-stopAll
Stops all instances.

-restart<*Inst_id*>
Restarts the specified instance.

-restartAll
Restarts all the instances.

`lfautil.sh` command:

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the `lfautil.sh` command.

Syntax

The `lfautil.sh` command has the following syntax and is in the *<USER_HOME_REMOTE>*`/utilities/` directory on the remote host where *<USER_HOME_REMOTE>* is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start Starts all the LFA instances on the remote host.

-stop Stops all the LFA instances on the remote host.

-status
Displays the status for the LFA instances on the remote host. For example:

```
==========================================
COMPONENT           PID           STATUS
==========================================
Log File Agent        23995           UP
==========================================
```

-restart
Restarts the LFA instances on the remote host.

Loading batches of data

In addition to streaming data directly, you can also load batches of historic data for test or other purposes.

Generic Receiver

The Generic Receiver is a component of IBM Operations Analytics - Log Analysis that supports the REST interface for loading data into IBM Operations Analytics - Log Analysis. The REST API uses JSON (JavaScript Object Notation) as an input and returns JSON as an output after the incoming logs are processed. If an error occurs, the API returns an error code and a message.

Processing a batch

Invoking the Generic Receiver API initiates the processing of a batch that is contained in the Input JSON. Buffer a set of log records to create a batch and send data in batches to IBM Operations Analytics - Log Analysis. The batches must be sent in the order in which logs are generated for a specific data source. The size of each batch must be less than the batch size (500000 bytes) supported by IBM Operations Analytics - Log Analysis. At the minimum, you can send data for a single log record in a batch. The Generic Receiver processes a batch by:

- Splitting the batch into multiple log records by using the Splitter that was specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Annotates every log record that is found by the Splitter by using the Annotator that is specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Indexing the annotated log record in the back-end search engine

As special cases, split and annotated steps are skipped if the Splitter or Annotator is specified as null in the SourceType. Even if there is no data to split, you need to send an empty string in the text field of the Input JSON.

Batching of data at the client might lead to an incomplete log record at the end of the batch. This incomplete log record gets buffered in IBM Operations Analytics - Log Analysis and stitched with the remaining data in the subsequent batch to form a complete log record. This stitching assumes that you are maintaining the log record order of the data that is sent to IBM Operations Analytics - Log Analysis. If the order is not maintained, then logs are not correctly split into log records.

Input JSON

The basic structure of an Input JSON file is:

```
{
"hostname":  ,    (String)
"logpath":" ,   (String)
"batchsize": ,  (String)
"inputType":    // Optional (String) "LOGS";
"flush":   // Optional (boolean)
"payload":   // (JSONObject)
{
"name1":"value1",    // Optional
...
...
"nameN":"valueN" ,      // Optional
text : "log record 1 log record 2 ..."  (String)
 }
}
```

The following parameters in the Input JSON are mandatory:

hostname
: The host name that corresponds to the data source for which you want to ingest data.

logpath
: The log path that corresponds to the data source for which you want to ingest data.

batchsize
: The number of BYTES of logs that are sent in one batch to IBM Operations Analytics - Log Analysis (less than 500,000).

inputType
: The type of input data: `LOGS`.

flush flag
: A flag that indicates to the Generic Receiver whether the last record in the batch is a complete log record. Typically, this flag would be set to true in the last batch upon reaching the end of file.

payload.txt
: This text contains the actual log records to be split, annotated, and indexed into IBM Operations Analytics - Log Analysis. The text portion is split into log records by the Splitter, annotated by the Annotator, and then indexed. If you do not have any log records, but want to index only structured (name-value pairs) data, you can specify this mandatory field as an empty string.

More metadata (optional) to be indexed with every log record of the batch can be specified as name-value pairs in the input JSON or the payload within the input JSON. This metadata is applicable at the batch level. For posting distinct metadata for each log record, send 1 log record at a time in the batch.

Post the input JSON to the following URL:

`http://<UNITY_HOST_NAME>:<UNITY_PORT>/Unity/DataCollector`

where `<UNITY_HOST_NAME>` is the machine on which you installed IBM Operations Analytics - Log Analysis and `<UNITY_PORT>` is the port on which it is running. The default port is 9988. The client (Java or Script) sending data into IBM Operations Analytics - Log Analysis needs to authenticate by using the form-based mechanism that is implemented in IBM Operations Analytics - Log Analysis before the Data Collector API is invoked. Refer to the authentication and security design document for details.

Output JSON

The output that is sent by the Generic Receiver after indexing logs contains the count and detailed information on the failure cases in a JSON Array. The details include the actual logRecord, specific error message, and any exception. The basic structure of an Output JSON file is:

```
{
"batchSize" : ,   // (int)
"numFailures" : ,  // (int)
"failures" :    // (JSONArray)
  [
  {
   "logRecord" : ,  // (JSONObject)
   "errorMessage": ,  // (String)
   "exception" : ,  // (JSONArray)
  },
  .
  .
  .
  {

  }
 ]
}
```

Serviceability

As you send data into IBM Operations Analytics - Log Analysis, you might encounter errors that occur before the incoming batch gets processed or errors that occur during processing of batch and indexing log records.

If errors occur before the incoming batch gets processed, the Generic receiver returns an error code and message. To correct the problem, process the error code, make any required changes, and resend the data.

Possible causes for error code 400 (HttpServletResponse.SC_BAD_REQUEST) include:

- Invalid input JSON
- Input batch size is greater than what is supported (500000 bytes)
- No data source is configured from the Admin UI for the host name and log path combination that is sent in the input JSON
- The input type (LOGS) specified in the batch does not match the value that is specified in the logsource that is configured from the Admin UI

Possible causes for error code 500 (HttpServletResponse.SC_INTERNAL_SERVER_ERROR) include:

- An exception that is encountered in any of the steps of the ingestion pipeline (for example, during splitting of a batch).
- An internal IBM Operations Analytics - Log Analysis database-related error.
- Any other exception in IBM Operations Analytics - Log Analysis.

If errors occur during processing of batch and indexing log records, the output JSON provides details of indexing failure. To correct the problem, process the error code, make any required changes, and resend only the affected log records. Sending the same log record twice to IBM Operations Analytics - Log Analysis results in duplicate records in the back-end index and duplicate records in the search results.

Batch loading historic log data with the Data Collector client

Use the Data Collector client to ingest data in batch mode. Use this method to review historic log data. This is the easiest method if you want to ingest large log files for historic analysis.

Before you begin

If you want to use the Data Collector client to load data from remote sources, you must configure the data collector on the remote host before you can configure the local data collector as described here. For more information, see "Configuring the Data Collector client to ingest data from remote hosts" on page 263.

About this task

If you want to load a log file that does not include time stamp information, ensure that the values for `timestamp` and `timestampFormat` are configured in `javaDatacollector.properties`. IBM Operations Analytics - Log Analysis cannot index log files without a time stamp, but if no time stamp information is found in a log file, the value that is configured in `javaDatacollector.properties` is used.

Procedure

To use the Data Collector client to load log file information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. At the command line, navigate to the `<HOME>/utilities/datacollector-client` directory.
3. Update the configuration file that is used by the Data Collector client, `javaDatacollector.properties`. Set the following properties, as appropriate:

   logFile
   The full path of the file you want to ingest.

   servletURL
   The URL of the Data Collector service.

   userid
   The user ID for the Data Collector service.

   password
   The password for the Data Collector service.

   datasource
   The datasource that you want to use to load data.

   timestamp
   The time stamp to use if a time stamp is not found in the log file.

   batchsize
   The number of BYTES of logs that are sent in one batch. The default value is 500,000.

   keystore
   The full path to the keystore file.

   inputType
   The valid input types are: `LOGS`, `CONFIGFILES`, `SUPPORTDOCS`. The default value is `LOGS`.

   flush flag
   If the default `true` is set, the client sends a flush signal to the Generic

Receiver for the last batch of the file. If set to false, no flush signal is sent when the end of file is reached.

The following sample javaDatacollector.properties file displays the configuration for loading the SystemOut.log log file.

```
#Full path of the file you want to read and upload to Unity
logFile = SystemOut.log
#The URL of the REST service. Update the host/port information if required
servletURL = https://hostname:9987/Unity/DataCollector
#The user ID to use to access the unity rest service
userid=unityuser
#The password to use to access the unity rest service
password=password
datasource=Systemout
#Time stamp to use if your content can not find a time stamp in log record.
The same time stamp would be used for all records
timestamp = 01/16/2013 17:27:23:964 GMT+05:30
#The number of BYTES of logs sent in one batch to Unity
batchsize = 500000
#The full path to the keystore file
keystore = /home/unity/IBM/LogAnalysisTest/wlp/usr/servers/Unity/
keystore/unity.ks
#input data type - LOGS, CONFIGFILES, SUPPORTDOCS
inputType = LOGS
#flush flag:
#true : (default) if the client should send a flush signal to the Generic
 Receiver for the last batch of this file
#false : if no flush signal to be sent upon reaching eod-of-file
flushflag = true
#Other properties (name/value pairs, e.g. middleware = WAS) that you want
 to add to all json records
#These properties need to be appropriately added to the index configuration
```

4. Ensure that the Data Collector client JAR file, datacollector-client.jar, has execute permissions.
5. Use the following command to run the Data Collector client with the correct inputs:

```
<HOME>/ibm-java/bin/java
-jar datacollector-client.jar
```

Results

After the task completes, the log file is indexed and can be searched in the **Search** workspace.

Configuring the Data Collector client to ingest data from remote hosts

If you want to use the Data Collector client to collect data from a remote server and return it to the local machine, you must configure the data collector on the remote host.

Before you begin

You must use the instance of IBMJava Runtime Engine (JRE) 1.7 that is installed by the remote installer. Before you configure the data collector, you must use the remote installer to install at least one instance of IBM Tivoli Monitoring Log File Agent or the EIF Receiver on a remote machine. For more information, see the *Configuring data collection for scalability on multiple remote nodes* topic in the Installation Guide.

About this task

To configure the Data Collector on the remote host, copy the data collector client files from your local version of the data collector files to the remote host.

Procedure

1. Copy the <HOME>/utilities/datacollector-client directory and all the files that are contained in it from the local installation of IBM Operations Analytics - Log Analysis to the remote machine.
2. Add the location of the log and keystore files to the javaDatacollector.properties file in the directory that you copied the data collector to in the previous step. The keystore file is named unity.ks and it is available in the <*Remote_install_dir*>/LogAnalysis/store/ directory on the remote machine. Where <*Remote_install_dir*> is the directory where you installed the remote instance as described in the *Prerequisites* section here.

Results

After you complete the configuration, you must complete the Data Collector configuration. For more information about how to do this, see “Batch loading historic log data with the Data Collector client” on page 262. You must ensure that the remote installation uses the IBMJava Runtime Engine (JRE) 1.7 that is installed by the remote installer. IBM Java Runtime Engine (JRE) 1.7 is stored in the <*Remote_install_dir*>/LogAnalysis/ibm-java/ directory.

Data Collector properties

Before you can use the data collector to stream data or load a batch of historic data, edit the javaDatacollector.props file.

The javaDatacollector.props file is in the <HOME>/IBM/LogAnalysis/ utilitiesdatacollector-client folder.

The logFile, hostname, logpath, and keystore parameters are required.

The userid, password, and keystore parameters are automatically populated with the default values that are created during the installation. If you want, you can change these but you do not need to.

Table 76. Data Collector properties

Parameter	Value
logFile	The full path of the file you want to load.
servletURL	The URL of the Data Collector service.
userid	The user ID for the Data Collector service.
password	The password for the Data Collector service.
datasource	The datasource that you want to use to load data.
timestamp	The time stamp to use if a time stamp is not found in the log file.
batchsize	The number of BYTES of logs sent in one batch. The default value is 500,000.
keystore	The full path to the keystore file.
inputType	The valid input type is LOGS.

Table 76. Data Collector properties (continued)

Parameter	Value
flush flag	If the default true is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to false no flush signal is sent when the end-of-file is reached.

Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent

You can use the IBM Tivoli Monitoring Log File Agent to load batches of historic data for testing and other purposes.

Procedure

1. Copy the log files that you want to load to a temporary directory on the IBM Operations Analytics - Log Analysis server. For example, to upload a batch of log files from an installation of WebSphere Application Server, you copy the SampleSystemOut.log file to the /tmp/logs/ directory.
2. Create a custom data source.
3. Copy the log file to the directory that you specified in the logpath parameter when you created the data source.

Extending storage space available to Apache Solr

You can add more Apache Solr storage directories outside the initial IBM Operations Analytics - Log Analysis Apache Solr installation location if the disk on which Apache Solr was installed reached maximum capacity.

Before you begin

Ensure that the Apache Solr storage directories are present on all Apache Solr servers and are writable.

About this task

Switching to a new Apache Solr directory is not instantaneous. Therefore, it is to monitor the disk usage of your Apache Solr directory to ensure that extra directories are added before the current storage directory reaches maximum capacity.

Procedure

To enable the storage extension capability, complete the following steps.

1. Stop IBM Operations Analytics - Log Analysis with the following command.

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
   ```
2. Open the unitysetup.properties file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF directory.
3. Add the following property to the directory

   ```
   ENABLE_SOLR_RELOCATION=true
   ```
4. Create the following properties file

   ```
   <HOME>/solrConfigs/storageConfig.properties
   ```

   For example,

```
/home/unity/IBM/LogAnalysis/solrConfigs/storageConfig.properties
```

5. Open the `storageConfig.properties` file and add the following property to the file.

   ```
   SOLR_STORAGE_DIR=storage-path-on-solr-nodes
   ```

   For example,

   ```
   SOLR_STORAGE_DIR=/opt/scala/ext_storage
   ```
6. Restart IBM Operations Analytics - Log Analysis with the following command.

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Results

The new IBM Operations Analytics - Log Analysis configuration file enables the specification of custom data storage locations. The new locations are written to when IBM Operations Analytics - Log Analysis crosses the default boundary of 1 day.

Changing the default boundary for creating Apache Solr collections

You can change the default boundary that is associated with extending Apache Solr storage space depending on your business needs.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file.
2. Locate and modify the value of the `COLLECTION_ASYNC_WINDOW` property from the default value of `1d` (1 day).

   Note: The minimum property size is `6h`.

   The boundary size can be specified in minutes (`m`), hours (`h`), or days (`d`).
3. Restart IBM Operations Analytics - Log Analysis with the following command.

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Changing the default password for the Data Collector and EIF Receiver

If you want, you can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics. This is optional.

Changing the default EIF Receiver or Data Collector password

You can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

About this task

After you install IBM Operations Analytics - Log Analysis, the EIF Receiver and the Data Collector are configured to use the default user name and password to connect to IBM Operations Analytics - Log Analysis. The encrypted passwords are defined in the following files:

- Data Collector client is named `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`.
- EIF Receiver is named`<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`.

IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to encrypt and decrypt passwords for your installation, in the following format:

```
password={aes}<Unique_string_of_alphanumeric_characters>
```

For example, the `javaDatacollector.properties` file uses the `unityuser` user ID to access the Data Collector server. In this example, IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to generate the following password:

```
{aes}7DB629EC03AABEC6C4484F160FB23EE8
```

The encrypted password is replicated to the configuration files for the Data Collector and the EIF Receiver.

Procedure

1. To change the default password, use the `unity_securityUtility.sh` command.

   For more information about this command, see "`unity_securityUtility.sh` command" on page 205.
2. Update the configuration files for the Data Collector or the EIF Receiver.
3. Optional: If you want to change the password on remote instances of the EIF Receiver, complete the previous steps and copy the `unity.conf` file from the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory on the local machine to the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory on the remote machine. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder that is used for the specific remote EIF instance.

Example

For example, you want to change the default password for the default user that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis to `myNewPassword`. Complete the following steps:

1. Go to the `IBM/LogAnalysis/utilities` directory.
2. Run the `unity_securityUtility.sh` command as follows:

   ```
    [utilities]$ ./unity_securityUtility.sh encode myNewPassword
    Using keystore file unity.ks
    <HOME>/IBM/LogAnalysis/utilities/../wlp/usr/servers/Unity/
   keystore/unity.ks
    {aes}E6FF5235A9787013DD2725D302F7D08
   ```
3. Copy the AES encrypted password to the relevant configuration files, for example copy it to the Data Collector file. You must copy the complete, encrypted string from the command output, including the `{aes}` prefix. For example:

   ```
   {aes}E6FF5235A9787013DD2725D302F7D088
   ```

`unity_securityUtility.sh` command

You can use the `unity_securityUtility.sh` command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Syntax

The `unity_securityUtility.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
unity_securityUtility.sh encode [textToEncode] [unity.ks]
```

Parameters

The `unity_securityUtility.sh` command has the following parameters:

encode

The encode action returns an AES encrypted version of the text that you enter as the text to encrypt.

[*textToEncode*]
Use the [*textToEncode*] parameter to enter the password that you want to encrypt. If you do not specify a password for this parameter, IBM Operations Analytics - Log Analysis prompts you for one.

[unity.ks]

The `unity.ks` file is the default keystore that is generated automatically during installation. It controls how the password is encrypted and decrypted.

The `unity.ks` file is used to encrypt and decrypt passwords for the following features:

- Java data collector client in the `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties` file.
- EIF Receiver in the `<HOME>/IBM/LogAnalysis/utilities/UnityEIFReceiver/config/unity.conf` file.

For an example of how to use this command, see "Changing the default EIF Receiver or Data Collector password" on page 266.

Installing logstash

Installing logstash on a remote node extends IBM Operations Analytics - Log Analysis functions so it can ingest and perform metadata searches against log data that is acquired by logstash.

logstash 1.4.2 is bundled and installed with IBM Operations Analytics - Log Analysis. You can install logstash on a local host but to improve system performance, install logstash on a remote node.

This document describes the version of logstash that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of logstash might have been published after this version of IBM Operations Analytics - Log Analysis. To download the most up-to-date logstash versions and updated documentation, see https://www.elastic.co/downloads/logstash.

logstash is an open source tool for managing events and logs. It can be used to collect logs, parse them, and send them to another tool such as IBM Operations Analytics - Log Analysis to store them for later use.

The logstash agent is an event pipeline that consists of three parts:

1. Inputs

2. Filters
3. Outputs

Inputs generate events. Filters modify events. Outputs send the event somewhere. For example, events can be sent to storage for future display or search, or to the IBM Operations Analytics - Log Analysis framework. Events can have a type, which is used to trigger certain filters. Tags can be used to specify an order for event processing as well as event routing to specific filters and outputs.

logstash can be used as a "pre-processor" to analyze sources and provide a semi-structured or structured feed to IBM Operations Analytics - Log Analysis for the purposes of searching and potential usage within custom analytics applications.

For more information on logstash events, see the section *the life of an event in logstash* at https://www.elastic.co/guide/en/logstash/current/index.html.

Dependencies

Supported version of logstash and its dependencies.

Supported logstash version

The supported version of logstash is 1.4.2 .

DSV Toolkit requirement

DSV Toolkit v1.1.0.1 or higher for generating IBM Operations Analytics - Log Analysis Insight Packs. The Insight Packs are used to index log records that have been annotated using logstash. You only require the DSV toolkit if you want to use logstash to perform ingestion, splitting, annotating or for when the data being read by logstash is in DSV format. For more information on this user scenario, see “Configuring logstash for rapid annotation and pre-indexing processing” on page 273.

Generic Annotation Insight Pack

Generic Annotation v1.1.0, or v1.1.1 (refresh 1) is recommended for the normalized timestamp splitter function, which recognizes a variety of timestamps.

Installing logstash on a remote node

You can install logstash on a remote node to improve system performance.

Before you begin

Ensure that the SSH user has the correct permissions for installation. For more information on SSH configuration, see “Secure Shell (ssh) configuration for remote logstash” on page 270 in the *Loading and streaming data guide*.

About this task

logstash is processor and system resource intensive. logstash can be installed on the local host but to improve system performance, install logstash on a remote node.

Procedure

1. To install logstash, run the following command:
   ```
   <HOME>/IBM/LogAnalysis/remote_install_tool/install.sh
   ```
2. The installation script installs logstash, and provides options to install the EIF receivers and log file Agent. To select each option, including logstash, select `y` or `Y`.
3. Provide the path to the installation location on the remote host.

Results

logstash is installed in the `<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/` directory. To confirm the installation, logon to the remote node as the configured SSH user and go to the installation location.

Example

The following are example deployments:

logstash example:

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/
```

Output plug-in configuration path:

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/logstash-scala/logstash/
config/logstash-scala.conf
```

Output plug-in jar directory

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/logstash-scala/logstash/outputs
```

Secure Shell (ssh) configuration for remote logstash:

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

The `ssh_config.properties` file is in the <HOME>/IBM/LogAnalysis/remote_install_tool/config directory. Configure the parameter values as outlined in Table 1.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password-based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file-based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

To set up command-line authentication, rename the ssh-config properties file or move the properties file to a new location. By default the configurations are selected from the properties file. If the file is unavailable, the user is prompted for command-line input.

Table 77. ssh_config parameters

Parameter	Value
REMOTE_HOST=	*<REMOTE SERVER IP/FQ HOSTNAME>*
PORT=	*<SSH PORT>* THE DEFAULT VALUE IS 22
USER=	*<SSH_USER>*
PASSWORD=	*<SSH PASSWORD>*

logstash configuration

logstash can be configured as a log file agent to ingest logs from a number of different sources.

About this task

There are two established use cases for using logstash with IBM Operations Analytics - Log Analysis, these are:

- Configuring logstash as an alternative to ITM LFA
- Configuring logstash for rapid annotation and pre-indexing processing

Both use cases are described in this section.

Configuring logstash as an alternative to ITM LFA:

logstash can be used as a log file agent to ingest logs from a number of different sources. It can also support integration with numerous alternative log file agents such as Lumberjack, Minuswell, Beaver, and Syslog.

About this task

Log records are written to the IBM Operations Analytics - Log Analysis that then sends the message to the IBM Operations Analytics - Log Analysis server for annotating and indexing.

Procedure

1. Update the logstash sample configuration file, `logstash/config/logstash-scala.conf`, with your configuration information.
   a. Define the input in the logstash configuration file.

      For example:

      ```
      input {
        file {
         type => "http"
         path => ["/tmp/myhttp.log"]
        }
       }
      ```

      Note: For Windows, the logstash file plug-in requires a drive letter specification for the path, for example:

      ```
      path => ["c:/tmp/myhttp.log"]
      ```

   b. Modify the logstash configuration file to add the `scala` output plug-in.

      The `scala` output plug-in buffers and sends the logstash event to the IBM Operations Analytics - Log Analysis server by using the Log Analysis server ingestion REST API. The logstash configuration file can contain one or more scala output plug-ins. The output plug-ins can be configured to write to different Log Analysis servers or to the same Log Analysis server with a different set of configurations.

Every event that is sent to the scala output plug-in must contain at least the host and path fields. The values of these fields are used by the scala output plug-in to determine the target data source for the event. Any event that does not contain either of these fields is dropped by the output plug-in.

The following are the default parameters, with sample values, for the IBM Operations Analytics - Log Analysis scala output plug-in:

```
output {
  scala {
    scala_url => "https://<la_server>:<port>/Unity/DataCollector"
    scala_user => "<LA_user>"
    scala_password => "<encrypted_pwd>"
    scala_keystore_path => "<install-dir>/LogAnalysis/store/unity.ks"
    batch_size => 500000
    idle_flush_time => 5
    sequential_flush => true
    num_concurrent_writers => 20
    use_structured_api => false
    disk_cache_path => "<install-dir>/LogAnalysis/Logstash/cache-dir"
    scala_fields =>
      {
       "host1@path1,host2@path2"
          => "event_field11,event_field12,...,event_field1N"
       "host3@path3"
          => "event_field21,event_field22,...,event_field2N"
      }
    date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
    log_file => "<install-dir>/LogAnalysis/Logstash/logs/scala_logstash.log"
    log_level => "info"
   }
```

Where:

- **scala_url** is the REST endpoint for the Log Analysis ingestion REST API.
- **scala_user** is the Log Analysis user name.
- **scala_password** is the Log Analysis user password.
- **scala_keystore_path** is the path to the Log Analysis keystore on the file system.
- **batch_size** is the maximum number of bytes that can be buffered for a data source before transmitting to the Log Analysis server. The default is *500000* bytes.

  Note: Significantly decreasing the batch size impacts on throughput. Increasing the batch size requires more heap memory.
- **idle_flush_time** is the maximum time between successive data transmissions for a data source.
- **sequential_flush** defines whether batches for each data source are sent sequentially. It is set to *true* to send the batches sequentially.

  Note: Sequential sending is required when the input contains multi-line records that are combined in an Insight Pack in the Log Analysis server.
- **num_concurrent_writers** is the number of threads that concurrently transmit batches of data to the Log Analysis server.
- **use_structured_api** determines whether data is transmitted to the Log Analysis server in the JSON format. It is set to *true* to transmit data in the JSON format.

Note: The target Log Analysis data source must be associated with a source type that uses the Log Analysis structured API.

- **disk_cache_path** is the path on the file system that temporarily buffers data. The scala output plug-in writes data to this path before transmission. The available disk space under the path must be large enough to store bursts of input data that is not immediately handled by the Log Analysis server.
- **scala_fields** is the map that specifies the names of fields that must be retrieved from the incoming logstash event and transmitted to the Log Analysis server. The keys for the map are a comma-separated list of host and path names that correspond to a Log Analysis data source.

  The scala plug-in extracts the host and path fields from each event before consulting the **scala_fields** map for a host and path combination entry. If there is an entry with field names, the scala plug-in extracts the corresponding field values from the event. The values are transmitted to the Log Analysis server. If the host and path entries are not in the **scala_fields** map, the scala plug-in extracts the contents of the message field from the event and transmits it to the Log Analysis server.
- **date_format_string** is the string value that all fields are transformed to before transmission to the Log Analysis server. The scala plug-in uses the **date_format_string** parameter to convert date values to the appropriate string value.
- **log_file** is the file that is used for logging information from the scala output plug-in.
- **log_level** is the level of logging information. The supported levels are fatal, error, warn, info, and debug.

2. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

   Ensure that the **File Path** matches the path that is specified in the logstash configuration file, logstash-scala.conf.

   Ensure that the **Type** matches the type of log file that is being ingested, for example **DB2Diag**.

   For example, if you specified /tmp/myhttp.log as an input file, then create a custom data source with path set to /tmp/myhttp.log.

What to do next

Start logstash as described in Starting logstash

Configuring logstash for rapid annotation and pre-indexing processing:

logstash can be used to split log records and do basic annotation. For log types not currently supported by IBM Operations Analytics - Log Analysis, this is an alternate approach to writing AQL to annotate log files.

About this task

logstash includes a broad list of filtering, manipulation, and processing capabilities, for example, the grok filter can be used to parse text into structured data. It allows you to match text without the need to master regular expressions. There are approximately 120 grok patterns shipped by default, though you can add more. It also includes patterns for known log file formats, such as Apache's combined access log format.

In this scenario, logstash is basically used as the splitter/annotator of the log file by leveraging the grok filter. The `scala_custom_eif` output plugin sends a single log record to the IBM Operations Analytics - Log Analysis EIF Receiver, with the annotations in a delimiter separated value (DSV) format. Then, using the DSV Toolkit, the user must create and install an insight pack that matches the DSV format so that IBM Operations Analytics - Log Analysis can index the annotations. Please follow these steps:

Procedure

1. Update the logstash sample configuration file, `logstash/config/logstash-scala.conf`, with your configuration information.
   a. Define the input in the logstash configuration file.

      For example:

      ```
      input {
        file {
         type => "apache"
         path => ["/tmp/logs/myapache.log"]
        }
       }
      ```

      Note: For Windows, the logstash file plugin requires a drive letter specification for the path, for example:

      ```
      path => ["c:/tmp/myapache.log"]
      ```

   b. Modify the logstash configuration file to add the `scala_custom_eif` output plugin.
   c. Add a filter or filters to the logstash configuration file to identify the pattern of the log file format. This also creates the annotations. To trigger the filter, the type must match the input type.

      For example:

      ```
      filter {
          if [type] == "http" {
              grok {
                  match => ["message", "%{IP:client} %{WORD:method}
      %{URIPATHPARAM:request}  %{NUMBER:bytes} %{NUMBER:duration}"]
                            }
          }
      }
      ```

      In this example, the fields client, method, request, bytes, and duration are annotated by the pattern. However, only the fields client, method and request are sent to IBM Operations Analytics - Log Analysis. Thus, those are the only three annotations that can be included in the index configuration. The output module sends the event text in DSV format as:

      ```
      "client", "method", "request"
      ```

      The user can also use one of the many predefined grok log format pattern such as:

      ```
      filter {
          if [type] == "apache" {
              grok {
                  match     => ["message", "%{COMBINEDAPACHELOG}"]
              }
          }
      }
      ```

2. Create an IBM Operations Analytics - Log Analysis DSV-generated Insight Pack in order to index the annotated data in IBM Operations Analytics - Log Analysis.

The `lsartifact/dsvProperties` directory contains a sample property file that can be used to generate an Insight Pack that ingests delimiter separated log records that are already formatted for Apache combined access log files. Use the DSV toolkit, which is available at `<HOME>/IBM/LogAnalysis/unity_content/tools`, to generate an Insight Pack from the DSV properties file. This means the user must configure the logstash configuration file, `/lstoolkit/logstash/config/logstash-scala.conf`, with the appropriate grok filter to enable the IBM Operations Analytics - Log Analysis output plugin to generate the comma delimited logs. For example, uncomment the apache grok filter in the `logstash-scala.conf` file and generate an Insight Pack using `ApacheDSV.properties` with the DSV tooling script. The `scala` plugin will generate a comma delimited event based on the grok filter that can be ingested (annotated and split) by the generated Insight Pack.

Note: The path to `logstash-scala.conf` is dependent on where you copied the `lstoolkit` directory on the logstash server (see step 3 of Installing the logstash Integration Toolkit).

3. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

   Ensure that the **File Path** matches the path that is specified in the logstash configuration file, `logstash-scala.conf`.

   Ensure that the **Type** matches the type of log file that is being ingested, for example **DB2Diag**.

   For example, if you specified `/tmp/myhttp.log` as an input file, then create a custom data source with path set to `/tmp/myhttp.log`.

What to do next

Start logstash. For more information on starting logstash, see "logstash operations" on page 276 in the *Installing logstash* section of the *Loading and streaming data* guide.

Example - Annotating Combined Apache log files:

Using logstash to annotate Apache log files.

Procedure

1. Edit your logstash configuration file. A sample is provided in `logstash-scala.conf`.
   a. In the input section, specify the Apache log file to be monitored.

```
input {
  file {
   type => "apache"
   path => ["/tmp/apache.log"]
  }
 }
```

   b. Add the logstash grok filter with the predefined `COMBINEDAPACHELOG` pattern to annotate the Apache log files.

   For example:

```
filter {
 if [type] == "apache" {
  grok {
     match => ["message", "%{COMBINEDAPACHELOG}"]
  }
 }
}
```

The COMBINEDAPACHELOG pattern is defined as:

```
COMBINEDAPACHELOG %{IPORHOST:clientip} %{USER:ident} %{USER:auth}
\[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}
(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response}
(?:%{NUMBER:bytes}|-) %{QS:referrer} %{QS:agent}
```

For more information about Apache log files, see http://httpd.apache.org/docs/2.4/logs.html.

The logstash event contains annotations for clientip, ident, auth, timestamp, verb, request, httpversion, rawrequest, response, bytes, referrer, and agent.

c. In the output section of the configuration file, specify the IBM Operations Analytics - Log Analysis output plug-in.

2. The logstash Integration Toolkit provides a properties file, `lsartifact/dsvProperties/ApacheDSV.properties`, which can be used with the DSV Toolkit to create an Apache Insight Pack. Edit this properties file to configure information about your IBM Operations Analytics - Log Analysis server:

   ```
   [SCALA_server]
    username: unityadmin
    password: unityadmin
    scalaHome: $HOME/IBM/LogAnalysis
   ```

3. Use the `dsvGen.py` script that is provided with the DSV Toolkit to generate and deploy the Apache Insight Pack:

   ```
   python dsvGen.py <path>/ApacheDSV.properties -d
   ```

4. In the IBM Operations Analytics - Log Analysis Administrative Settings UI, create a data source, which has the Apache source type that is created by the DSV toolkit in step 4, in your logstash configuration file.
5. Start logstash with the configuration file, and start ingesting Apache log files.

logstash operations

You can use the `logstash-util` script to start, stop, restart, or provide the status of logstash.

About this task

You can use the `logstash-util` script for logstash process lifecycle management.

Procedure

1. To start, stop, restart, or provide the status of logstash, run the following command:

   ```
   <install-dir>/LogAnalysis/utilities/logstash-util.sh start| stop| restart| status
   ```

   where *<install-dir>* is the name of the logstash installation location.
2. To confirm that logstash is running, run the `logstash-util` script and use the `status` option. The `status` option also displays the logstash process identifier.

logstash best practices

Best practices for logstash based on information from their user community.

For performance reasons it is recommend that logstash be installed on a different server than IBM Operations Analytics - Log Analysis. logstash is processor, memory, and disk intensive if the annotation and indexing functions are utilized.

Users who have memory constraints do not use logstash as a forwarding agent. They do not install logstash on the end client servers. They use other applications such as rsyslog to forward logs to a central server with logstash. See https://support.shotgunsoftware.com/entries/23163863-Installing-logstash-Central-Server for an example configuration.

Users with logstash at the end client who are concerned about performance have used applications such as Redis to forward logs to a central server with logstash. See the following for configuration of Redis http://www.linux-magazine.com/Online/Features/Consolidating-Logs-with-logstash .

To fine tune logstash, especially the startup time, users can tweak Java's minimum and maximum heap size with the -Xms and -Xmx flags. The -Xms parameter is the initial Java memory heap size when the JVM is started, and the -Xmx parameter is the maximum heap size.

References

Links for more information on the logstash application.

logstash website:
  http://logstash.net

Getting Started with logstash Guide:
  http://logstash.net/docs/1.4.2/tutorials/getting-started-with-logstash

logstash Download:
  http://logstash.net (Click download button)

The logstash Book:
  http://www.logstashbook.com/

IBM Operations Analytics - Log Analysis wiki:
  http://www.ibm.com/developerworks/servicemanagement/ioa/log/downloads.html

IBM Operations Analytics - Log Analysis wiki: Logstash Toolkit Resources:
  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Logstash%20Toolkit%20Resources

Known issues

Known issues when using logstash with IBM Operations Analytics - Log Analysis.

There are a number of known issues and their workarounds described in this section. To get the latest information on any issues or workarounds, please consult the IBM Operations Analytics - Log Analysis wiki:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Welcome

Could not load FFI Provider:

Starting logstash fails with the Ruby exception “Could not load FFI Provider”.

Symptoms
The Ruby exception “Could not load FFI Provider”.

Causes
The most common cause of this error is that `/tmp` is mounted with the **`noexec`** flag.

Resolving the problem
You can resolve this either by:

- Making /tmp mounted without the **noexec** flag
- Edit the startlogstash-scala script and amend the start command as follows:

  ```
  LSCMD="$MYJAVA -jar -Djava.io.tmpdir=</some/tmp/dir> $LSJAR agent
  --pluginpath $PLUGPATH -f $CONF"
  ```

  Where </some/tmp/dir> is a temporary directory.

Duplication of log records on the SCALA server:

On occasion, when the logstash agent is re-started, and the log file being monitored is updated (for example, via a streaming log), logstash will ingest the entire file again rather than restarting from where it stopped monitoring.

Symptoms
The problem results in a duplication of log records on the SCALA server.

Causes
Several problems have been reported on the logstash forum (https://logstash.jira.com/secure/Dashboard.jspa) that its sincedb pointer (which tracks the last monitored position in the log file) sometimes is not updated correctly. In addition, using control-C to terminate the logstash agent does not always kill logstash. The result is a "phantom" logstash agent that is still monitoring log files. This can also result in duplicate log records.

Resolving the problem

1. A workaround to avoid duplicate log records after restarting logstash is to set the **sincedb_path** parameter in the file plugin to /dev/null, thereby telling logstash to ignore tracking the last-monitored file position, and always start monitoring from the end of the file. However, this will result in logstash ignoring any updates to the log file while the logstash agent is down. For example, in logstash-scala.conf, update:

   ```
   input {
       file {
           type => "apache"
           path => ["/tmp/logs/myapache.log"]
           sincedb_path => "/dev/null"
       }
   }
   ```

   Before re-starting logstash after making these configuration changes, you may also want to clean up any sincedb databases that were already created. By default, the sincedb database is stored in the directory $HOME, and have filenames starting with ".sincedb_".
2. When terminating the logstash agent using control-C, verify that the logstash java process was actually terminated. You can use the following command to see if logstash is still running:

   ```
   ps -ef | grep logstash
   ```

Logs do not appear in the Search UI:

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Symptoms
Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Causes

Log records ingested by logstash are forwarded to the IBM Operations Analytics - Log Analysis server for splitting and annotating, and indexing. If the IBM Operations Analytics - Log Analysis server goes down during this process, it is possible to lose some log records.

Loading and streaming data

Before you can perform a search on log or other data, you must first load the data into IBM Operations Analytics - Log Analysis. When the file is loaded the data is indexed and is then available to be searched.

There are two main scenarios for loading data:

- Batch loading historic data. For example, you may want to ingest historic log data in a single batch for analysis or for testing.
- Streaming data from a monitored application. You may want to load data that is streamed from a local or remote server.

You can load or stream data from local or remote servers. However, each tool is designed for a particular scenario. This is explained in the *Intended uses of data loading components* table. IBM Operations Analytics - Log Analysis is installed with an internal version of the IBM Tivoli Monitoring Log File Agent. However, IBM Operations Analytics - Log Analysis can also load data from a separate installation of the IBM Tivoli Monitoring Log File Agent, known as an external IBM Tivoli Monitoring Log File Agent.

Table 78. Intended uses of data loading components

	Load batch of historic data		**Stream data**	
Component	**Local**	**Remote**	**Local**	**Remote**
Data Collector client	Yes	Yes	No	No
Internal IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
External IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
logstash	No	No	No	Yes
Generic Receiver	Yes	Yes	No	No

Note: You must create a Data Source before you configure data loading. For information about creating a Data Source, see the *Administering IBM Operations Analytics - Log Analysis* section of the Information Center. For an overview of the process that you must follow to configure and use IBM Operations Analytics - Log Analysis, see the *Steps to get started with IBM Operations Analytics - Log Analysis* topic in the *Overview of IBM Operations Analytics - Log Analysis* section of the Information Center.
You can load log data into IBM Operations Analytics - Log Analysis using a number of different methods:

Data Collector client

Use the Data Collector client to ingest data in batch mode. This is the easiest method if you want to ingest a large log file for historic analysis if you want to test your IBM Operations Analytics - Log Analysis configuration before attempting the more complex IBM Tivoli Monitoring

Log File Agent configuration. The Data Collector client is not designed for ingesting data from remote sources. If you want to ingest a batch of historical data from a remote source, use the IBM Tivoli Monitoring Log File Agent.

For a video that demonstrates how to batch upload a WebSphere Application Server or DB2 file using the Data Collector client, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Videos. For information about batch uploading alternative log file types such as Oracle alert logs, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Batch%20uploading%20Oracle%20Alert%20logs.

IBM Tivoli Monitoring Log File Agent

Use the IBM Tivoli Monitoring Log File Agent for scenarios where you want to stream log data from your production environment or to stream data from a remote server.

For a video that demonstrates how to upload a WebSphere Application Server or DB2 file using the IBM Tivoli Monitoring Log File Agent, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Videos.

logstash

logstash can be used as a method to collect and load data into IBM Operations Analytics - Log Analysis using the logstash Integration Toolkit. For information about the logstash Integration Toolkit, including information about how to download and install it, see logstash Integration Toolkit.

Generic Receiver

Use the Generic Receiver to load data from the REST interface into IBM Operations Analytics - Log Analysis.

You cannot use IBM Operations Analytics - Log Analysis to index log records that contain non-ASCII characters. If your log records contain non-ASCII characters, the records are not added when you use the IBM Tivoli Monitoring Log File Agent or the Data Collector client. When you use the Data Collector client errors that relate to non-ASCII characters are added to the Generic Receiver log.

Example scenarios

The following table outlines a number of example scenarios to help illustrate how you use the different components for different scenarios.

Table 79. Example data loading scenarios

Example	Component
I want to load a batch of historic log data to test the environment.	Data Collector client
I want to monitor an application on a remote server.	IBM Tivoli Monitoring Log File Agent
I want to use logstash to monitor log files on a remote server.	logstash
I want to load a batch of historic log data in the JSON format.	Generic Receiver

Supported operating systems

The supported operating systems that you can install IBM Operations Analytics - Log Analysis are listed in the *Installing* Guide. In addition to these, you also need to know what operating systems are supported by the data streaming and loading scenarios. For example, if you want to use the internal to stream data from a remote source, you need to know the supported operating systems.

Table 80. Supported operating systems for data loading

Scenario	**Feature**	**Supported operating systems**
Use the Data Collector to load a batch of historic data	Data Collector	• Red Hat Enterprise Linux Server Edition Version 5 or Version 6 (64 bit) • SUSE Linux Enterprise Server 11 (64 bit)
Use the internal IBM Tivoli Monitoring Log File Agent to stream data	Internal IBM Tivoli Monitoring Log File Agent	See the *Requirements for the monitoring agent* topic in the documentation for your version of IBM Tivoli Monitoring at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring
Use an external IBM Tivoli Monitoring Log File Agent to stream data	External IBM Tivoli Monitoring Log File Agent	See the *Requirements for the monitoring agent* topic in the documentation for IBM Tivoli Monitoring 6.2.3.1 at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring

Configuring data streaming

Before you can stream data, you must configure the tools that you use to send the data to IBM Operations Analytics - Log Analysis.

IBM Tivoli Monitoring Log File Agent configuration scenarios

You can use the internal IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis or you can use an external IBM Tivoli Monitoring Log File Agent to stream data from local or remote servers.

You can also use the IBM Tivoli Monitoring Log File Agent to upload a batch of historic data. For more information, see “Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent” on page 265.

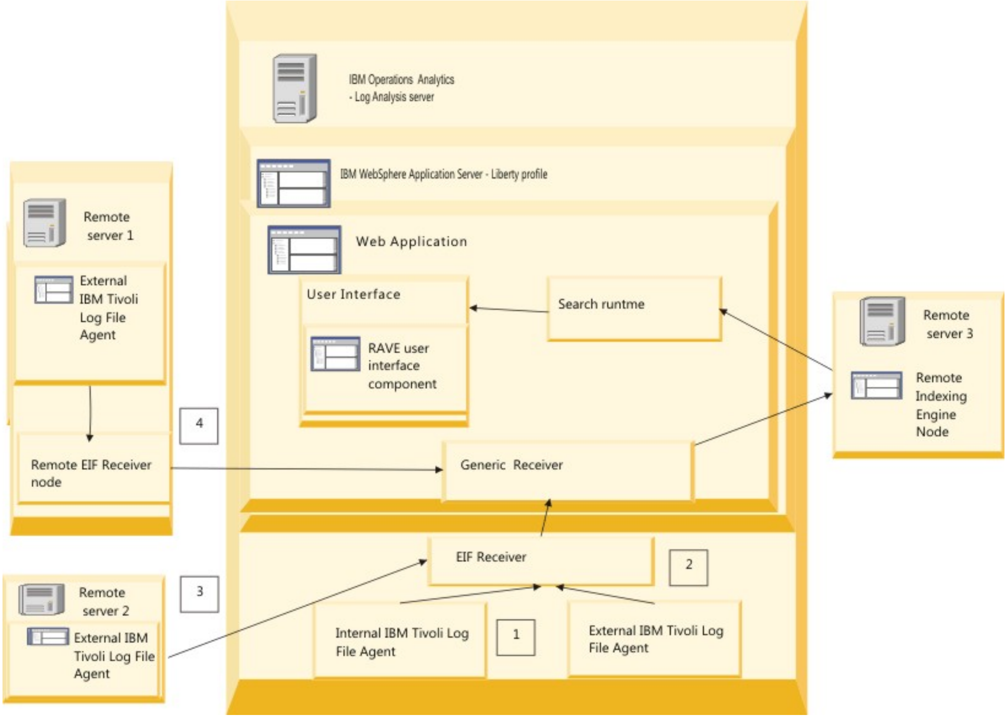
You can integrate the IBM Tivoli Monitoring Log File Agent with IBM Operations Analytics - Log Analysis in two ways.

You can use it with the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis. This is known as the internal IBM Tivoli Monitoring Log File Agent.

You can also use it with an IBM Tivoli Monitoring Log File Agent that has been installed separately as part of another installation.

You can use local and remote versions of both types of IBM Tivoli Monitoring Log File Agent.

The following graphic illustrates these possibilities:


IBM Operations Analytics - Log Analysis server
IBM WebSphere Application Server - Liberty profile
Web Application
User Interface
Search runtme
RAVE user interface component
Generic Receiver
EIF Receiver
Internal IBM Tivoli Log File Agent
External IBM Tivoli Log File Agent
1
2
3
4
Remote server 1
External IBM Tivoli Log File Agent
Remote EIF Receiver node
Remote server 2
External IBM Tivoli Log File Agent
Remote server 3
Remote Indexing Engine Node

The following possible scenarios are illustrated in the graphic:

1. Internal IBM Tivoli Monitoring Log File Agent on a local server
In this scenario, you use the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis to load data from the local installation of IBM Tivoli Monitoring to IBM Operations Analytics - Log Analysis.

2. External IBM Tivoli Monitoring Log File Agent on a local server
In this scenario, you use a version of the IBM Tivoli Monitoring Log File Agent that was not installed with IBM Operations Analytics - Log Analysis but that is installed on the same server as IBM Operations Analytics - Log Analysis.

3. External IBM Tivoli Monitoring Log File Agent on a remote server
In this scenario, you use an installation of an external IBM Tivoli Monitoring Log File Agent to push data to IBM Tivoli Monitoring Log File Agent. To facilitate this integration, you modify the properties of the IBM Tivoli Monitoring Log File Agent.

4. Remote instance of the internal IBM Tivoli Monitoring Log File Agent

In this scenario, you use a the remote installer tool to install a remote instance of the internal IBM Tivoli Monitoring Log File Agent.

The following table summarizes the different configurations required for the scenarios.

Table 81. Configuration for data streaming scenarios

Data streaming scenario	IBM Tivoli Monitoring Log File Agent type	Log file location	Required parameters in `.conf` file
1	Internal and local	Local	`Datasources`
2	Internal and remote. You use the remote installer to create the remote instance.	Remote	`Datasources`, `ServerLocation`, `ServerPort`, `BufEvtMaxSize`.
3	Local and external	Local	`Datasources`
4	Remote and external	Remote	`Datasources`, `SshAuthType`, `SshHostList`, `SshPassword`, `SshPort`, `SshPrivKeyfile`, `SshPubKeyfile`, `SshUserid`.

Configuring IBM Tivoli Monitoring Log File Agents for use with IBM Operations Analytics - Log Analysis

You can configure IBM Tivoli Monitoring Log File Agents to start IBM Operations Analytics - Log Analysis.

About this task

For more information about how configuration and format files are used, see "IBM Operations Analytics - Log Analysis configuration and format files" on page 234.

For more information about the required parameters in the configuration file, see "Configuration file parameters" on page 202.

CAUTION:

You cannot use non-ASCII characters in the installation path. The installation path cannot exceed 80 characters.

For more information, about this and about how to configure the monitoring agent in step 3 see:

http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.2/com.ibm.itm.doc_6.3fp2/install/unixconfig_ma.htm?lang=en

Procedure

1. To configure IBM Tivoli Monitoring Log File Agent, run the command:

```
./itmcmd config -A pc
```

where pc is the product code for your agent. For example: ./itmcmd config –A lo.

2. You are prompted to supply the following information:

   Enter instance name (default is:):
   Enter the instance name. For example, *rhelagent*.

   Conf file (default is:):
   Enter the configuration file path. For example, /unity/IBM/ITM/config/lo/.

   Format File (default is:):
   Enter the format file path. For example, /unity/IBM/ITM/config/lo/.

   Note: All fields must be completed. Blank fields cause IBM Tivoli Monitoring Log File Agent to fail.

3. Where prompted, provide the monitoring agent configuration information.
4. To start the IBM Tivoli Monitoring Log File Agent, run the command

   ```
   ./itmcmd agent  -o instance name start lo
   ```

Configuring IBM(r) Tivoli(r) Monitoring Log File Agents

Before you use the IBM Tivoli Monitoring Log File Agent, you may want to modify the configuration and format files.

About this task

For more information about how configuration and format files are used, see "IBM Operations Analytics - Log Analysis configuration and format files" on page 234.

For more information about the required parameters in the configuration file, see "Configuration file parameters" on page 202.

Procedure

1. Open the configuration file that you want to use.
2. Define the required parameters in the configuration file. The required parameters are different depending on the data loading scenario.
   - If you want to stream data from a local server, specify the data sources in the DataSources parameter.
   - If you want to push data from a remote directory, you must specify values for the Datasources, ServerLocation, ServerPort, and BufEvtMaxSize parameter.
   - If you want to use an external IBM Tivoli Monitoring Log File Agent that is not installed as part of IBM Operations Analytics - Log Analysis, you must specify values for the Datasources, SshAuthType, SshHostList, SshPassword, SshPort, SshPrivKeyfile, SshPubKeyfile, and SshUserid parameters.
3. Define the format file as required.
4. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the .fmt file for each rotating log type to allow for the appropriate name change. Open the .fmt file, and edit the line:

   ```
   -file FILENAME
   ```

   to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the FILENAME must read:

   ```
   -file SystemOut*.log
   ```

5. Save your changes.

Example

For example:

```
===============
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PASSWORD
SshPassword=<password>

====================
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PUBLICKEY
SshPrivKeyfile = <SshUserid_Private_Key_File_Path>
(Or)
SshPubKeyfile = <SshUserid_Private_Key_File_Path>

=====================
```

where `<password>` is the password that you want to use.

<SshUserid_Private_Key_File_Path> is the full path for the file that contains the private key of the user that is specified in the SshUserid user. For example, if you save the password to a file called `password.txt` in the `<HOME>/utilities` directory, the full path is as follows:

```
SshPrivKeyfile = <HOME>/utilities/password.txt
```

Configuring IBM Tivoli Monitoring Log File Agent subnodes

Create an IBM Tivoli Monitoring Log File Agent subnode to group an explicit set of configurations that the IBM Tivoli Monitoring Log File Agent uses to identify and process a log event.

About this task

The subnode consists of a format (`.fmt`) file and a configuration (`.conf`) file. A single instance of the IBM Tivoli Monitoring Log File Agent can have multiple subnodes. Each subnode behaves like a single thread running in the same instance of the IBM Tivoli Monitoring Log File Agent.

You can create subnodes for the following use cases:

Improve performance by making generic format settings more specific
To improve overall performance, you can create specific configurations to replace more generic ones. For example, you can specify the same regular expression (REGEX) in a generic `.fmt` file to parse both WebSphere Application Server (WAS) and DB2 log files. However as the content of the log files differs, this is inefficient. To improve performance, replace the single `.fmt` files with 2 new files containing 2 specific REGEXs for WAS and DB2 in 2 new subnodes.

Improve performance by making generic configuration settings more specific
Similarly, you can improve performance by replacing generic configurations with more specific ones. For example, you can specify the same roll over behaviour in a generic `.conf` to process both WAS and DB2 log files. However as the roll over behaviour in the log files differs, this

configuration results in some of the logs not being processed correctly and is inefficient. To improve performance, replace the single `.conf` with 2 new files in 2 new subnodes.

Improve performance for many data sources
If you use a large number of data sources to monitor the log events, you can create subnodes to spread the workload.

Remote monitoring with IBM Tivoli Monitoring Log File Agent 6.3
With IBM Tivoli Monitoring Log File Agent 6.3, you can modify the `.conf` file to monitor logs from multiple remote sources. However, the user credentials may not be the same for the remote machines. You can only maintain 1 set of user credentials in the IBM Tivoli Monitoring Log File Agent configuration file. In this case, you create multiple subnodes with different user credentials in each. This allows you to monitor multiple remote sources from a single IBM Tivoli Monitoring Log File Agent node with multiple subnodes.

There is also a limitation on the naming of subnodes. For more information, see "Character limits for IBM Tivoli Monitoring Log File Agent subnodes names" on page 232.

Procedure

1. Go to the directory where the IBM Tivoli Monitoring Log File Agent is installed. For example, if you are using the internal IBM Tivoli Monitoring Log File Agent that is delivered with IBM Operations Analytics - Log Analysis, the directory is `<Add_path>`.
2. To open IBM Tivoli Monitoring Log File Agent configuration window, run the following command:
   ```
   bin/CandleManage
   ```
3. Right click on the **Tivoli Log File Agent** service and click **Configure**.
4. Click on the instance that you want to configure and click **OK**. The **Configure Tivoli Log File Agent** window is displayed.
5. On the **Log File Adapter Configuration** tab, ensure that the **Conf file** and **Format File** fields are blank.
6. Click on the **Log File Adapter Global Settings** tab and note the directory that is specified in the **Configuration file autodiscovery directory**. This is the directory where you will save the subnode configuration files. Click **OK**.
7. In the subsequent window, you can ignore the other changes and click **Save** to save your changes.
8. Enter the root user password when prompted to implement your changes.
9. Copy the subnode configuration files to the directory that you noted in step 6. The IBM Tivoli Monitoring Log File Agent automatically detects the changes. You do not need to restart the IBM Tivoli Monitoring Log File Agent instance.

Results

The procedure describes how to configure subnodes in the IBM Tivoli Monitoring Log File Agent UI. You can also use the command line. To use the command line to configure the subnodes:

1. Go to the directory where the IBM Tivoli Monitoring Log File Agent is installed.
2. Run the following command:
   ```
   itmcmd config -A lo
   ```

3. Follow the onscreen instructions.
4. Specify the configuration file autodiscovery directory.
5. Complete the configuration.
6. Save the subnode configuration file to the directory that you specified in step 4.

Character limits for IBM Tivoli Monitoring Log File Agent subnodes names:

When you name a subnode, ensure that you are aware of the character and naming limitations.

32 character limitation

The IBM Tivoli Monitoring Log File Agent uses msn to name and identify the subnode. IBM Tivoli Monitoring limits the length of this name to 32 characters. The limit includes the identifier, the dash, and the semi-colon. This leaves 28 new characters for the host name, subnode and configuration file name.

The subnode name is specified in the following format:

```
LO:<Hostname>_<Subnode>-<Conffilename>
```

where `LO` is an identifier that is assigned to all subnodes. <*Hostname*> is the host name of the machine where the subnode is installed. <*Subnode*> is the name of the subnode.<*Conffilename*> is the name of the subnode configuration file.

For example:

```
LO:nc1234567890_WASInsightPack-lfawas
```

However, IBM Tivoli Monitoring limits the length of this name to 32 characters. The example name is 35 characters long. The limit includes the identifier, the dash, and the semi-colon, leaving 28 characters for the host name, subnode and configuration file name. To work around this limitation, IBM Tivoli Monitoring renames the subnode as:

```
LO:nc1234567890_WASInsightPack-l
```

This name is 32 characters long. The host name uses 12 characters. The subnode uses 14 characters.

This limitation can cause an issue if you use similar names for the configuration files. For example, after you name the first subnode, you create another subnode called:

```
LO:nc1234567890_WASInsightPack-lfawas2
```

IBM Tivoli Monitoring renames this subnode as:

```
LO:nc1234567890_WASInsightPack-l
```

As you can see, due to the truncation, both subnodes now have the same name, meaning that the IBM Tivoli Monitoring Log File Agent will not detect the new configuration.

Increasing the limit

The host name is used for integrations with Tivoli Endpoint Manager, where it helps you to identify subnodes. However, this is not required for IBM Operations

Analytics - Log Analysis. You remove the host name from the default naming convention so that you can use all 28 characters for the configuration file name.

To change the host name setting:

1. Stop the IBM Tivoli Monitoring Log File Agent.
2. Open the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/lo_default_workload_instance.conf` file.
3. Change the default value for the following property from `Y` (Yes) to `N` (No):
   ```
   CDP_DP_USE_HOSTNAME_IN_SUBNODE_MSN='N'
   ```
4. Save the updated file.
5. Restart the IBM Tivoli Monitoring Log File Agent.

After you change this configuration setting, the subnode name no longer includes the host name. For example, the subnodes in the previous example are now named `LO:WASInsightPack-lfawas` and `LO:WASInsightPack-lfawas2`.

Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data

If you use a IBM Tivoli Monitoring Log File Agent that is installed on a remote server to stream data to IBM Operations Analytics - Log Analysis, you must update the configuration and format files for the IBM Tivoli Monitoring Log File Agent.

Before you begin

You must create a custom data source before you configure data loading. For information about creating a data source, see “Data Source creation” on page 339.

About this task

You can use the configuration files in the `Unity_HOME/IBM-LFA-6.30/config/lo` directory as a basis for the configuration files on your remote server. However, you must ensure that the configuration files that you create:

- contain a line separator between each property that you define in the `.conf` file.
- use the `.conf` file extension and that the format file uses the `.fmt` extension.

To enable the IBM Tivoli Monitoring Log File Agent configuration, complete the following procedure:

Procedure

1. Specify a value or values for the `DataSources` property. If you have multiple locations, you can list the locations and use a comma as a separator. Ensure that you do not leave any spaces. For example, you specify the following values to represent 2 data sources:
   ```
   DataSources=/opt/IBM/WAS1/logs/SystemOut.log,/opt/IBM/WAS2/logs/SystemOut.log
   ```
   When you create a data source for a remote machine, you must enter the correct version of the host name for that machine. To find the correct host name, run the following command on the remote machine:
   ```
   uname -a
   ```
   Enter the name that is returned by this command in the host name parameter for the data source.

2. Specify the server location for the EIF receiver server. For example, for a server that is located at 111.222.333.444, specify the following value:

   ```
   ServerLocation=111.222.333.444
   ```

3. Specify the port that the EIF receiver uses. For example:

   ```
   ServerPort=5529
   ```

4. Specify the `BufEvtPath` for the LFA. The cached events are stored in this file. For example:

   ```
   BufEvtPath=/opt/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache
   ```

5. Specify the maximum buffer size for the LFA. This is the maximum size that the cache is allowed to be. If the cache is full, events are dropped and performance can decline. The value that you enter here is in kilobytes. For example:

   ```
   BufEvtMaxSize=102400
   ```

6. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the `.fmt` file for each rotating log type to allow for the appropriate name change. Open the `.fmt` file, and edit the line:

   ```
   -file FILENAME
   ```

   to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the `FILENAME` must read:

   ```
   -file SystemOut*.log
   ```

7. Save your changes.

What to do next

Allow time for the log data to be ingested and then search for a value contained in your log file to validate that the configuration has succeeded.

IBM Operations Analytics - Log Analysis configuration and format files

If you use an internal or external IBM Tivoli Monitoring Log File Agent, you can edit the configuration and property files to suit your specific installation.

The IBM Tivoli Monitoring Log File Agent configuration for a particular data source is defined in the following files:

- A `<name>.conf` file that contains the properties that are used by the IBM Tivoli Monitoring Log File Agent for processing the log files.
- A `<name>.fmt` file that contains an expression and format that is used by the agent to identify matching log file records and to identify the properties to include in the Event Integration Format (EIF) record. The EIF is sent from the agent to the receiving server. The receiving server is the server where the IBM Operations Analytics - Log Analysis server is installed. The `<name>.fmt` file uses a regular expression to determine matching records in the log file and to send each matching record to the IBM Operations Analytics - Log Analysis server in an EIF event.

If you want to use the IBM Tivoli Monitoring Log File Agent to send your log files to IBM Operations Analytics - Log Analysis server, you must customize the regular expression and define your own stanza in the `<name>.fmt` file to capture the log records that are to be sent. The event record format must include the host name, file name, log path, and text message. The IBM Operations Analytics - Log

Analysis server uses these values to process the logs. For more information about the IBM Tivoli 6.3 Log File Agent and the configuration files and properties, see Tivoli Log File Agent User's Guide.

The file names must be identical for both files. For example, `WASContentPack_v1.1.0-lfawas.conf` and `WASContentPack_v1.1.0-lfawas.fmt`.

After you modify the configuration files as required, you use the IBM Tivoli Monitoring Log File Agent to load the data into IBM Operations Analytics. For a general description of how to do this, see "Using the IBM Tivoli Monitoring Log File Agent" on page 237

If you use an external instance of the IBM Tivoli Monitoring Log File Agent to load data into the IBM Operations Analytics - Log Analysis server, you must install the configuration files into the agent. This configuration ensures that the agent knows where the log files for a data source are located, how to process the records in the log file, and the server to which records are sent.

LFA configuration file examples

The following example shows the files that are installed as part of the WebSphere Insight Pack that is included as standard with IBM Operations Analytics - Log Analysis.

The `WASContentPack_v1.1.0-lfawas.conf` file contains many properties, including the following examples:

```
# Files to monitor.  The single file /tmp/regextest.log, or any file like
/tmp/foo-1.log or /tmp/foo-a.log.
      LogSources=/home/unityadm/IBM/LogAnalysis/logsources
   /WASInsightPack/*

      # Our EIF receiver host and port.
      ServerLocation=<EIF Receiver host name>
      ServerPort=5529
```

The `WASContentPack_v1.1.0-lfawas.fmt` file contains the following regular expression that matches any record within a monitored log file. In this example, the regular expression matches all the log records in the file and to the Operations Analytics server as an EIF event. The EIF event contains the host name where the agent is running, the file name of the log file, the log file path of the log file, and the log file record itself.

```
 // Matches records for any Log file:
     //

     REGEX AllRecords
     (.*)
     hostname LABEL
     -file FILENAME
     logpath PRINTF("%s",file)
     text $1
     END
```

Configuration file parameters

The IBM Tivoli Monitoring Log File Agent uses the information that is specified in the configuration file to process log file information.

Table 1 explains that parameters that you can modify in this file.

Table 82. Parameter summary

Parameter	Description
`DataSources`	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA conf file, the directory you specify must not contain any subdirectories.
`SshAuthType`	You must set this value to either `PASSWORD` or `PUBLICKEY`. If you set this value to `PASSWORD`, IBM Operations Analytics - Log Analysis uses the value that is entered for `SshPassword` as the password for Secure Shell (SSH) authentication with all remote systems. If you set this value to `PUBLICKEY`, IBM Operations Analytics - Log Analysis uses the value that is entered for `SshPassword` as pass phrase that controls access to the private key file.
`SshHostList`	You use the `SshHostList` value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system. If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.
`SshPassword`	If the value of the `SshAuthType` parameter is `PASSWORD`, enter the account password for the user that is specified in the `SshUserid` parameter as the value for the `SshPassword` parameter. If the value of the `SshAuthType` parameter is `PUBLICKEY`, enter the pass phrase that decrypts the private key that is specified in the `SshPrivKeyfile` parameter.
`SshPort`	You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to `22`.

Table 82. Parameter summary (continued)

Parameter	Description
SshPrivKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshPubKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

Using the IBM Tivoli Monitoring Log File Agent

You can use the log file agent to load log file information into IBM Operations Analytics - Log Analysis.

Before you begin

Consider the size of the log files that you want to load. If a log file is in the region of 50 MB, or more, in size, increase the size of the log file agent cache. In the appropriate configuration file, set BufEvtMaxSize=102400. For WAS log files, update <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf. For DB2 log files, update <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf.

You must delete the appropriate existing cache file. For WAS log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache and for DB2 log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache

For very large log files, update the cache size of the EIF receiver. In the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf file, increase the value of the BufEvtMaxSize property.

Lines in a log that are longer than 4096 characters are, by default, ignored by the IBM Tivoli Monitoring Log File Agent. To force it to read lines longer than 4096 characters, add the EventMaxSize=<*length_of_longest_line*> property to the .conf file that will be used while loading the log.

For WAS update $UNITY_HOME/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf file. DB2 update $UNITY_HOME/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the `<HOME>/IBM/LogAnalysis/utilities` directory, run the following commands:

- `<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop`
- `<HOME>/IBM/LogAnalysis/utilities/unity.sh -start`

About this task

The IBM Tivoli Monitoring Log File Agent might be on the same server as IBM Operations Analytics - Log Analysis and monitoring a local directory. In this scenario, the installation of IBM Operations Analytics - Log Analysis completes all of the configuration required.

If the IBM Tivoli Monitoring Log File Agent is on the same server as IBM Operations Analytics - Log Analysis, but monitoring remote directories, some additional configuration is required. If you want to monitor log files on remote servers, you must make some specific settings changes. For more information about these specific settings, see the *Configuring remote monitoring that uses the predefined configuration files* topic under *IBM Tivoli Log File Agent Configuration* in the *Extending IBM Operations Analytics - Log Analysis* section.

If your configuration requires it, you can use a remote IBM Tivoli Monitoring Log File Agent. In this scenario, install and configure the IBM Tivoli Monitoring Log File Agent based on the your requirements. For more information, see the IBM Tivoli Monitoring documentation: http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/welcome.htm

Procedure

To use the log file agent to load log information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. Ensure that the log file you want to add is in the appropriate directory. For WAS logs, place the log file in the following directory:

   ```
   <HOME>/IBM/LogAnalysis/logsources/WASInsightPack
   ```

   For DB2 logs, place the log file in the following directory:

   ```
   <HOME>/IBM/LogAnalysis/logsources/DB2InsightPack
   ```

   For Generic annotator log files, place the log file in the following directory:

   ```
   $UNITY_HOME/logsources/GAInsightPack
   ```

   The log file is automatically picked up and analyzed. Depending on the size of the log file, processing it could take some time.
3. Optional: To monitor progress, check the following log files:
   - `<HOME>/IBM/LogAnalysis/logs/GenericReceiver.log`
   - `<HOME>/IBM/LogAnalysis/logs/UnityEifReceiver.log`

   When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `$UNITY_HOME/logs` directory to ensure that the data ingestion has completed correctly.

   This example illustrates the addition of a batch of log records. The result is indicated in the `RESPONSE MESSAGE` section of the log file:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO  - LogEventPoster : -
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 2078,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    ++++++++++++++++++++++++++++++++++++
2013-04-2 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    EIF event delivery to Generic Receiver -- SUCCESS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

In this log, the number of log records processed is indicated in the line:

```
{    "batchSize": 2078,    "failures": [    ],    "numFailures": 0 }
```

2078 log records were successfully ingested. The `numFailures` value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the `numFailures` value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a `TIMEOUT FLUSH` event. These events are added to the log file at the end of the session of data collection:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

2013-04-20 04:54:26,341 [pool-4-thread-1] INFO  - LogEventService :
 TIMEOUT FLUSH for logsource:nc9118041070::
  /home/example/LogAnalytics/logsources/
WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO  - LogEventPoster : ---
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 1714,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO  - LogEventService :
   POST RESULT:
{"failures":[],"batchSize":1714,"numFailures":0}

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

To calculate the number of events that have been processed, calculate the sum of all of the `batchSize` values. To calculate the number of events ingested, calculate the sum of all of the `batchSize` values and deduct the total sum of `numFailure` values.

If the ingestion fails, an error message is recorded in the `UnityEIFReceiver.log`:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster : Not Found
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
"CTGLA0401E : Missing log source ","RESPONSE_CODE":404}
```

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++++++++++++++++++++++++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   FAILURE -  ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413 Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the `$UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`.

500 Internal Server Error: Displayed when there is any issue withIBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404 Not Found: Displayed when a Log Source is not found for a hostname and log path combination in the request.

409 Conflict: Displayed if the data batch is posted for a Log Source that is an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the `inputType` field in the request JSON does not match the `inputType` field in the Collection for the requested hostname and log path combination.

200 OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400 Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Results

After the task completes, the log file is indexed and can be searched using the **Search** field on the IBM Operations Analytics - Log Analysis Dashboard.

Considerations when using the IBM Tivoli Monitoring Log File Agent

Before you configure the IBM Tivoli Monitoring Log File Agent to ingest data, update the IBM Tivoli Monitoring Log File Agent to ensure that the configuration is appropriate to the log file that you are likely to ingest.

Log file size

If your log files are likely to exceed 50 MB, increase the size of the IBM Tivoli Monitoring Log File Agent cache: In the appropriate configuration file, set `BufEvtMaxSize=102400`. For WAS log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf`. For DB2 log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf`.

You must delete the appropriate existing cache file. For WAS log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache` and for DB2 log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache`

For very large log files, update the cache size of the EIF receiver. In the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf` file, increase the value of the `BufEvtMaxSize` property.

For WAS, update `<HOME>/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf` file. DB2 update `<HOME>/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf` file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the `<HOME>/IBM/LogAnalysis/utilities` directory, run the following commands:

- `unity.sh -stop`
- `unity.sh -start`

Maximum log line length

The IBM Tivoli Monitoring Log File Agent monitors each log file line. The default maximum line length that can be processed by the IBM Tivoli Monitoring Log File Agent is 4096 bytes. This is equivalent to 4096 ASCII characters. This limitation is related to the log line and not the log record. If a log record consists of multiple log lines, such as in the case of a stack trace, the limit applies to each line. This is a limitation of the IBM Tivoli Monitoring Log File Agent and does not apply if you use an alternative data collection mechanism.

Performance implications of using the IBM Tivoli Monitoring Log File Agent

Loading logs using the IBM Tivoli Monitoring Log File Agent is a CPU bound process. If your system does not meet the minimum requirements you will need to increase the `MaxEventQueueDepth`. On some systems, altering this value may produce a noticeable impact on performance. This will buffer additional IBM Tivoli Monitoring Log File Agent events while they are waiting to be processed. The required value for `MaxEventQueueDepth` may vary depending on the size of the rolled log and the number/speed of your CPU's. If you choose not to increase this value, then older events may be replaced on the event queue by newer events and not sent to the IBM Operations Analytics - Log Analysis server.

To minimize the chance of data loss due to CPU bottlenecks, and to reduce the latency between when a log record is written to the file and when it is loaded, we recommend that the maximum size of a log be small enough so that you system does not fall behind while processing the logs.

Common IBM Tivoli Monitoring Log File Agent configuration conflicts

When you create a remote IBM Tivoli Monitoring Log File Agent (LFA) node and a custom data source and both use the same log path, you can create a conflict.

When you create a custom data source and use it monitor a directory on a remote LFA subnode and you later create another data source, like a remote data source, that monitors the same directory, you can create a conflict in the LFA configuration. These conflicts may cause errors in the Log Analysis log files and reduce the performance of Log Analysis.

The following example is provided to help you to understand this situation.

To avoid these conflicts, you need to avoid monitoring the same directory with different data sources. If you want to monitor two files in the same directory, include the file name in the **Log Path** field when you create the data source.

Example

For example, you are an administrator and you want to monitor files from an LFA that is installed on a remote server as described in the Knowledge Center documentation. See “Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data” on page 233. In this case, the LFA is not part of the Log Analysis product.

First, you must create a custom data source called `Customdatasource` to load data from remote instance of the LFA. In the Data Source creation wizard, you specify the host name and the following log path:

```
/opt/WAS/WAS_logs/myLogFile.log
```

Next, you need to create the configuration and format files for the LFA sub nodes. You create two files, `lfa1.conf` and `lfa1.fmt`. In the `lfa1.conf` file, you specify the following data source:

```
Datasources=/WAS/WAS_logs/some_dir/*
```

Logs that are subsequently generated or appended are ingested by the `Datasource1` data source.

After some time, you create another data source to load data from the same remote server. The new log file is called `newLogFile.log` and it is located in the same directory as the file that you created the `Customdatasource` data source for. You create a remote data source called `Remotedatasource` and specify the log path as:

```
/opt/WAS/WAS_logs/newLogFile.log
```

Finally, you push the log files into Log Analysis.

However, after you push the log file, you notice some strange behaviour in the Log Analysis log files. The `GenericReceiver.log` log file shows that the data is being ingested for `/opt/WAS/WAS_logs/newLogFile.log`. However, it also says that the `/opt/WAS/WAS_logs/newLogFile.log` log file is not a valid data source.

This occurs because the same log file is being monitored by both data sources. As a result, it is monitored by two different LFA sub nodes and in two different streams. The data is loaded but this can waste resources and decrease the overall performance.

To avoid this situation, you must be aware of any possible conflicts especially when you create a custom data source that monitors a directory rather than a file.

Regular expression support for the LFA

The IBM Tivoli Monitoring Log File Agent (LFA) supports specific implementations of regular expressions.

Single-line unstructured data

If you want to use the DSV toolkit to extract and export the data in the comma-separated value (CSV) format for use with the DSV toolkit, you can use a regular expression to extract and export the data.

For example, consider the following log file record:

```
10453072 23460 E5D27197E653C548BDA744E8B407845B AOBEAI1 /EAI      I H R SACP9002
BPUSRSYS/612   23460 - XGNEA108:662:000042:06:E036977:WWS00003:7000:16:1:REV=N
Proc Time=000.03
```

You can configure Log Analysis to use a regular expression to extract and export the data in the comma-separated value (CSV) format. For example, here is an example of a regular expression that is defined in the `.fmt` file:

```
REGEX EAILOG
⌂([0-9]*)(.*)SACP9002(.*):([0-9]*):([0-9]*):([0-9]*):([a-zA-Z0-9]*):
([a-zA-Z0-9]*):([a-zA-Z0-9]*):
(.*)Proc Time=([0-9]*.[0-9]*)
timestamp $1 CustomSlot1
discard $2
SACP9002 $3
bankID $4 CustomSlot3
branchID $5 CustomSlot4
discard3 $6
tellerSID $7 CustomSlot5
workstationID $8 CustomSlot6
transactionTypeID $9 CustomSlot7
discard4 $10
responseTime $11 CustomSlot8
msg PRINTF("%s,%s,%s,%s,%s,%s,%s",timestamp,bankID,branchID,tellerSID,workstationID,
transactionTypeID,responseTime)
END
```

Manipulating date time information for the Generic Annotation Insight Pack

If you use the Generic Annotation Insight Pack or the date time rule set from the Generic Annotation Insight Pack in a custom Insight Pack, you can use some limited regular expressions that you can use to parse time and date information.

The second delimiter, which is a colon (`:`), is not supported. The regular expression replaces the second delimiter with a period (`.`), which is supported. For example, to change a date from `15/12/2014 12:12:12:088 GMT` to `15/12/2014 12:12:12.088 GMT`, you can add the following regular expression to the `.fmt` file:

```
// Matches records for any Log file:
// Log Analytics Data Source chas_access.log

REGEX nongr
([0-9][0-9])/([0-9][0-9])/([0-9][0-9]) ([0-9][0-9]):([0-9][0-9])
:([0-9][0-9]):([0-9][0-9][0-9]) ([A-Z][A-Z][A-Z])
(.*Batch Status for.*)
month $1
day $2
year $3
hour $4
minute $5
second $6
ms $7
zone $8
message $9
hostname example.com
-file /opt/la/IBM/LogAnalysis/logs/GenericReceiver.log
RemoteHost ""
logpath PRINTF("%s",file)
text PRINTF("%s/%s/%s %s:%s:%s.%s %s %s", month, day, year, hour, minute,
second, ms, zone, message)
END
```

Troubleshooting data loading

When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `<HOME>/logs` directory to ensure that the data ingestion has completed correctly.

This example illustrates the addition of a batch of log records. The result is indicated in the RESPONSE MESSAGE section of the log file:

```
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO  - LogEventPoster : -----------
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 2078,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    ++++++++++++++++++++++++++++++++++++
2013-04-2 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    EIF event delivery to Generic Receiver -- SUCCESS
```

In this log, the number of log records processed is indicated in the line:

```
{    "batchSize": 2078,    "failures": [    ],    "numFailures": 0 }
```

2078 log records were successfully ingested. The numFailures value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the numFailures value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a TIMEOUT FLUSH event. These events are added to the log file at the end of the session of data collection:

```

2013-04-20 04:54:26,341 [pool-4-thread-1] INFO  - LogEventService :
 TIMEOUT FLUSH for datasource:nc9118041070::
  /home/yogesh/IBM/LogAnalysis/logsources/WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO  - LogEventPoster : -----------
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 1714,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO  - LogEventService :
   POST RESULT:
{"failures":[],"batchSize":1714,"numFailures":0}

```

To calculate the number of events that have been processed, calculate the sum of all of the batchSize values. To calculate the number of events ingested, calculate the sum of all of the batchSize values and deduct the total sum of numFailure values.

If the ingestion fails, an error message is recorded in the UnityEIFReceiver.log:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster : Not Found
```

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
"CTGLA0401E : Missing data source ","RESPONSE_CODE":404}
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++++++++++++++++++++++++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   FAILURE -  ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413 Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the `$UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`.

500 Internal Server Error: Displayed when there is any issue withIBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404 Not Found: Displayed when a data source is not found for a hostname and log path combination in the request.

409 Conflict: Displayed if the data batch is posted for a data source that is an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the `inputType` field in the request JSON does not match the `inputType` field in the Collection for the requested hostname and log path combination.

200 OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400 Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Configuring the EIF Receiver

How to configure remote or local installations of the Tivoli Event Integration Facility (EIF) receiver to work with IBM Operations Analytics - Log Analysis.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Configuring receiver buffer size and timeout

When collecting data using the IBM Tivoli Monitoring Log File Agent (LFA) and Tivoli Event Integration Facility (EIF) Adapter flow, you might need to change the rate at which events are flushed to the generic receiver for indexing. Incoming events are buffered at the EIF receiver side.

About this task

To improve overall IBM Operations Analytics - Log Analysis performance, you can configure the 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. When the event rate decreases, decrease the buffer size and keep the timeout interval at the default value or increase it, depending on the event rate.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the buffer size and timeout parameters:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder used for the specific remote EIF instance.
3. Change the Timeout and Buffer Size parameters to suit your operating environment:
   ```
   #Timeout in Seconds
   logsource.buffer.wait.timeout=10
   #Buffer Size in Bytes
   logsource.max.buffer.size=250000
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see "`eifutil.sh` command" on page 70.

Results

With higher buffer sizes, notice that it takes a longer time to fill the buffer with events and for batches to be posted to the receiver.

Configuring the EIF receiver user account

The Tivoli Event Integration Facility (EIF) receiver uses the default `unityuser` user account to access the generic receiver. You can change the user account or the default user password in the `unity.conf` configuration file.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the default EIF user or password:

1. Stop IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

  ```
  <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
  ```

- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder that is used for the specific remote EIF instance.
3. Change the following `userid` and `password` parameters to suit your operating environment:

   ```
   unity.data.collector.userid=unityuser

   unity.data.collector.password=password
   ```

   To encrypt the password, use the `unity_securityUtility.sh` command. For more information, see "Changing the default password for the Data Collector and EIF Receiver" on page 266.
4. Save your changes.
5. Restart IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to restart IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
     ```

   - If you use a remote installation of the EIF, use the `eifutil.sh -restart` command to restart the instances. For more information, see "`eifutil.sh` command" on page 70.

Results

The EIF receiver uses the new credentials to access the generic receiver.

Configuring the number of events in the EIF Receiver

You can configure the number of events that the EIF Receiver stores for each internal queue. If you intend to ingest a large quantity of data and at a high rate, configure these values to larger values. However, increasing this value also increases the memory requirements for EIF Receiver.

About this task

Ensure that you have sufficient memory to support the number of events in the queue.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change this setting:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder used for the specific remote EIF instance.
3. Locate these lines and change the value to reflect your requirements:
   ```
   unity.data.collector.eif.consumer.num.events=1000000
   unity.data.collector.event.manager.num.events=20000
   ```
   The following settings are applicable per data source:
   ```
   unity.data.collector.event.service.num.events=20000
   unity.data.collector.event.poster.num.events=500
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

Configuring the EIF Receiver memory clean up interval

IBM Operations Analytics - Log Analysis ensures that the memory used for data collection with the Log File Agent using a property in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf` file. The EIF Receiver uses this value to manage the memory usage. The configuration cycle is set to a value in minutes with a default value of 2 minutes.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To configure this property:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the <*remote_deployment_location*>`/LogAnalysis/DataForwarders/EIFReceivers/`<*eif_inst_#*>`/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder that is used for the specific remote EIF instance.
3. Change the parameters to suit your operating environment:

```
#gc interval is in minutes
unity.data.collector.gc.interval=2
```

4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see "`eifutil.sh` command" on page 70.

Configuring scalable data streaming from multiple, remote sources

To facilitate dynamic data streaming that is scalable across multiple remote sources, you must configure IBM Operations Analytics - Log Analysis after you install it.

To enable data collection from remote hosts, you must complete the following steps:

1. Install Apache Solr on the remote machine.
2. Set up Secure Shell (SSH) communication.
3. Configure SSH to work with the remote installer utility.
4. Use the remote installer utility to install instances of the Event Integration Facility (EIF) or the IBM Tivoli Monitoring Log File Agent (LFA) on remote machines.
5. Configure the EIF so that it is compatible with the remote instances that your create. If you use the LFA, you do not have to configure the local installation. However, you do have to manually configure the sub nodes.

You can also maintain and administer these connections after you set them up.

As an alternative to streaming data, You can batch load data. For more information, see "Loading and streaming data" on page 223.

Installing Apache Solr on remote machines

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

About this task

If no local instances of Apache Solr exist, then you need to install the instances on the remote machine as soon as you install IBM Operations Analytics - Log Analysis. If there is a local instance of Apache Solr, you can install the remote instances whenever you want.

You must use a non-root user to run the script.

You cannot use the installer to install Apache Solr on a local machine.

You cannot use the installer to install multiple Apache Solr nodes on a single remote machine.

To install Apache Solr on multiple remote machines, run the script separately for each remote machine. You cannot use the installer to install instances of Apache Solr simultaneously or in parallel.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:

   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```

2. To run the `remote_deploy.sh` script, enter the following command:

   ```
   ./remote_deploy_solr.sh -install
   ```

3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password-less SSH authentication is disabled. If password-less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the <HOME>/IBM/LogAnalysis/utilities/config directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

   SSH Port
   Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

   Top-level Installation Directory
   To use the default value, which is `<HOME>`, press enter. Alternatively, you can enter the path to the directory where you want to install the DE.

   Apache Solr Search Port
   To use the default value, 9989, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

Apache Solr Query Service Port

To use the default value, 7205, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

4. To start the installation, press enter. In most cases, the installation takes about 5 minutes to complete.

Results

The results of the installation are output in the log file in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs/ManageSolrnodes.log` file.

To view the status for the instances of Apache Solr that are installed remote machines, run the `unity.sh -status` command.

Example

Here is an example script output:

```
Remote Hostname in FQDN format:12345.example.com
username:unity
password:*********
SSH port: [22]
Top-level Installation Directory: [/home/unity]
Solr Search Port: [9989]
Solr Query Service Port: [7205]

Script is ready for remote installation of Solr:
Review the following inputs ....
-----------------------------------------------------------------------------
Remote Host Name: 12345.example.com
Remote User Name: unity
Remote SSH Port: 22
Top-level remote installation directory: /home/unity
Solr v9.0 - remote installation directory:
/home/unity/IBM/LogAnalysis
Solr - remote ports: 9989, 7205
------------------------------------------------------------------------
['q' - Abort]['Enter' - Install]

Sat Nov 16 03:08:38 CST 2013 Starting remote installation of Solr
, this will take couple of minutes to complete  ....
Sat Nov 16 03:08:38 CST 2013 Waiting for remote installation to complete ....
Sat Nov 16 03:11:47 CST 2013 Successfully installed Solr
Solr on remote host:12345.example.com ....
```

Removing Apache Solr instances:

Before you remove an installation of IBM Operations Analytics - Log Analysis, you must remove Apache Solr.

About this task

Note: Do not remove Apache Solr if IBM Operations Analytics - Log Analysis is still being used. IBM Operations Analytics - Log Analysis does not function properly when any instances of Apache Solr are removed. For this reason, only remove Apache Solr when you are about to uninstall IBM Operations Analytics - Log Analysis.

If you installed Apache Solr locally and remotely, remove the local instance first, then remove the remotely installed instances.

This process uses Installation Manager to remove Apache Solr instances. You can also do so silently. To run the silent removal, run following `imcl -c` command, enter 3 to modify the installation, and remove the instance.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:
```
cd <HOME>/IBM/LogAnalysis/solr_install_tool
```
2. To run the `remote_deploy.sh` uninstall script, enter the following command:
```
./remote_deploy.sh -uninstall
```
3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password less SSH authentication is disabled. If password less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the `<`*`UNITY_HOME`*`>/utilities/config` directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

   SSH Port
   Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

   Top-level Installation Directory
   To use the default value, which is `<HOME>/IBM/LogAnalysis`, press enter. Alternatively, you can enter the path to the directory where Apache Solr is installed.
4. To start the removal, press enter. You can view the logs in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs` directory.

Results

When all the remote nodes are removed, you can safely uninstall IBM Operations Analytics - Log Analysis.

Setting up Secure Shell to use key-based authentication

Secure Shell (SSH) is a cryptographic network protocol for secure data communication between different computers. You set up key-based authentication between the IBM Operations Analytics - Log Analysis servers and the remote computers to which it connects.

About this task

Benefits of using key-based authentication:
- Data is transferred across a secure channel.
- The administrator is no longer concerned about the password changes for the remote servers.
- The passphrase is independent of the individual server password policy.

- One passphrase is used for multiple servers. Only the public key file must be copied to the client server.

For more information you can view the man pages for **ssh-keygen** by running this command:

```
man ssh-keygen
```

Procedure

1. To generate public and private keys, enter the following command:

   ```
   ssh-keygen -t rsa
   ```

   or either of the following commands:

   ```
   ssh-keygen
   (This command generates the same results as ssh-keygen -t rsa.)
   ```

   ```
   ssh-keygen -t dsa
   (If you specify dsa, the generated keys include _dsa in their file names.)
   ```

   The following example shows what a valid output might look like:

   ```
   bash-3.2$
   bash-3.2$ ssh-keygen -t rsa
   Generating public/private rsa key pair.
   Enter file in which you want to save the key (/home/unity/.ssh/id_rsa):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /home/unity/.ssh/id_rsa.
   Your public key has been saved in /home/unity/.ssh/id_rsa.pub.
   The key fingerprint is:
   4a:ef:d5:7a:d8:55:b3:98:a1:1f:62:be:dd:c4:60:6e unity@<variable>.example.com
   The key's randomart image is:
   +--[ RSA 2048]----+
   |                 |
   |                 |
   |                 |
   |             . ..|
   |      . S   .o+.o|
   |     . o   =o++. |
   |       . . +o+E.o |
   |        . ..o=.o  |
   |         . .o.. . |
   +-----------------+
   bash-3.2$
   ```

   Enter the passphrase. (The **Enter passphrase** field can remain blank to specify an empty passphrase.)

2. To view the contents of the public key file, run the following commands:

   ```
   cd ~/.ssh
   ls -l id_rsa*
   cat id_rsa.pub
   ```

   The command output is:

   ```
   bash-3.2$
   bash-3.2$ cat .ssh/id_rsa.pub
   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDg0/GGoxGzyC7AwjbwnpOhCaeztIRt6yhAg
   GKdwM7nb7Iiv0RgwT4/48E26K1Ur9HrI1W/j0K0JHQw
   vaAFibqeLmqLdK9ctCE901ywTOPFcYeBYPUF9vp/MgaypgGxVwDbW/e0SNPb7YAtZpjRoqeUq
   oYoKzFXXspQkxdhcQfpx0RYMbQdGGg03hDCM2wr2KP
   VuTVniF2IvDu1C4fcRkUPr8aQNMiuEcJgV3VHhlau/0Uo0YpH53NXKhn/sx8xdyTVsKQ1rhW8
   g07HIVc2Tf9ZF2gYXn/HbjE509xK/APu2nztt0h+Air
   JyT5jYMi/IvSI0zbPyc0p9WijPeG8r/v unity@<variable>.in.ibm.com
   bash-3.2$
   ```

3. Create a directory called `.ssh` on the remote server. Use this to store the public key.

4. Copy the public key file (`id_rsa.pub`) to the `.ssh` directory on the remote client:

```
scp /home/unity/.ssh/id_rsa.pub
<username>@<remotehostname>:/
<HOME>/.ssh/id_rsa.pub
```

where <*hostname*> is the system host name and <*username*> is the system user name.

5. Add the content of the public key to the `authorized_keys` file on the remote host.

   ```
   bash-3.2$ ssh <username>@<remotehostname>
   bash-3.2$ cd ~/.ssh
   bash-3.2$ cat id_rsa.pub >> authorized_keys
   bash-3.2$ rm id_rsa.pub
   bash-3.2$ exit
   ```

6. Ensure that there are no duplicate keys for the same client in the authorized_keys file.
7. Log in to the remote computer to ensure that key-based SSH is working:

   ```
   ssh <username>@<hostname>
   ```

   Enter the passphrase, if prompted.

   ```
   bash-3.2$ bash-3.2$ ssh <username>@<remotehostname>
   Enter passphrase for key '/home/unity/.ssh/id_rsa':
   Last unsuccessful login: Mon Jul 15 14:22:37 2013 on ssh from <variable>.example.com
   Last login: Mon Jul 15 14:26:54 2013 on ssh from <variable>.example.com
   $
   ```

   Configuration of key-based authentication is complete.

Results

The steps may not work because different versions of SSH are supported by the operating systems that are used by the remote servers. For more information about how to solve this issue, see the *Secure Shell (SSH) configuration does not work* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Configuring secure shell (SSH) communication for multiple remote hosts:

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

Before you begin

Before you configure SSH for multiple remote hosts, you must configure SSH between IBM Operations Analytics - Log Analysis and the remote hosts. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the Information Center.

About this task

By default, the SSH properties file, `ssh-config.properties` file, is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory. If you save the file to another location, the utility requests that the user enters values for the remote host, user, and password. In this case, the utility does not use the values specified in the file.

If you save the `ssh-config.properties` file in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory, the `eif_remote_install_tool` utility uses the properties specified in the file.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

If you specify values for both the password and the private key file path, the utility uses the file to create a password-less SSH connection.

If you do not specify a value for the password or the private key file path, IBM Operations Analytics - Log Analysis cannot create a connection and instead generates an error message in the log:

```
    ERROR:
    example.unity.remote.SshConfigException:
Property file config/ssh-config.properties must contain at least one of:
PASSWORD, PATH_OF_PASSWORD_LESS_SSH_KEY
    Correct SSH configuration OR reconfigure and retry
    Installation Aborted....!
```

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory and open the `ssh-config.properties` file.
2. Specify values for the following properties for each remote host:
   - Remote host
   - Remote user ID
   - Port
   - Connection timeout in milliseconds. The default is 6000.

   For example:

   ```
   REMOTE_HOST=<REMOTE_HOST>
   PORT=<PORT>
   TIME_OUT=60000
   USER=<REMOTE_USER>
   ```
3. For password-based authentication, you also need to specify the password in the configuration file. For example:

   ```
   PASSWORD=password1
   ```
4. For public key based authentication, specify the path to the directory that contains the private key file. For example:

   ```
   PATH_OF_PASSWORD_LESS_SSH_KEY=/home/pass/.ssh/id_rsa
   ```
5. If your installation of SSH requires a passphrase, specify the passphrase. For example:

   ```
   PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=passphrase1
   ```

Configuring data collection for scalability on multiple remote nodes

To facilitate scalable data collection on multiple remote nodes, use the `install.sh` command to install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server.

Before you begin

Before you run the command, you must configure secure shell (SSH) communication between the local installation of IBM Operations Analytics - Log Analysis and the remote host. For more information about how to do so, see "Configuring secure shell (SSH) communication for multiple remote hosts" on page 66.

About this task

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis.

You can use the remote installer in the following scenarios:

- If you have a high rate of data ingestion on multiple data sources. For example, if you have 100 or more events per second and 20 or more data sources.
- If you require improved throughput performance on the remote server.
- If the hardware resources on the remote server are restrained.
- If you want to optimize performance according to the conditions described on the Performance developer works page here: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM Log Analytics Beta/page/Performance and tuning

You can use the command to deploy up to 20 instances of the Tivoli Event Integration Facility Receiver or a single instance of the IBM Tivoli Monitoring Log File Agent on a remote node. The command deploys and configures IBM Java 1.7. The command also configures the deployed Tivoli Event Integration Facility Receiver instance to communicate with the IBM Operations Analytics - Log Analysis Data Collector interface.

However, this command does not configure the IBM Tivoli Monitoring Log File Agent subnode. You must configure this setting manually. Both the remote and local instance of the IBM Tivoli Monitoring Log File Agent can monitor remote data sources. For more information about configuring IBM Tivoli Monitoring Log File Agent, see "Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data" on page 233.

To ensure that the remote instances of the Tivoli Event Integration Facility work with the local Data Collector interface, you must create the remotely deployedTivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances as part of the same installation. This is because the encryption configuration and signature generation is done during the main installation. If you install IBM Operations Analytics - Log Analysis after you set up the remote nodes, you must install the remote Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances again. However, you can remove remote instances of the Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent without installing IBM Operations Analytics - Log Analysis again.

Note: If you use the script to install the remote instance on a server that uses the SUSE Linux Enterprise Server 11 operating system, the script fails. To resolve this issue, see the *Cannot install remote EIF instance on SUSE* topic in the *Troubleshooting* IBM Operations Analytics - Log Analysis guide.

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory and run the `install.sh` command. You are prompted for a series of inputs.
2. Enter the remote installation directory. This value must be the location where the deployed artifacts are installed on the remote host.
3. If you want to deploy the Tivoli Event Integration Facility Receiver, select it. If you do, enter the Tivoli Event Integration Facility Receiver instances that you want to deploy.
4. If you want to deploy the IBM Tivoli Monitoring Log File Agent instance on the remote node, select it.

Results

After you complete the procedure, you can now collect data from the remote hosts.

What to do next

After the initial setup, you will want to periodically change the configuration. IBM provides two commands to start and stop the instances so that you can update the configuration.

To administer Tivoli Event Integration Facility Receiver instances, use the `eifutil.sh` command.

To administer IBM Tivoli Monitoring Log File Agent instances, use the `lfautil.sh` command.

`eifutil.sh` command:

To administer EIF Receiver instances, use the `eifutil.sh` command.

Syntax

The `eifutil.sh` command has the following syntax and is in the *<USER_HOME_REMOTE>*`/DataForwarders/EIFReceivers/utilities` where *<USER_HOME_REMOTE>* is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where *<Inst_ID>* is the ID for the specific EIF instance.

Parameters

`-status`
Displays the status for the installed instances. For example:

```
=============================================================================
COMPONENT            Instance         PID            PORT            STATUS
=============================================================================
EIF Receiver         eif_inst_1      13983           6601              UP
EIF Receiver         eif_inst_2      14475           6602              UP
EIF Receiver         eif_inst_3      14982           6603              UP
EIF Receiver         eif_inst_4      15474           6604              UP
EIF Receiver         eif_inst_5      15966           6605              UP
=============================================================================
```

-start <*Inst_id*>
Starts the specified instance.

-stop <*Inst_id*>
Stops the specified instance.

-startAll
Starts all instances.

-stopAll
Stops all instances.

-restart<*Inst_id*>
Restarts the specified instance.

-restartAll
Restarts all the instances.

lfautil.sh command:

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the `lfautil.sh` command.

Syntax

The `lfautil.sh` command has the following syntax and is in the <*USER_HOME_REMOTE*>`/utilities/` directory on the remote host where <*USER_HOME_REMOTE*> is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start Starts all the LFA instances on the remote host.

-stop Stops all the LFA instances on the remote host.

-status
Displays the status for the LFA instances on the remote host. For example:

```
==========================================
COMPONENT            PID            STATUS
==========================================
Log File Agent        23995             UP
==========================================
```

-restart
Restarts the LFA instances on the remote host.

Loading batches of data

In addition to streaming data directly, you can also load batches of historic data for test or other purposes.

Generic Receiver

The Generic Receiver is a component of IBM Operations Analytics - Log Analysis that supports the REST interface for loading data into IBM Operations Analytics - Log Analysis. The REST API uses JSON (JavaScript Object Notation) as an input and returns JSON as an output after the incoming logs are processed. If an error occurs, the API returns an error code and a message.

Processing a batch

Invoking the Generic Receiver API initiates the processing of a batch that is contained in the Input JSON. Buffer a set of log records to create a batch and send data in batches to IBM Operations Analytics - Log Analysis. The batches must be sent in the order in which logs are generated for a specific data source. The size of each batch must be less than the batch size (500000 bytes) supported by IBM Operations Analytics - Log Analysis. At the minimum, you can send data for a single log record in a batch. The Generic Receiver processes a batch by:

- Splitting the batch into multiple log records by using the Splitter that was specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Annotates every log record that is found by the Splitter by using the Annotator that is specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Indexing the annotated log record in the back-end search engine

As special cases, split and annotated steps are skipped if the Splitter or Annotator is specified as null in the SourceType. Even if there is no data to split, you need to send an empty string in the text field of the Input JSON.

Batching of data at the client might lead to an incomplete log record at the end of the batch. This incomplete log record gets buffered in IBM Operations Analytics - Log Analysis and stitched with the remaining data in the subsequent batch to form a complete log record. This stitching assumes that you are maintaining the log record order of the data that is sent to IBM Operations Analytics - Log Analysis. If the order is not maintained, then logs are not correctly split into log records.

Input JSON

The basic structure of an Input JSON file is:

```
{
"hostname":  ,    (String)
"logpath":" ,   (String)
"batchsize": ,  (String)
"inputType":    // Optional (String) "LOGS";
"flush":   // Optional (boolean)
"payload":   // (JSONObject)
{
"name1":"value1",    // Optional
...
...
"nameN":"valueN" ,      // Optional
text : "log record 1 log record 2 ..."  (String)
 }
}
```

The following parameters in the Input JSON are mandatory:

hostname
: The host name that corresponds to the data source for which you want to ingest data.

logpath
The log path that corresponds to the data source for which you want to ingest data.

batchsize
The number of BYTES of logs that are sent in one batch to IBM Operations Analytics - Log Analysis (less than 500,000).

inputType
The type of input data: `LOGS`.

flush flag
A flag that indicates to the Generic Receiver whether the last record in the batch is a complete log record. Typically, this flag would be set to true in the last batch upon reaching the end of file.

payload.txt
This text contains the actual log records to be split, annotated, and indexed into IBM Operations Analytics - Log Analysis. The text portion is split into log records by the Splitter, annotated by the Annotator, and then indexed. If you do not have any log records, but want to index only structured (name-value pairs) data, you can specify this mandatory field as an empty string.

More metadata (optional) to be indexed with every log record of the batch can be specified as name-value pairs in the input JSON or the payload within the input JSON. This metadata is applicable at the batch level. For posting distinct metadata for each log record, send 1 log record at a time in the batch.

Post the input JSON to the following URL:

```
http://<UNITY_HOST_NAME>:<UNITY_PORT>/Unity/DataCollector
```

where `<UNITY_HOST_NAME>` is the machine on which you installed IBM Operations Analytics - Log Analysis and `<UNITY_PORT>` is the port on which it is running. The default port is 9988. The client (Java or Script) sending data into IBM Operations Analytics - Log Analysis needs to authenticate by using the form-based mechanism that is implemented in IBM Operations Analytics - Log Analysis before the Data Collector API is invoked. Refer to the authentication and security design document for details.

Output JSON

The output that is sent by the Generic Receiver after indexing logs contains the count and detailed information on the failure cases in a JSON Array. The details include the actual logRecord, specific error message, and any exception. The basic structure of an Output JSON file is:

```
{
"batchSize" : ,   // (int)
"numFailures" : ,  // (int)
"failures" :    // (JSONArray)
  [
  {
   "logRecord" : ,  // (JSONObject)
   "errorMessage": ,  // (String)
   "exception" : ,  // (JSONArray)
  },
  .
  .
  .
  {
   }
  ]
}
```

Serviceability

As you send data into IBM Operations Analytics - Log Analysis, you might encounter errors that occur before the incoming batch gets processed or errors that occur during processing of batch and indexing log records.

If errors occur before the incoming batch gets processed, the Generic receiver returns an error code and message. To correct the problem, process the error code, make any required changes, and resend the data.

Possible causes for error code 400 (HttpServletResponse.SC_BAD_REQUEST) include:

- Invalid input JSON
- Input batch size is greater than what is supported (500000 bytes)
- No data source is configured from the Admin UI for the host name and log path combination that is sent in the input JSON
- The input type (LOGS) specified in the batch does not match the value that is specified in the logsource that is configured from the Admin UI

Possible causes for error code 500 (HttpServletResponse.SC_INTERNAL_SERVER_ERROR) include:

- An exception that is encountered in any of the steps of the ingestion pipeline (for example, during splitting of a batch).
- An internal IBM Operations Analytics - Log Analysis database-related error.
- Any other exception in IBM Operations Analytics - Log Analysis.

If errors occur during processing of batch and indexing log records, the output JSON provides details of indexing failure. To correct the problem, process the error code, make any required changes, and resend only the affected log records. Sending the same log record twice to IBM Operations Analytics - Log Analysis results in duplicate records in the back-end index and duplicate records in the search results.

Batch loading historic log data with the Data Collector client

Use the Data Collector client to ingest data in batch mode. Use this method to review historic log data. This is the easiest method if you want to ingest large log files for historic analysis.

Before you begin

If you want to use the Data Collector client to load data from remote sources, you must configure the data collector on the remote host before you can configure the local data collector as described here. For more information, see “Configuring the Data Collector client to ingest data from remote hosts” on page 263.

About this task

If you want to load a log file that does not include time stamp information, ensure that the values for `timestamp` and `timestampFormat` are configured in `javaDatacollector.properties`. IBM Operations Analytics - Log Analysis cannot index log files without a time stamp, but if no time stamp information is found in a log file, the value that is configured in javaDatacollector.properties is used.

Procedure

To use the Data Collector client to load log file information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. At the command line, navigate to the <HOME>/utilities/datacollector-client directory.
3. Update the configuration file that is used by the Data Collector client, javaDatacollector.properties. Set the following properties, as appropriate:

   logFile
   The full path of the file you want to ingest.

   servletURL
   The URL of the Data Collector service.

   userid The user ID for the Data Collector service.

   password
   The password for the Data Collector service.

   datasource
   The datasource that you want to use to load data.

   timestamp
   The time stamp to use if a time stamp is not found in the log file.

   batchsize
   The number of BYTES of logs that are sent in one batch. The default value is 500,000.

   keystore
   The full path to the keystore file.

   inputType
   The valid input types are: LOGS, CONFIGFILES, SUPPORTDOCS. The default value is LOGS.

   flush flag
   If the default true is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to false, no flush signal is sent when the end of file is reached.

   The following sample javaDatacollector.properties file displays the configuration for loading the SystemOut.log log file.

```
#Full path of the file you want to read and upload to Unity
logFile = SystemOut.log
#The URL of the REST service. Update the host/port information if required
servletURL = https://hostname:9987/Unity/DataCollector
#The user ID to use to access the unity rest service
userid=unityuser
#The password to use to access the unity rest service
password=password
datasource=Systemout
#Time stamp to use if your content can not find a time stamp in log record.
The same time stamp would be used for all records
timestamp = 01/16/2013 17:27:23:964 GMT+05:30
#The number of BYTES of logs sent in one batch to Unity
batchsize = 500000
#The full path to the keystore file
keystore = /home/unity/IBM/LogAnalysisTest/wlp/usr/servers/Unity/
```

```
keystore/unity.ks
#input data type - LOGS, CONFIGFILES, SUPPORTDOCS
inputType = LOGS
#flush flag:
#true : (default) if the client should send a flush signal to the Generic
 Receiver for the last batch of this file
#false : if no flush signal to be sent upon reaching eod-of-file
flushflag = true
#Other properties (name/value pairs, e.g. middleware = WAS) that you want
 to add to all json records
#These properties need to be appropriately added to the index configuration
```

4. Ensure that the Data Collector client JAR file, `datacollector-client.jar`, has execute permissions.
5. Use the following command to run the Data Collector client with the correct inputs:

   ```
   <HOME>/ibm-java/bin/java
   -jar datacollector-client.jar
   ```

Results

After the task completes, the log file is indexed and can be searched in the **Search** workspace.

Configuring the Data Collector client to ingest data from remote hosts

If you want to use the Data Collector client to collect data from a remote server and return it to the local machine, you must configure the data collector on the remote host.

Before you begin

You must use the instance of IBMJava Runtime Engine (JRE) 1.7 that is installed by the remote installer. Before you configure the data collector, you must use the remote installer to install at least one instance of IBM Tivoli Monitoring Log File Agent or the EIF Receiver on a remote machine. For more information, see the *Configuring data collection for scalability on multiple remote nodes* topic in the Installation Guide.

About this task

To configure the Data Collector on the remote host, copy the data collector client files from your local version of the data collector files to the remote host.

Procedure

1. Copy the `<HOME>/utilities/datacollector-client` directory and all the files that are contained in it from the local installation of IBM Operations Analytics - Log Analysis to the remote machine.
2. Add the location of the log and keystore files to the `javaDatacollector.properties` file in the directory that you copied the data collector to in the previous step. The keystore file is named `unity.ks` and it is available in the <*Remote_install_dir*>`/LogAnalysis/store/` directory on the remote machine. Where <*Remote_install_dir*> is the directory where you installed the remote instance as described in the *Prerequisites* section here.

Results

After you complete the configuration, you must complete the Data Collector configuration. For more information about how to do this, see "Batch loading historic log data with the Data Collector client" on page 262. You must ensure that the remote installation uses the IBMJava Runtime Engine (JRE) 1.7 that is installed by the remote installer. IBM Java Runtime Engine (JRE) 1.7 is stored in the <*Remote_install_dir*>/LogAnalysis/ibm-java/ directory.

Data Collector properties

Before you can use the data collector to stream data or load a batch of historic data, edit the javaDatacollector.props file.

The javaDatacollector.props file is in the <HOME>/IBM/LogAnalysis/ utilitiesdatacollector-client folder.

The logFile, hostname, logpath, and keystore parameters are required.

The userid, password, and keystore parameters are automatically populated with the default values that are created during the installation. If you want, you can change these but you do not need to.

Table 83. Data Collector properties

Parameter	Value
logFile	The full path of the file you want to load.
servletURL	The URL of the Data Collector service.
userid	The user ID for the Data Collector service.
password	The password for the Data Collector service.
datasource	The datasource that you want to use to load data.
timestamp	The time stamp to use if a time stamp is not found in the log file.
batchsize	The number of BYTES of logs sent in one batch. The default value is 500,000.
keystore	The full path to the keystore file.
inputType	The valid input type is LOGS.
flush flag	If the default true is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to false no flush signal is sent when the end-of-file is reached.

Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent

You can use the IBM Tivoli Monitoring Log File Agent to load batches of historic data for testing and other purposes.

Procedure

1. Copy the log files that you want to load to a temporary directory on the IBM Operations Analytics - Log Analysis server. For example, to upload a batch of

log files from an installation of WebSphere Application Server, you copy the `SampleSystemOut.log` file to the `/tmp/logs/` directory.
2. Create a custom data source.
3. Copy the log file to the directory that you specified in the `logpath` parameter when you created the data source.

Extending storage space available to Apache Solr

You can add more Apache Solr storage directories outside the initial IBM Operations Analytics - Log Analysis Apache Solr installation location if the disk on which Apache Solr was installed reached maximum capacity.

Before you begin

Ensure that the Apache Solr storage directories are present on all Apache Solr servers and are writable.

About this task

Switching to a new Apache Solr directory is not instantaneous. Therefore, it is to monitor the disk usage of your Apache Solr directory to ensure that extra directories are added before the current storage directory reaches maximum capacity.

Procedure

To enable the storage extension capability, complete the following steps.

1. Stop IBM Operations Analytics - Log Analysis with the following command.
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
   ```
2. Open the `unitysetup.properties` file in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
3. Add the following property to the directory
   ```
   ENABLE_SOLR_RELOCATION=true
   ```
4. Create the following properties file
   ```
   <HOME>/solrConfigs/storageConfig.properties
   ```

   For example,
   ```
   /home/unity/IBM/LogAnalysis/solrConfigs/storageConfig.properties
   ```
5. Open the `storageConfig.properties` file and add the following property to the file.
   ```
   SOLR_STORAGE_DIR=storage-path-on-solr-nodes
   ```

   For example,
   ```
   SOLR_STORAGE_DIR=/opt/scala/ext_storage
   ```
6. Restart IBM Operations Analytics - Log Analysis with the following command.
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Results

The new IBM Operations Analytics - Log Analysis configuration file enables the specification of custom data storage locations. The new locations are written to when IBM Operations Analytics - Log Analysis crosses the default boundary of 1 day.

Changing the default boundary for creating Apache Solr collections

You can change the default boundary that is associated with extending Apache Solr storage space depending on your business needs.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file.
2. Locate and modify the value of the `COLLECTION_ASYNC_WINDOW` property from the default value of `1d` (1 day).

   Note: The minimum property size is `6h`.

   The boundary size can be specified in minutes `(m)`, hours `(h)`, or days `(d)`.
3. Restart IBM Operations Analytics - Log Analysis with the following command.
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Changing the default password for the Data Collector and EIF Receiver

If you want, you can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics. This is optional.

Changing the default EIF Receiver or Data Collector password

You can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

About this task

After you install IBM Operations Analytics - Log Analysis, the EIF Receiver and the Data Collector are configured to use the default user name and password to connect to IBM Operations Analytics - Log Analysis. The encrypted passwords are defined in the following files:

- Data Collector client is named `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`.
- EIF Receiver is named`<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`.

IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to encrypt and decrypt passwords for your installation, in the following format:

```
password={aes}<Unique_string_of_alphanumeric_characters>
```

For example, the `javaDatacollector.properties` file uses the `unityuser` user ID to access the Data Collector server. In this example, IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to generate the following password:

```
{aes}7DB629EC03AABEC6C4484F160FB23EE8
```

The encrypted password is replicated to the configuration files for the Data Collector and the EIF Receiver.

Procedure

1. To change the default password, use the `unity_securityUtility.sh` command.

For more information about this command, see "unity_securityUtility.sh command" on page 205.

2. Update the configuration files for the Data Collector or the EIF Receiver.
3. Optional: If you want to change the password on remote instances of the EIF Receiver, complete the previous steps and copy the unity.conf file from the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/ directory on the local machine to the *<remote_deployment_location>*/LogAnalysis/DataForwarders/EIFReceivers/*<eif_inst_#>*/config/unity.conf directory on the remote machine. Where *<remote_deployment_location>* is the directory on the remote machine where you deployed the EIF instance. *<eif_inst_#>* is the folder that is used for the specific remote EIF instance.

Example

For example, you want to change the default password for the default user that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis to myNewPassword. Complete the following steps:

1. Go to the IBM/LogAnalysis/utilities directory.
2. Run the unity_securityUtility.sh command as follows:

```
 [utilities]$ ./unity_securityUtility.sh encode myNewPassword
 Using keystore file unity.ks
 <HOME>/IBM/LogAnalysis/utilities/../wlp/usr/servers/Unity/
keystore/unity.ks
 {aes}E6FF5235A9787013DD2725D302F7D08
```

3. Copy the AES encrypted password to the relevant configuration files, for example copy it to the Data Collector file. You must copy the complete, encrypted string from the command output, including the {aes} prefix. For example:

```
{aes}E6FF5235A9787013DD2725D302F7D088
```

unity_securityUtility.sh command

You can use the unity_securityUtility.sh command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Syntax

The unity_securityUtility.sh command is in the <HOME>/IBM/LogAnalysis/utilities directory and it has the following syntax:

```
unity_securityUtility.sh encode [textToEncode] [unity.ks]
```

Parameters

The unity_securityUtility.sh command has the following parameters:

encode

The encode action returns an AES encrypted version of the text that you enter as the text to encrypt.

[*textToEncode*]
Use the [*textToEncode*] parameter to enter the password that you want to encrypt. If you do not specify a password for this parameter, IBM Operations Analytics - Log Analysis prompts you for one.

[unity.ks]

The `unity.ks` file is the default keystore that is generated automatically during installation. It controls how the password is encrypted and decrypted.

The `unity.ks` file is used to encrypt and decrypt passwords for the following features:

- Java data collector client in the `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties` file.
- EIF Receiver in the `<HOME>/IBM/LogAnalysis/utilities/UnityEIFReceiver/config/unity.conf` file.

For an example of how to use this command, see “Changing the default EIF Receiver or Data Collector password” on page 266.

Installing logstash

Installing logstash on a remote node extends IBM Operations Analytics - Log Analysis functions so it can ingest and perform metadata searches against log data that is acquired by logstash.

logstash 1.4.2 is bundled and installed with IBM Operations Analytics - Log Analysis. You can install logstash on a local host but to improve system performance, install logstash on a remote node.

This document describes the version of logstash that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of logstash might have been published after this version of IBM Operations Analytics - Log Analysis. To download the most up-to-date logstash versions and updated documentation, see https://www.elastic.co/downloads/logstash.

logstash is an open source tool for managing events and logs. It can be used to collect logs, parse them, and send them to another tool such as IBM Operations Analytics - Log Analysis to store them for later use.

The logstash agent is an event pipeline that consists of three parts:

1. Inputs
2. Filters
3. Outputs

Inputs generate events. Filters modify events. Outputs send the event somewhere. For example, events can be sent to storage for future display or search, or to the IBM Operations Analytics - Log Analysis framework. Events can have a type, which is used to trigger certain filters. Tags can be used to specify an order for event processing as well as event routing to specific filters and outputs.

logstash can be used as a "pre-processor" to analyze sources and provide a semi-structured or structured feed to IBM Operations Analytics - Log Analysis for the purposes of searching and potential usage within custom analytics applications.

For more information on logstash events, see the section *the life of an event in logstash* at https://www.elastic.co/guide/en/logstash/current/index.html.

Dependencies

Supported version of logstash and its dependencies.

Supported logstash version

The supported version of logstash is 1.4.2 .

DSV Toolkit requirement

DSV Toolkit v1.1.0.1 or higher for generating IBM Operations Analytics - Log Analysis Insight Packs. The Insight Packs are used to index log records that have been annotated using logstash. You only require the DSV toolkit if you want to use logstash to perform ingestion, splitting, annotating or for when the data being read by logstash is in DSV format. For more information on this user scenario, see "Configuring logstash for rapid annotation and pre-indexing processing" on page 273.

Generic Annotation Insight Pack

Generic Annotation v1.1.0, or v1.1.1 (refresh 1) is recommended for the normalized timestamp splitter function, which recognizes a variety of timestamps.

Installing logstash on a remote node

You can install logstash on a remote node to improve system performance.

Before you begin

Ensure that the SSH user has the correct permissions for installation. For more information on SSH configuration, see "Secure Shell (ssh) configuration for remote logstash" on page 270 in the *Loading and streaming data guide*.

About this task

logstash is processor and system resource intensive. logstash can be installed on the local host but to improve system performance, install logstash on a remote node.

Procedure

1. To install logstash, run the following command:
   ```
   <HOME>/IBM/LogAnalysis/remote_install_tool/install.sh
   ```
2. The installation script installs logstash, and provides options to install the EIF receivers and log file Agent. To select each option, including logstash, select `y` or `Y`.
3. Provide the path to the installation location on the remote host.

Results

logstash is installed in the `<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/` directory. To confirm the installation, logon to the remote node as the configured SSH user and go to the installation location.

Example

The following are example deployments:

logstash example:

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/
```

Output plug-in configuration path:

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/logstash-scala/logstash/
config/logstash-scala.conf
```

Output plug-in jar directory

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/logstash-scala/logstash/outputs
```

Secure Shell (ssh) configuration for remote logstash

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

The `ssh_config.properties` file is in the <HOME>/IBM/LogAnalysis/ remote_install_tool/config directory. Configure the parameter values as outlined in Table 1.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password-based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file-based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

To set up command-line authentication, rename the ssh-config properties file or move the properties file to a new location. By default the configurations are selected from the properties file. If the file is unavailable, the user is prompted for command-line input.

Table 84. ssh_config parameters

Parameter	Value
`REMOTE_HOST=`	*<REMOTE SERVER IP/FQ HOSTNAME>*
`PORT=`	*<SSH PORT>* THE DEFAULT VALUE IS 22
`USER=`	*<SSH_USER>*
`PASSWORD=`	*<SSH PASSWORD>*

logstash configuration

logstash can be configured as a log file agent to ingest logs from a number of different sources.

About this task

There are two established use cases for using logstash with IBM Operations Analytics - Log Analysis, these are:

- Configuring logstash as an alternative to ITM LFA
- Configuring logstash for rapid annotation and pre-indexing processing

Both use cases are described in this section.

Configuring logstash as an alternative to ITM LFA

logstash can be used as a log file agent to ingest logs from a number of different sources. It can also support integration with numerous alternative log file agents such as Lumberjack, Minuswell, Beaver, and Syslog.

About this task

Log records are written to the IBM Operations Analytics - Log Analysis that then sends the message to the IBM Operations Analytics - Log Analysis server for annotating and indexing.

Procedure

1. Update the logstash sample configuration file, `logstash/config/logstash-scala.conf`, with your configuration information.
   a. Define the input in the logstash configuration file.

      For example:

      ```
      input {
        file {
         type => "http"
         path => ["/tmp/myhttp.log"]
        }
       }
      ```

      Note: For Windows, the logstash file plug-in requires a drive letter specification for the path, for example:

      ```
      path => ["c:/tmp/myhttp.log"]
      ```
   b. Modify the logstash configuration file to add the `scala` output plug-in.

      The `scala` output plug-in buffers and sends the logstash event to the IBM Operations Analytics - Log Analysis server by using the Log Analysis server ingestion REST API. The logstash configuration file can contain one or more `scala` output plug-ins. The output plug-ins can be configured to write to different Log Analysis servers or to the same Log Analysis server with a different set of configurations.

      Every event that is sent to the `scala` output plug-in must contain at least the `host` and `path` fields. The values of these fields are used by the `scala` output plug-in to determine the target data source for the event. Any event that does not contain either of these fields is dropped by the output plug-in.

      The following are the default parameters, with sample values, for the IBM Operations Analytics - Log Analysis `scala` output plug-in:

      ```
      output {
        scala {
          scala_url => "https://<la_server>:<port>/Unity/DataCollector"
          scala_user => "<LA_user>"
          scala_password => "<encrypted_pwd>"
          scala_keystore_path => "<install-dir>/LogAnalysis/store/unity.ks"
          batch_size => 500000
          idle_flush_time => 5
          sequential_flush => true
          num_concurrent_writers => 20
          use_structured_api => false
          disk_cache_path => "<install-dir>/LogAnalysis/Logstash/cache-dir"
          scala_fields =>
            {
             "host1@path1,host2@path2"
                => "event_field11,event_field12,...,event_field1N"
             "host3@path3"
                => "event_field21,event_field22,...,event_field2N"
      ```

```
    }
  date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
  log_file => "<install-dir>/LogAnalysis/Logstash/logs/scala_logstash.log"
  log_level => "info"
 }
```

Where:

- **scala_url** is the REST endpoint for the Log Analysis ingestion REST API.
- **scala_user** is the Log Analysis user name.
- **scala_password** is the Log Analysis user password.
- **scala_keystore_path** is the path to the Log Analysis keystore on the file system.
- **batch_size** is the maximum number of bytes that can be buffered for a data source before transmitting to the Log Analysis server. The default is *500000* bytes.

  Note: Significantly decreasing the batch size impacts on throughput. Increasing the batch size requires more heap memory.
- **idle_flush_time** is the maximum time between successive data transmissions for a data source.
- **sequential_flush** defines whether batches for each data source are sent sequentially. It is set to *true* to send the batches sequentially.

  Note: Sequential sending is required when the input contains multi-line records that are combined in an Insight Pack in the Log Analysis server.
- **num_concurrent_writers** is the number of threads that concurrently transmit batches of data to the Log Analysis server.
- **use_structured_api** determines whether data is transmitted to the Log Analysis server in the JSON format. It is set to *true* to transmit data in the JSON format.

  Note: The target Log Analysis data source must be associated with a source type that uses the Log Analysis structured API.
- **disk_cache_path** is the path on the file system that temporarily buffers data. The scala output plug-in writes data to this path before transmission. The available disk space under the path must be large enough to store bursts of input data that is not immediately handled by the Log Analysis server.
- **scala_fields** is the map that specifies the names of fields that must be retrieved from the incoming logstash event and transmitted to the Log Analysis server. The keys for the map are a comma-separated list of host and path names that correspond to a Log Analysis data source.

  The scala plug-in extracts the host and path fields from each event before consulting the **scala_fields** map for a host and path combination entry. If there is an entry with field names, the scala plug-in extracts the corresponding field values from the event. The values are transmitted to the Log Analysis server. If the host and path entries are not in the **scala_fields** map, the scala plug-in extracts the contents of the message field from the event and transmits it to the Log Analysis server.
- **date_format_string** is the string value that all fields are transformed to before transmission to the Log Analysis server. The scala plug-in uses the **date_format_string** parameter to convert date values to the appropriate string value.

- **log_file** is the file that is used for logging information from the scala output plug-in.
- **log_level** is the level of logging information. The supported levels are fatal, error, warn, info, and debug.

2. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

   Ensure that the **File Path** matches the path that is specified in the logstash configuration file, logstash-scala.conf.

   Ensure that the **Type** matches the type of log file that is being ingested, for example **DB2Diag**.

   For example, if you specified /tmp/myhttp.log as an input file, then create a custom data source with path set to /tmp/myhttp.log.

What to do next

Start logstash as described in Starting logstash

Configuring logstash for rapid annotation and pre-indexing processing

logstash can be used to split log records and do basic annotation. For log types not currently supported by IBM Operations Analytics - Log Analysis, this is an alternate approach to writing AQL to annotate log files.

About this task

logstash includes a broad list of filtering, manipulation, and processing capabilities, for example, the grok filter can be used to parse text into structured data. It allows you to match text without the need to master regular expressions. There are approximately 120 grok patterns shipped by default, though you can add more. It also includes patterns for known log file formats, such as Apache's combined access log format.

In this scenario, logstash is basically used as the splitter/annotator of the log file by leveraging the grok filter. The scala_custom_eif output plugin sends a single log record to the IBM Operations Analytics - Log Analysis EIF Receiver, with the annotations in a delimiter separated value (DSV) format. Then, using the DSV Toolkit, the user must create and install an insight pack that matches the DSV format so that IBM Operations Analytics - Log Analysis can index the annotations. Please follow these steps:

Procedure

1. Update the logstash sample configuration file, logstash/config/logstash-scala.conf, with your configuration information.

   a. Define the input in the logstash configuration file.

      For example:

      ```
      input {
        file {
         type => "apache"
         path => ["/tmp/logs/myapache.log"]
        }
       }
      ```

      Note: For Windows, the logstash file plugin requires a drive letter specification for the path, for example:

      ```
      path => ["c:/tmp/myapache.log"]
      ```

b. Modify the logstash configuration file to add the `scala_custom_eif` output plugin.

c. Add a filter or filters to the logstash configuration file to identify the pattern of the log file format. This also creates the annotations. To trigger the filter, the type must match the input type.

   For example:

```
filter {
    if [type] == "http" {
        grok {
            match => ["message", "%{IP:client} %{WORD:method}
%{URIPATHPARAM:request}  %{NUMBER:bytes} %{NUMBER:duration}"]
                      }
    }
}
```

   In this example, the fields client, method, request, bytes, and duration are annotated by the pattern. However, only the fields client, method and request are sent to IBM Operations Analytics - Log Analysis. Thus, those are the only three annotations that can be included in the index configuration. The output module sends the event text in DSV format as:

```
"client", "method", "request"
```

   The user can also use one of the many predefined grok log format pattern such as:

```
filter {
    if [type] == "apache" {
        grok {
            match     => ["message", "%{COMBINEDAPACHELOG}"]
        }
    }
}
```

2. Create an IBM Operations Analytics - Log Analysis DSV-generated Insight Pack in order to index the annotated data in IBM Operations Analytics - Log Analysis.

   The `lsartifact/dsvProperties` directory contains a sample property file that can be used to generate an Insight Pack that ingests delimiter separated log records that are already formatted for Apache combined access log files. Use the DSV toolkit, which is available at `<HOME>/IBM/LogAnalysis/unity_content/tools`, to generate an Insight Pack from the DSV properties file. This means the user must configure the logstash configuration file, `/lstoolkit/logstash/config/logstash-scala.conf`, with the appropriate grok filter to enable the IBM Operations Analytics - Log Analysis output plugin to generate the comma delimited logs. For example, uncomment the apache grok filter in the `logstash-scala.conf` file and generate an Insight Pack using `ApacheDSV.properties` with the DSV tooling script. The `scala` plugin will generate a comma delimited event based on the grok filter that can be ingested (annotated and split) by the generated Insight Pack.

   Note: The path to `logstash-scala.conf` is dependent on where you copied the `lstoolkit` directory on the logstash server (see step 3 of Installing the logstash Integration Toolkit).

3. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

   Ensure that the **File Path** matches the path that is specified in the logstash configuration file, `logstash-scala.conf`.

   Ensure that the **Type** matches the type of log file that is being ingested, for example **DB2Diag**.

For example, if you specified /tmp/myhttp.log as an input file, then create a custom data source with path set to /tmp/myhttp.log.

What to do next

Start logstash. For more information on starting logstash, see "logstash operations" on page 276 in the *Installing logstash* section of the *Loading and streaming data* guide.

Example - Annotating Combined Apache log files:

Using logstash to annotate Apache log files.

Procedure

1. Edit your logstash configuration file. A sample is provided in logstash-scala.conf.
   a. In the input section, specify the Apache log file to be monitored.

   ```
   input {
     file {
      type => "apache"
      path => ["/tmp/apache.log"]
     }
    }
   ```

   b. Add the logstash grok filter with the predefined COMBINEDAPACHELOG pattern to annotate the Apache log files.

   For example:

   ```
   filter {
    if [type] == "apache" {
     grok {
        match => ["message", "%{COMBINEDAPACHELOG}"]
     }
    }
   }
   ```

   The COMBINEDAPACHELOG pattern is defined as:

   ```
   COMBINEDAPACHELOG %{IPORHOST:clientip} %{USER:ident} %{USER:auth}
   \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}
   (?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response}
   (?:%{NUMBER:bytes}|-) %{QS:referrer} %{QS:agent}
   ```

   For more information about Apache log files, see http://httpd.apache.org/docs/2.4/logs.html.

   The logstash event contains annotations for clientip, ident, auth, timestamp, verb, request, httpversion, rawrequest, response, bytes, referrer, and agent.

   c. In the output section of the configuration file, specify the IBM Operations Analytics - Log Analysis output plug-in.
2. The logstash Integration Toolkit provides a properties file, lsartifact/dsvProperties/ApacheDSV.properties, which can be used with the DSV Toolkit to create an Apache Insight Pack. Edit this properties file to configure information about your IBM Operations Analytics - Log Analysis server:

   ```
   [SCALA_server]
    username: unityadmin
    password: unityadmin
    scalaHome: $HOME/IBM/LogAnalysis
   ```

3. Use the dsvGen.py script that is provided with the DSV Toolkit to generate and deploy the Apache Insight Pack:

   ```
   python dsvGen.py <path>/ApacheDSV.properties -d
   ```

4. In the IBM Operations Analytics - Log Analysis Administrative Settings UI, create a data source, which has the Apache source type that is created by the DSV toolkit in step 4, in your logstash configuration file.
5. Start logstash with the configuration file, and start ingesting Apache log files.

logstash operations

You can use the `logstash-util` script to start, stop, restart, or provide the status of logstash.

About this task

You can use the `logstash-util` script for logstash process lifecycle management.

Procedure

1. To start, stop, restart, or provide the status of logstash, run the following command:

   ```
   <install-dir>/LogAnalysis/utilities/logstash-util.sh start| stop| restart| status
   ```

   where *<install-dir>* is the name of the logstash installation location.
2. To confirm that logstash is running, run the `logstash-util` script and use the `status` option. The `status` option also displays the logstash process identifier.

logstash best practices

Best practices for logstash based on information from their user community.

For performance reasons it is recommend that logstash be installed on a different server than IBM Operations Analytics - Log Analysis. logstash is processor, memory, and disk intensive if the annotation and indexing functions are utilized.

Users who have memory constraints do not use logstash as a forwarding agent. They do not install logstash on the end client servers. They use other applications such as rsyslog to forward logs to a central server with logstash. See https://support.shotgunsoftware.com/entries/23163863-Installing-logstash-Central-Server for an example configuration.

Users with logstash at the end client who are concerned about performance have used applications such as Redis to forward logs to a central server with logstash. See the following for configuration of Redis http://www.linux-magazine.com/Online/Features/Consolidating-Logs-with-logstash .

To fine tune logstash, especially the startup time, users can tweak Java's minimum and maximum heap size with the -Xms and -Xmx flags. The -Xms parameter is the initial Java memory heap size when the JVM is started, and the -Xmx parameter is the maximum heap size.

References

Links for more information on the logstash application.

logstash website:
http://logstash.net

Getting Started with logstash Guide:
http://logstash.net/docs/1.4.2/tutorials/getting-started-with-logstash

logstash Download:
http://logstash.net (Click download button)

The logstash Book:
http://www.logstashbook.com/

IBM Operations Analytics - Log Analysis wiki:
http://www.ibm.com/developerworks/servicemanagement/ioa/log/downloads.html

IBM Operations Analytics - Log Analysis wiki: Logstash Toolkit Resources:
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Logstash%20Toolkit%20Resources

Known issues

Known issues when using logstash with IBM Operations Analytics - Log Analysis.

There are a number of known issues and their workarounds described in this section. To get the latest information on any issues or workarounds, please consult the IBM Operations Analytics - Log Analysis wiki:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Welcome

Could not load FFI Provider

Starting logstash fails with the Ruby exception “Could not load FFI Provider”.

Symptoms

The Ruby exception “Could not load FFI Provider”.

Causes

The most common cause of this error is that /tmp is mounted with the **noexec** flag.

Resolving the problem

You can resolve this either by:

- Making /tmp mounted without the **noexec** flag
- Edit the startlogstash-scala script and amend the start command as follows:

  ```
  LSCMD="$MYJAVA -jar -Djava.io.tmpdir=</some/tmp/dir> $LSJAR agent
  --pluginpath $PLUGPATH -f $CONF"
  ```

  Where </some/tmp/dir> is a temporary directory.

Duplication of log records on the SCALA server

On occasion, when the logstash agent is re-started, and the log file being monitored is updated (for example, via a streaming log), logstash will ingest the entire file again rather than restarting from where it stopped monitoring.

Symptoms

The problem results in a duplication of log records on the SCALA server.

Causes

Several problems have been reported on the logstash forum (https://logstash.jira.com/secure/Dashboard.jspa) that its sincedb pointer (which tracks the last monitored position in the log file) sometimes is not updated correctly. In addition, using control-C to terminate the logstash agent does not always kill logstash. The result is a "phantom" logstash agent that is still monitoring log files. This can also result in duplicate log records.

Resolving the problem

1. A workaround to avoid duplicate log records after restarting logstash is to set the **sincedb_path** parameter in the file plugin to /dev/null, thereby telling logstash to ignore tracking the last-monitored file position, and always start monitoring from the end of the file. However, this will result in logstash ignoring any updates to the log file while the logstash agent is down. For example, in logstash-scala.conf, update:

   ```
   input {
       file {
           type => "apache"
           path => ["/tmp/logs/myapache.log"]
           sincedb_path => "/dev/null"
       }
   }
   ```

   Before re-starting logstash after making these configuration changes, you may also want to clean up any sincedb databases that were already created. By default, the sincedb database is stored in the directory $HOME, and have filenames starting with ".sincedb_".
2. When terminating the logstash agent using control-C, verify that the logstash java process was actually terminated. You can use the following command to see if logstash is still running:

   ```
   ps -ef | grep logstash
   ```

Logs do not appear in the Search UI

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Symptoms

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Causes

Log records ingested by logstash are forwarded to the IBM Operations Analytics - Log Analysis server for splitting and annotating, and indexing. If the IBM Operations Analytics - Log Analysis server goes down during this process, it is possible to lose some log records.

Administering

Read this section to understand how to use Log Analysis to administer the data model and other aspects of Log Analysis.

Getting started with Log Analysis

Complete these tasks to help you to get started with Log Analysis.

Logging in to IBM Operations Analytics - Log Analysis

This topic outlines how to log in to IBM Operations Analytics - Log Analysis and how to change the default username and password.

By default, the `unityuser` user is assigned the `UnityUser` role. Log in as this user to access the Search workspace. The `UnityAdmin` role allows you to access the Search workspace and also the administrative workspaces. This role is assigned to the `unityadmin` user. The default passwords for each of these users is the same as the username.

You can change the default passwords for each of these users by changing the value in the basic user registry. For more information, see the *Creating roles, groups, and users in the file-based user registry* section of the documentation.

To log in to the IBM Operations Analytics - Log Analysis administrative workspaces:

1. In a web browser, type the address: `http://`*`ip_address`*`:9988/Unity/Admin`. If you use SSL communication, the address is `https://`*`ip_address`*`:9987/Unity`.
2. When prompted, type the username and password for a user with administrator access permissions and click **Go**.

To log in to the IBM Operations Analytics - Log Analysis Search workspace:

1. In a web browser, type the address: `http://`*`ip_address`*`:9988/Unity`. If you use SSL communication, the address is `https://`*`ip_address`*`:9987/Unity`.
2. When prompted, type the username and password for a user with administrator or user access permissions and click **Go**.

Note: If the 9988 or 9987 port is blocked by a firewall, IBM Operations Analytics - Log Analysis might not display. Check the firewall status and unblock the port where necessary.

Note: If you are running IBM Operations Analytics - Log Analysis over a slow network, a warning might be displayed indicating that a script is unresponsive script. To proceed, click **Continue**.

To log in to IBM Operations Analytics - Log Analysis on a Dynamic Host Configuration Protocol (DCHP) server, use the Fully Qualified Domain Name (FQDN) or the host name to log in. You cannot use the IP address as you would for non-DHCP servers. For more information, see the *Cannot display custom apps on Dynamic Host Configuration Protocol (DHCP) server* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Installing sample files

This topic outlines how to load sample artifacts and data so that you can get started using IBM Operations Analytics - Log Analysis quickly and to allow you to understand how you can apply IBM Operations Analytics - Log Analysis to your own environment.

About this task

When you load the sample data, sample IBM Operations Analytics - Log Analysis artifacts, sample data, and sample Custom Apps are loaded. In addition, a number of sample Custom Apps are loaded:

- **sample-Web-App**: This Custom App displays a sample dashboard of a typical web application built using web servers, an application server, and a database.
- **sample-WAS-Troubleshooting**: This Custom App displays a sample dashboard for WebSphere Application Server SystemOut logs from multiple servers.
- **sample-events-hotspots**: This Custom App displays a sample event analysis dashboard built for sample IBM Tivoli Netcool/OMNIbus events.

Note: If you enable Lightweight Directory Access Protocol (LDAP) authentication and you want to load the sample data, you must create a user called unityadmin that is assigned to a group called unityadmin. If you do not, you cannot load the sample data.

Procedure

1. Log into the Search workspace: https://*<ipaddress>*:*<port>*/Unity where *<port>* is the port specified during installation for use by the web console. The default value is 9987. The default administrative username and password are `unityadmin` and `unityadmin` respectively.
2. On the **Getting Started** page, click **Install Sample Data** > **Start Now**. The sample data loads.

Enabling the GUI search history

You can enable history for the search on the IBM Operations Analytics - Log Analysis UI.

About this task

IBM Operations Analytics - Log Analysis uses a cookie that is saved in the temporary directory of the browser to remember the last 10 terms that are entered in the search field.

To view the last 10 search results, select the search list. To add a term to the list of searches, enter the term in the search field and click **Search**.

The cookie that saves the search terms expires every 30 days by default.

Procedure

To enable search history for the IBM Operations Analytics - Log Analysis UI, you must ensure that your browser settings are configured to save cookies.
To clear your search history, delete the saved cookies for your browser.

Defining a default search

If you want to define a default initial search that is displayed when you log into the Search workspace, complete the steps in this topic.

About this task

You can define the search query. However, any additional parameters such as the Data Source or Time filter cannot be defined for your default search. To define your search:

Procedure

1. Open the `unitysetup.properties` file located in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
2. Locate and, if necessary, remove the comment notation from the line:
   ```
   SEARCH_QUERY_FOR_DEFAULT_SEARCH=*
   ```
3. Add the text that you want to define as your default query: For example:
   ```
   SEARCH_QUERY_FOR_DEFAULT_SEARCH=TUNE9001W
   ```
4. Save the file.
5. Use the following command to restart IBM Operations Analytics - Log Analysis:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Results

If there are search results in the last 15 minutes for the specified query, the results of the query are displayed. If you have defined a value for the search and no results are found, a message is displayed indicating that no matches were found.

Creating and updating the data model

Before you load data into Log Analysis, you must create the data sources, source types, index configuration, and physical source IDs required to model your data ingestion, indexing, annotation, and grouping of data.

You also need to add and delete objects from you model.

Data Sources workspace

The Data Sources workspace allows you to define your data sources. This information is used to process the content of the log file and facilitate search by converting the file contents into a structured format.

Data Source creation

You create data sources to ingest data from a specific source.

Before you can search a log file, you must use the data source creation wizard to create a data source. To start the wizard, open the Admin UI, click **Add** > **Data Sources**. The wizard contains text to help you to complete the task.

You can create three types of data source. Select one of the following types:

- To stream data from a file that is stored locally on the IBM Operations Analytics - Log Analysis server, create a local data source. The IBM Tivoli Monitoring Log File Agent on the same server is automatically configured to retrieve local data.

- To stream data from a file that is stored on a remote server, create a remote data source. Enter a host name, user name, and password. The IBM Tivoli Monitoring Log File Agent on the same server is automatically configured to pull data from a remote server.

  Note: If the remote server uses the Windows operating system, this setting requires that the ssh daemon is configured on the remote server.
- To stream data from a file that is stored on a remote IBM Operations Analytics - Log Analysis server where data delivery is manually configured, or where data ingestion is configured for logstash or the Data Collector client, create a custom data source. Enter a host name. There is no automatic configuration or verification.

If you select local or remote as the location, IBM Operations Analytics - Log Analysis creates a new collection automatically. If you select custom as the location, you can choose to associate it with an existing collection or you can create a new collection.

The log file specified for the **File path** field is automatically ingested into IBM Operations Analytics - Log Analysis for data sources with **Local file** or **Remote file** configuration options.

If you want to ingest rolling files, click the **Rolling file pattern** check box and enter the rolling file pattern. For example, for a WebSphere Application Server log file the base line file is `SystemOut.log`. The rolling file pattern is `SystemOut_*.log`. If you use this setting, the data latency depends on how frequently the files are rolled in the source application.

To complete the process, enter a name for the data source.

You can also edit an existing data source.

Database Connections

A Database Connection uniquely identifies a source of structured data in IBM Operations Analytics - Log Analysis.

You can associate semi-structured data, such as a log, with structured data, such as the transaction status in the application database. To make this association, you must define a Database Connection, which specifies the details required by IBM Operations Analytics - Log Analysis to access the database. After you have defined a Database Connection, you can use it when you configure any Application that you require.

Adding a Database Connection:

This topic outlines the steps that you must follow to configure an events database as a Database Connection.

Procedure

To add a Database Connection:

1. In the Data Sources workspace, click **Add** > **Database Connections**. The **Add Database Connections** tab is displayed
2. You are prompted to supply the following information:

   Name The name of the Database Connection.

Schema Name
The name of the table containing the event, for example, `MISC`.

JDBC URL
The IP address and path for the events database, for example, `jdbc:derby://`*`ip_address`*`:1627//opt/unity/database/UnityDB`.

JDBC Driver
The JDBC driver, typically `org.apache.derby.jdbc.ClientDriver`.

Username
Type the username for the Database Connection.

Password
Type the password for the Database Connection.

3. Click **OK**. A new entry is displayed in the **Database Connections** list.

Editing a Database Connection:

You can edit the details of an existing Database Connection.

Procedure

To edit a Database Connection:

1. In the Data Sources workspace, expand the **Database Connections** list.
2. Select the Database Connection that you want to edit and click **Edit**. The Database Connection is opened in a new tab.
3. Edit the data source as required.
4. To save the changes, click **OK**.

Deleting a Database Connection:

You can delete an existing Database Connection.

Procedure

To delete a Database Connection:

1. In the Data Sources workspace, expand the **Database Connections** list.
2. Select the Database Connection that you want to delete and click **Delete**.
3. Click **OK**

Data Types workspace

The Data Types workspace allows you to configure the type of log information that is consumed.

Collections

Collections allow you to group together log data from different data sources that have the same source type. For example, you might want to assign all the data sources for a WebSphere Application Server cluster into a single Collection so that you can search them as a group. This section outlines how to manage your Collections.

Adding a Collection:

This topic outlines the steps you must follow to add a Collection to IBM Operations Analytics - Log Analysis.

Procedure

To add a Collection:

1. In the Data Types workspace, click **Add** > **Collection**. The **Add Collection** tab is displayed
2. Enter the details for the collection that you want to add:

   Name Provide a unique name for the Collection.

   Source Type
   From the list, select the type of log file data that you want the Collection to contain.
3. Click **OK**. A message is displayed requesting that you confirm that you want to create the Collection and indicating that you cannot edit the Source Type property after the Collection is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a Collection:

After a Collection has been created, you cannot edit the properties associated with the Collection. Use the **Edit** button to review the existing properties for a Collection.

Procedure

To view a collection:

1. In the Data Types workspace, expand the **Collections** list.
2. Select the Collection that you want to view and click **Edit**. The Collection is opened in a new tab.
3. To close the Collection, click **Close**.

Deleting a Collection:

You can delete an existing Collection.

Procedure

To delete a Collection:

1. In the Data Types workspace, expand the **Collections** list.
2. Select the Collection that you want to delete and click **Delete**. A message is displayed listing the data sources that are associated with the Collection. These data sources are deleted when the Collection is deleted. All data associated with the Collection is deleted when the Collection is deleted.
3. Click **Delete**.

Source Types

A Source Type defines how data of a particular type is processed by IBM Operations Analytics - Log Analysis. This determines how data is indexed, split, and annotated.

Index configuration:

You can use the index configuration of a Source Type to provide important information about how data of that Source Type is indexed in IBM Operations Analytics - Log Analysis.

Index configuration is specified using JSON configuration notation. When creating a source type, you can specify a file containing JSON configuration notation, or you can enter the configuration directly to the **Index Config** field.

To index data appropriately, different types of data require different index configuration.

Index configuration example: PMI data:

This sample JSON index configuration can be used to index PMI data:

```
{ "indexConfigMeta":{
  "name": "PMIConfig",
  "description": "IndexMappingforPMIdata",
  "version": "0.1",
  "lastModified": "9/10/2012"
  },
 "fields": {
  "hostname": {
  "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": false,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.hostname"]
   }
  },
 "sourceip": {
  "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": true,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.sourceip"]
   }
  },
 "logpath": {
  "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": false,
  "sortable": false,
  "filterable": false,
  "searchable": false,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.logpath"]
   }
  },
 "logsource": {
  "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": false,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.logsource"]
   }
  },
"timeformat": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": false,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.timeformat"]
  }
 },
"description": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": false,
 "sortable": false,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.description"]
  }
 },
"PoolSize": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.poolsize"]
  }
 },
"FreePoolSize": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.FreePoolSize"]
  }
 },
"WaitingThreadCount": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.WaitingThreadCount"]
  }
 },
"PercentUsed": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.PercentUsed"]
  }
 },
"UseTime": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.UseTime"]
  }
 },
"WaitTime": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.WaitTime"]
  }
 },
"HeapSize": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.HeapSize"]
  }
 },
"UsedMemory": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.UsedMemory"]
  }
 },
"UpTime": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.UpTime"]
  }
 },
"ProcessCpuUsage": {
 "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": true,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.ProcessCpuUsage"]
   }
  },
 "CPUUsageSinceLastMeasurement": {
  "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": true,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.CPUUsageSinceLastMeasurement"]
   }
  },
 "webcontainerthreads": {
  "dataType": "TEXT",
  "retrievable": true,
  "retrieveByDefault": true,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.webcontainerthreads"]
   }
  },
 "timestamp": {
  "dataType": "DATE",
  "retrievable": true,
  "retrieveByDefault": true,
  "sortable": true,
  "filterable": true,
  "searchable": true,
  "tokenizer": "literal",
  "source": {
   "paths": ["metadata.timestamp"],
   "dateFormats": ["dd/MM/yyyy HH:mm:ss.SSS"]
   }
  }
 }
}
```

Index configuration example: Expert advice data:

This sample JSON index configuration can be used to index expert advice data:

```
{ "sourceTypeClass": 0,
 "name": "Test",
 "splitRuleSet": null,
 "extractRuleSet": null,
 "indexConfigMeta": {
  "name": "Test",
  "description": "Testing UVI Integration",
  "version": "0.1",
  "lastModified": "17/08/2012"
  },
 "fields": {
  "hostname": {
   "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.hostname"]
  }
 },
"timestamp": {
 "dataType": "DATE",
 "dateFormat": "dd/MM/yyyy HH:mm:ss.SSS",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.timestamp"],
  "dateFormats": ["dd/MM/yyyy HH:mm:ss.SSS"],
  "combine": "FIRST"
  }
 },
"logpath": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": false,
 "sortable": false,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.logpath"]
  }
 },
"regex_class": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": false,
 "sortable": false,
 "filterable": false,
 "searchable": false,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.regex_class"]
  }
 },
"Hostname": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
 "searchable": true,
 "tokenizer": "literal",
 "source": {
  "paths": ["metadata.Hostname"]
  }
 },
"service": {
 "dataType": "TEXT",
 "retrievable": true,
 "retrieveByDefault": true,
 "sortable": true,
 "filterable": true,
   "searchable": true,
   "tokenizer": "literal",
   "source": {
    "paths": ["metadata.Service"]
    }
   },
  "logsource": {
   "dataType": "TEXT",
   "retrievable": true,
   "retrieveByDefault": true,
   "sortable": true,
   "filterable": true,
   "searchable": true,
   "tokenizer": "literal",
   "source": {
    "paths": ["metadata.logsource"]
    }
   },
  "logRecord": {
   "dataType": "TEXT",
   "retrievable": true,
   "retrieveByDefault": true,
   "sortable": true,
   "filterable": true,
   "searchable": true,
   "tokenizer": "literal",
   "source": {
    "paths": ["metadata.text"]
    }
   },
  "className": {
   "dataType": "TEXT",
   "retrievable": true,
   "retrieveByDefault": false,
   "sortable": false,
   "filterable": true,
   "searchable": false,
   "tokenizer": "literal",
   "source": {
    "paths": ["annotations.ClassnameWSOutput.text"]
    }
   }
 }
}
```

Physical Source IDs and Source Types:

To help you to identify the physical source of data that is loaded from logical data sources, you can add a physical source ID to each source type.

If you want to use the Data Ingestion dashboard, you need to configure a Physical Source ID for each Source Type that you want to monitor with the dashboard. You can use the identifiers that are specified in the properties file to display statistics for specific host names, log paths, and Physical Source IDs.

For example if you are loading data from a number of remote servers, you can use the Physical Source ID to identify the data that is loaded from a specified machine.

Enabling the Physical Source ID

The physical source ID is composed of three identifiers, which you can assign to a source type. This type of data source is different than the logical data source, which you can create in IBM Operations Analytics - Log Analysis UI.

The identifiers are specified in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties file. For example, by default the physical ID includes identifiers for the host name, instance ID, and the log path:

```
DATA_INGESTION_STATS_LABELS_KEY_COUNT=3
DATA_INGESTION_STATS_LABEL_KEY_1=HOSTNAME
DATA_INGESTION_STATS_LABEL_KEY_2=INSTANCE ID
DATA_INGESTION_STATS_LABEL_KEY_3=LOG PATH
```

where DATA_INGESTION_STATS_LABELS_KEY_COUNT=3 is the number of label key identifiers that you want to display on the dashboard. DATA_INGESTION_STATS_LABEL_KEY_N is individual key label identifier. You can use this to specify parameters such as the host name, an instance ID, and the log path.

You can add or remove these identifiers. If you do so, remember to update the DATA_INGESTION_STATS_LABELS_KEY_COUNT=3 parameter to match the number of key labels.

Configuring the granularity

The DATA_INGESTION_STATS_GRANULARITY parameter determines the interval at which the physical source ID is updated. To specify the interval, edit the following property in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties file. For example:

```
DATA_INGESTION_STATS_GRANULARITY=15MINUTES
```

The default value is 15 minutes. This setting means that the log records displayed for the specified Physical Source ID are updated every 15 minutes.

Configuring the data retention period

The DATA_INGESTION_STATS_RETENTION_PERIOD parameter specifies the amount of time that the Physical Source ID is saved for statistically use. To specify the time for which data is retained, edit the following property. The period can be 1DAY, 30DAYS, 1MONTH, or 6MONTHS:

```
DATA_INGESTION_STATS_RETENTION_PERIOD=30DAYS
```

In this example, the data is stored for 30 days before it is deleted. In other words, all data older than 30 days is deleted.

Adding a Physical Source ID

1. Add the key label identifiers that you want to display on the dashboard to the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties file.
2. Log in to IBM Operations Analytics - Log Analysis and open the **Data Types** workspace.
3. Create a new Source Type or edit an existing Source Type.
4. Click **Index Configuration** and click **Add Physical Source ID**.
5. Enter the parameters that you specified in step 1 and save your changes.
6. Open the Data Ingestion dashboard and configure the widgets to display the key labels that you specified in the Physical Source ID.

Log annotator:

You can use annotations to extract and identify pieces of information from unstructured text, for example, IP address, host name, and time stamp. Doing this allows you to search for specific types of information, rather than being limited to a simple text search of the log files.

You can search annotated information more efficiently. For example, you can search annotated information for log entries where the severity is `E`. If this information had not been annotated, you could perform only a simple text search for instances of the letter `E`, which might return irrelevant matches.

To identify annotations in a log file or configuration file, use rules specified in Annotation Query Language (AQL). Annotation can be configured for each data source type.

These annotator types are available:

Default
When using the default annotator type, you must specify a rule for splitting the text into different fields and a rule for extracting values from the identified fields.

Java When using the Java annotator type, annotations are made to data sources of this type based on the annotator Java class you specify. The `CustomAnnotatorImpl.class` class in `annotator.jar` is provided for this purpose.

Script When using the Script annotator type, annotations are made to data sources of this type based on the annotator script you specify. The `DefaultPythonAnnotator.py` script is provided for this purpose.

None No annotations are made to data sources of this Source Type.

The Java and Script annotator types are provided as customization options for the default annotator type.

Adding a Source Type:

This topic outlines the steps that you must complete to add a Source Type to IBM Operations Analytics - Log Analysis.

Procedure

To add a Source Type:

1. In the Data Types workspace, click **Add** > **Source Type**. The **Add Source Type** tab is displayed
2. Provide values for each of the required fields to create an appropriate Source Type to meet your requirements:

   Name Provide a unique name for the source type.

   Input type
   Provide the type of data that is processed.

   Enable splitter:
   Click to enable this option and select the Rule Set or File Set that is used to split the log records in the log files processed:

Rule Set: Choose from the options that are provided or add a Rule Set before selecting it.

File Set: Select the File Set option if you want to use a set of custom rules.

Enable annotator
Click to enable this option and select the Rule Set or File Set that is used to annotate the log records in the log files processed:

Rule Set: Choose from the options that are provided or add an additional Rule Set before selecting it.

File Set: Select the File Set option if you want to use a set of custom rules.

Deliver data on annotator execution failure: Select this option if you want records that fail during annotation to be added.

Index Config
Click **Edit Index Configuration** and specify the index configuration that is used to determine how annotated data is indexed.

Add physical source ID
If you want to view statistics for physical sources of data on the Data Ingestion dashboard in the Analytics Console, you need to specify a Physical Data Source ID. For more information, see the *Physical Data Source IDs* topic in the *Analytics Console* section of the IBM Operations Analytics - Log Analysis documentation.

3. Click **OK**. A message is displayed requesting that you confirm that you want to create the Source Type and indicating that you cannot edit some properties after the Source Type is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a Source Type:

After a Source Type has been created, you cannot edit the properties associated with the Source Type. Use the **Edit** button to review the existing properties for a Source Type.

Procedure

To edit a Source Type:

1. In the Data Types workspace, expand the **Source Types** list.
2. Select the Source Type that you want to view and click **Edit**. The Source Type is opened in a new tab.
3. To close the Source Type, click **Close**.

Deleting a Source Type:

You can delete an existing Source Type. However, a Source Type can only be deleted if it is not referenced by a Collection. Artifacts must be deleted in a specific order. The Collection must be deleted first, then the Source Type, and finally any Rule Set or File Set associated with the Source Type.

Procedure

To delete a Source Type:

1. In the Data Types workspace, expand the **Source Types** list.
2. Select the Source Type that you want to delete and click **Delete**.
3. Click **OK**.

Rule Set

A Rule Set is a file containing rules specified in Annotation Query Language (AQL). There are two available types of Rule Set:

Split Used to split unstructured or semi-structured data into different fields.

Annotate
Used to annotate the fields already identified in a file containing unstructured or semi-structured data.

Adding a Rule Set:

This topic outlines the steps that you must complete to create a Rule Set.

Procedure

To add a Rule Set:

1. In the Data Types workspace, click **Add** > **Rule Set**. The **Add Rule Set** tab is displayed
2. Enter the required information in each of the fields. Place your cursor on each of the fields for information about the requirements of that field.
3. Enter the following details for the Rule Set that you want to add:

   Name Provide a unique name for the rule set.

   Type Specify the type of rule set you want to create. The types available are `Split` and `Annotate`.

   Rule File Directory
   Specify the list of paths to the module source directories.
4. Click **OK**. A message is displayed requesting that you confirm that you want to create the Rule Set and indicating the properties that cannot be changed after the Rule Set is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a Rule Set:

After a Rule Set has been created, you cannot edit the name or type properties associated with the Rule Set.

About this task

You cannot edit the **Name** or **Type** associated with the Rule Set. Use the **Edit** button to review the existing properties for a Rule Set.

Procedure

To view the properties of a Rule Set:

1. In the Data Types workspace, expand the **Rule Sets** list.

2. Select the Rule Set that you want to view and click **Edit**. The Rule Set is opened in a new tab.
3. To close the Rule Set, click **Close**.

Deleting a Rule Set:

You can delete an existing Rule Set. However, a Rule Set can only be deleted if it is not referenced by a Source Type. Artifacts must be deleted in a specific order. Delete the Collection first, next delete the Source Type, and finally delete any Rule Set or File Set.

Procedure

To delete a Rule Set:

1. In the Data Types workspace, expand the **Rule Sets** list.
2. Select the Rule Set that you want to delete and click **Delete**.
3. Click **OK**.

File Sets

A File Set allows you to define a set of custom rules that are used to split or annotate a type of log data.

Adding a File Set:

This topic outlines the steps that you must complete to add a File Set.

Procedure

To add a File Set:

1. In the Data Types workspace, click **Add** > **File Sets**. The **Add File Set** tab is displayed
2. Enter the required information in each of the fields. Place your cursor on each of the fields for information about the requirements of that field.
3. Click **OK**. A message is displayed requesting that you confirm that you want to create the File Set and indicating the properties that cannot be changed after the File Set is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a File Set:

After a File Set has been created, you cannot edit the properties associated with the File Set.

About this task

You cannot edit the properties associated with an existing File Set. Use the **Edit** button to review the existing properties for a File Set.

Procedure

To view the properties of a File Set:

1. In the Data Types workspace, expand the **File Sets** list.
2. Select the File Set that you want to view and click **Edit**. The File Set is opened in a new tab.

3. To close the File Set, click **Close**.

Deleting a File Set:

You can delete an existing File Set. However, a File Set can only be deleted if it is not referenced by a Source Type. Artifacts must be deleted in a specific order. Delete the Collection first, next delete the Source Type, and finally delete any Rule Set or File Set.

Procedure

To delete a File Set:

1. In the Data Types workspace, expand the **File Sets** list.
2. Select the File Set that you want to delete and click **Delete**.
3. Click **OK**.

Administrative tasks

You can change the refresh rate for dashboards, modify the service topology file and configure the time settings as part of the administrative tasks which may need to complete from time to time.

Configuring automatic refreshes for new dashboards

You can use the auto-refresh feature to regularly refresh a dashboard at scheduled intervals.

About this task

You can configure automatic refreshes only for a dashboard if all the charts in the dashboard use a relative time filter.

The default maximum number of charts that can be refreshed simultaneously across all dashboards is 20. For example, you can refresh 10 dashboards simultaneously if each dashboard has two charts. To edit the maximum number of charts that can be refreshed simultaneously, edit the `MAX_AUTO_REFRESH_CHARTS=<20>` property at the following location: `<UNITY_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`, where <20> is the number of charts that can be refreshed simultaneously.

Auto-refresh supports intervals of 0, 1, 5, 15, 30, and 60 minutes.

Procedure

1. Open the dashboard.
2. To set the auto-refresh interval, click **Actions**, then **Auto-Refresh** and select the interval that you want to specify. For example, to refresh the dashboard every 5 minutes, click **5 minutes**.

Results

The dashboard and the associated charts are automatically refreshed with the most current data at the specified interval.

A check mark beside the specified interval indicates that an auto-refresh interval was already specified.

Configuring automatic refreshes for existing dashboards

You can use the auto-refresh feature to regularly refresh an existing dashboard at scheduled intervals.

About this task

You can configure automatic refreshes only for a dashboard if all the charts in the dashboard use a relative time filter.

The default maximum number of charts that can be refreshed simultaneously across all dashboards is 20. For example, you can refresh 10 dashboards simultaneously if each dashboard has two charts. To edit the maximum number of charts that can be refreshed simultaneously, edit the `MAX_AUTO_REFRESH_CHARTS=<20>` property at the following location: `<UNITY_HOME>/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`, where <20> is the number of charts that can be refreshed simultaneously.

Auto-refresh supports intervals of 0, 1, 5, 15, 30, and 60 minutes.

Procedure

Using a JSON editor, open and edit the original file with the following properties:

autoRefreshInterval:1
where <1> is the auto-refresh interval.

searchType: relative
where <`relative`> is the time filter.

Results

The dashboard and the associated charts are automatically refreshed with the most current data at the specified interval.

A check mark beside the specified interval indicates that an auto-refresh interval was already specified.

Editing groups in the service topology JSON file

If your services are provided using a web application that is deployed across a range of servers such as web servers, application servers, and database servers, you might want to define the scope and structure of your application or service. To represent your application or service hierarchy, you can create a group or edit the default groups in the IBM Operations Analytics - Log Analysis service topology JSON file.

About this task

After you create a service topology, you can then associate your data source with the appropriate node in the service topology. The hierarchy of data sources, reflecting your service topology, is shown in the **Data Sources** lists in the Data Sources workspace and the Search workspace. When you select a data source to narrow your search criteria, that data source, and any data sources in the service topology tree are searched.

To create a service topology that meets your requirements, edit the existing default service topology JSON file. The JSON file has structure with each node in the service topology defined by a type, a name, and a value. Data sources can only be assigned to hostname nodes.

A sample of the JSON file is displayed:

```
[
 {
  "type": "Service",
  "name": "Day Trader",
  "value":
   [
    {
     "type": "Application",
     "name": "Trading Application",
     "value":
      [
       {
        "type": "Middleware",
        "name": "WAS",
        "value":
         [
          {
           "type": "Hostname",
           "name": "nc9118041001",
          },
          {
           "type": "Hostname",
           "name": "nc9118041002",
          },
          {
           "type": "Hostname",
           "name": "nc9118041003",
           "value": []
          },
          {
           "type": "Hostname",
           "name": "nc9118041005",
           "value": []
          }
         ]
       },
       {
```

To edit the JSON file:

Procedure

1. Create a backup copy of the `unityServiceTopology.json` file that is located in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/com/ibm/tivoli/loganalytics/framework` directory and copy it to a secure location.
2. Using a JSON editor, open and edit the original file to reflect your service topology.
3. Save the file.
4. To ensure that your updates are reflected on the Admin UI, clear the browser cache before you log in.

Results

After you have defined the service topology JSON file, you can then add it when you configure a data source. If you assign a data source to a node in the service topology, the assignment is maintained regardless of any updates you make to service topology. Changes to the service topology JSON are not reflected in data sources that have been made before the update.

Changing the default timezone

IBM Operations Analytics - Log Analysis uses Coordinated Universal Time as the default timezone. To change the default timezone, complete this procedure.

About this task

You must change this setting after you install IBM Operations Analytics - Log Analysis but before you load any data, including the sample data that is provided on the **Getting Started** page.

Note: You cannot change the timezone after you load data. IBM Operations Analytics - Log Analysis cannot resolve the different time stamps and this conflict causes errors in the search that cannot be resolved. After you change the timezone and load data, do not change the timezone again.

IBM Operations Analytics - Log Analysis supports the Java timezone classification. For a list of supported timezone names, see the "Supported timezone names" on page 210 topic in the *Configuring* reference section

Procedure

1. Install IBM Operations Analytics - Log Analysis. Do not load or install any data.
2. To stop IBM Operations Analytics - Log Analysis, enter the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
   ```
3. To change the default timezone value, edit the `UNITY_TIME_ZONE` parameter in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file. For example, to change the timezone to Western Europe for Paris, France, edit the parameter as follows:
   ```
   UNITY_TIME_ZONE=Europe/Paris
   ```

   Note: You must use the full timezone name rather than the timezone abbreviation in the timezone parameter.
4. Save your changes.
5. To restart IBM Operations Analytics - Log Analysis, enter the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Configuring the timeout for the IBM Operations Analytics - Log Analysis server

You can extend the timeout value for the IBM Operations Analytics - Log Analysis server beyond the default of 120 minutes. For example, if you want to use IBM Operations Analytics - Log Analysis as a monitoring console you can extend the timeout value.

About this task

To change the default value, modify the `server.xml` file that is used by the IBM Operations Analytics - Log Analysis server.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis, enter the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
   ```
2. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/server.xml` file
3. Add the following line to the file. The time is measured in minutes. For example, to set the time out value to 8 hours, add the following line:
   ```
   <ltpa expiration="480"/>
   ```
4. Save the file.
5. To start theIBM Operations Analytics - Log Analysis again, enter the following command:
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
   ```

Results

The timeout value is changed.

Note: The session can time out in less time than is specified in the `server.xml` file. This issue can occur for a number of reasons outside of IBM Operations Analytics - Log Analysis. For example, it can occur if the browser's timeout value is less than the value specified in the `server.xml` file.

Adding or removing IBM Tivoli Monitoring Log File Agent from an existing installation

After you install IBM Operations Analytics - Log Analysis, you can add or remove the IBM Tivoli Monitoring Log File Agent from an existing installation.

Before you begin

IBM Operations Analytics - Log Analysis is installed.

Procedure

1. Navigate to the <*user_home*>/IBM/InstallationManager/eclipse/ directory and run the command:
   ```
   ./launcher
   ```

   Note: If you are accessing the installation environment remotely, ensure that your virtual desktop software is configured to allow you to view the graphical user interface for the IBM Installation Manager.
2. Click **Modify** and add or remove the IBM Tivoli Monitoring Log File Agent.

Managing the Hadoop service on datanodes

Standard

After you configure the Hadoop service, you can use the `server.sh` script to manage the service.

Procedure

You can choose to manage the service on an individual datanode or manage all of the service instances together.

- To manage the service on an individual datanode, use the `LA` user that you created when you configured the service to log in to a Hadoop datanode server.
  1. Run the `<LA_SERVICE_HOME>/bin/server.sh` script with one of the following parameters:

     Start Starts the service on the Hadoop datanode server.

     Stop Stops the service on the Hadoop datanode server.

     Status Retrieves the status for the Hadoop datanode server.
- To manage all of the instances, select one datanode to act as a `LA_Service_Controller_Node`. This will manage the service on all of the datanodes.
  1. (Optional) Create password-less SSH for the `LA` user from this datanode to all of the datanodes, including this datanode, in the Hadoop cluster.
  2. Use the `LA` user to login to the `LA_Service_Controller_Node` datanode.
  3. Run the `<LA_SERVICE_HOME>/bin/server.sh` script with one of the following parameters:

     clusterStart
     Starts the service on each Hadoop datanode server.

     clusterStop
     Stops the service on each Hadoop datanode server.

     clusterStatus
     Retrieves the status for each Hadoop datanode server.

  If you do not configure password-less SSH connections during the configuration, you are prompted for the password for each datanode server.

Sharing a Hadoop cluster across multiple IBM Operations Analytics - Log Analysis instances

Standard

Procedure

1. To share a Hadoop cluster across multiple IBM Operations Analytics - Log Analysis instances, you must integrate the Hadoop service for each IBM Operations Analytics - Log Analysis instance.

   For more information, see the *Integrating the Hadoop service* topic in *Configuration* guide.

   a. You must use a different value for each of the following folders:

   ```
   <LA_HADOOP_TIER> on the HDFS datanode
   <LA_SERVICE_HOME> in the LA home directory
   ```

   Alternatively to repeating the steps for each IBM Operations Analytics - Log Analysis instance, you can create a copy of the resultant folders from a IBM Operations Analytics - Log Analysis instance of the same version.
2. Modify the `PORT` and `PROCESS_ID` values in the *<LA_SERVICE_HOME>*/bin/env.sh file

Administrating data ingestion statistics

You can export your data ingestion statistics. You can also configure non-billable data sources.

Entry If you are using the Entry Edition, you can also configure Log Analysis to send email notification when you reach the daily data ingestion limit. For more information, see "Configuring email notifications for the data ingestion limit" on page 71.

Entry If you are using the Entry Edition, you can also configure Log Analysis to send email notification when you reach the daily data ingestion limit. For more information, see the *Configuring email notifications for the data ingestion limit* topic in the *Administration and Installation* guide.

Server Statistics workspace

Use the Server Statistics workspace to display the rolling 30 day average and the peak average for data ingestion.

To calculate the 30 day rolling average, IBM Operations Analytics - Log Analysis measures the amount of data that is ingested over the previous 30 days, including the current day and divides by 30. Data that is ingested on the day of the upgrade is counted as part of the average.

The workspace displays details in the following fields:

30 day ingestion average
Displays the rolling 30 day average for the current day.

Peak 30 day ingestion average
Displays the highest rolling 30 day average.

Date of peak 30 day ingestion average
Displays the date when the peak day occurred.

The **Daily ingestion and Rolling 30 Day Average** graph displays the daily ingestion total and the rolling 30 day average for the specified date range. The graph is automatically updated when you enter a date.

If you want to refresh the chart without changing the date, click the **Refresh** button.

`export_statistics` command

Use the `export_statistics` command in the <HOME>/IBM/LogAnalysis/utilities directory to export statistics data to a table or comma-separated values (CSV) file line.

Syntax

This command has the syntax:

```
export_statistics <base_uri> [-u -p]| -f | -o | -t | -s | - h
```

where `<base_uri>` is the location of the IBM Operations Analytics - Log Analysis for which you want to export data. If you do not specify a value for the `<base_uri>` parameter, the `$UNITY_BASE_URI` environmental variable is used.

Parameters

These parameters are also available:

-u The user name for a user with administrative privileges. The format for this parameter is `--u=username`. Where `username` is a user with administrative privileges. If you do not specify a value for the -u parameter, the $UNITY_USER environmental variable is used.

-p The password for the user that you specified. The format for this parameter is --p=password. Where password is the password for the user name that you specified. If you do not specify a value for the -p parameter, the $UNITY_PASS environmental variable is used.

-f The format in which you want to export the data. The format for this parameter is --f format. Where format is table or CSV. The default value is Table.

-o Outputs the statistics to a file. The format for this parameter is --o path. Where path is the file name and path for the export file. The default value is stdout.

-t Specifies the type of data that you want to export. The format for this parameter is --t type. Where type is summary, daily, or thirtydayavg. The results are presented in table format with each column having a name. For example, Data Source. The default value is daily.

-s Use this parameter to separate records when outputting to a CSV file. The format for this parameter is --s separator. Where separator is the separator that you want to use in the CSV file. The default value is a comma (,).

-h (Optional) Displays help for this command. The *<base_uri>* parameter is not required for this parameter.

Note: A discrepancy might occur between the file size of the file that is ingested and the value that is returned by the export_statistics command. The value returned is less than or equal to the file size of the ingested file. This difference occurs because incomplete records in the log file are discarded by the Splitter reducing the file size.

The command outputs different information, depending on the type parameter that you specify, summary, daily, or thirtydayavg. The column headings for daily data are:

Data Source
The Data Source with which the ingested data is associated.

Collection
The Collection with which the Data Source containing the data is associated.

Date The date on which the data was ingested.

Ingested Bytes
The volume of data, in bytes, that has been ingested.

Billable
The volume of data that is used for billing purposes.

Log Path
The path to the data source.

Hostname
The host name that indicates the source of the log file.

The column headings for the 30 day, rolling average are as follows:

Current Thirty Day Rolling Avg
The current 30 day, rolling average.

Thirty Day Rolling Avg High-Water Mark
Shows the average peak amount of data that is used over the last 30 days.

Date of Thirty Day Rolling Avg High-Water Mark
Shows the date when the amount of data ingested peaked.

The column headings for the 30 day average are as follows:

Date Shows the date for each average.

Thirty Day Average
Shows the average amount of data that is used over 30 days.

Example

This example downloads the daily data source statistics from `http://unityserver.example.com:9988/Unity` using the user name `unityadmin` and the password `secretpassword`. The statistics are output to a CSV file: `~/Desktop/LogStats.csv`.

```
export_statistics http://unityserver.example.com:9988/Unity
--username=unityadmin --password=secretpassword --f csv --o ~/Desktop/LogStats.csv
```

To export the daily statistics, use the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

The command outputs the following information:

```
Data Source|Collection  |Date      |Ingested Bytes|Billable|Log Path              |
------------+--------------+--------------+------------------+------------+--------
localtest  |  localtest |2013-11-14|22640         |True    |/alogtest/SystemOut.log
localtest  |  localtest |2013-11-13|396200        |True    |/alogtest/SystemOut.log
----------------------------------------------------------------------------------|
| Hostname                 |
-----------+---------------
| <hostname1>.example.com
| <hostname2>..example.com
```

To export a summary, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

This command outputs the following information:

```
Current Thirty Day Rolling Avg| Thirty Day Rolling Avg High-Water Mark|
----------------------------------+-----------------------------------
              13961           | 13961                                 |
----------------------------------------------------------------------
|Date of Thirty Day Rolling Avg High-Water Mark
--------+--------------------------------------
|2013-11-14 00:00:00 -0500
```

To export the 30 day average data, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t thirtydayavg
```

This command outputs the following information:

```
     Date     |  Thirty Day Average
--------------+--------------------
  2013-11-14  |        13961
  2013-11-13  |        13206
```

Configuring non-billable data sources

As part of the interim fix, Netcool Operations Insight customers are entitled to load Netcool Operations Insight and any other Cloud and Smarter Infrastructure product logs into IBM Operations Analytics - Log Analysis at no extra cost. For non-Cloud and Smarter Infrastructure logs, customers are entitled to load up to 2 Gigabyte (GB) average data per day of the 30 day rolling period for free.

About this task

To comply with billable data requirements, you need to segregate Cloud and Smarter Infrastructure and non-Cloud and Smarter Infrastructure data. To segregate this data, you must create separate data sources for billable (non-Cloud and Smarter Infrastructure) and non-billable (Cloud and Smarter Infrastructure) data.

For example, if you use IBM Operations Analytics - Log Analysis to load data from Netcool Operations Insight product logs and also from UNIX Operating System (OS) logs, you must create two separate data sources, one data source for the Netcool Operations Insight logs and one data source for the UNIX OS logs.

Procedure

1. Create a file that is called `seed.txt` in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory.

   Use the first line to specify the free ingestion limit in Megabytes (MBs). The free ingestion limit is up to 2GB average data per day of the 30 day rolling period. Read your license agreement to understand the free data ingestion limit for Cloud and Smarter Infrastructure and non-Cloud and Smarter Infrastructure logs. Use the next lines to specify the paths or directories where your Cloud and Smarter Infrastructure related applications are installed. The log files for most Cloud and Smarter Infrastructure products are stored in the installation directories. These locations must match the log path that is defined when the data source is created. In some Cloud and Smarter Infrastructure integrations, a data source can have an arbitrary string for a log path. In these cases, the strings in the `seed.txt` file must match the string that is specified in the data source.

   The following sample shows a `seed.txt` file. Lines that start with a hashtag (#) are commented out and are ignored.

   ```
   #FREE INGESTION LIMIT (MB)
   2048
   #Data Source locations to be ignored by statistics tracker
   /home/LAuser1/LogAnalysis/logs
   /home/LAuser1/my/log/path
   ```

2. To restart IBM Operations Analytics - Log Analysis, enter the following command:

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

3. Create a data source for the non-billable data.

   If the log path that you specify when you create the data source matches any of the file paths that you specified in the `seed.txt` file, IBM Operations Analytics - Log Analysis does not count the data from these data sources in the billing statistics.

   For more information about how to create a data source, see “Data Source creation” on page 339.

Results

The free data ingestion limit of 2GBs average data for the 30 day rolling period is displayed as a horizontal red line on the Server Statistics UI.

Deleting data

To remove data from IBM Operations Analytics - Log Analysis, use the deletion tool.

About this task

When you delete data with this tool, a log file that is called `DeleteApplication.log` is created and stored in the `<HOME>/logs` directory.

The tool has one limitation. Do not run this utility for use case 1 when data ingestion is in progress for the specified data source. For example, if you are loading data into a data source, do not delete data from the same data source until ingestion is complete.

The tool is configured to run use case 4 by default. If you do not change the default value for the retention period, the default value of 24 hours is used.

For more information about the `delete.properties` parameters and values, see "delete.properties" on page 368.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/utilities/deleteUtility/delete.properties` file.
2. Locate the lines and specify the use case that you want to run:

   ```
   [useCase]
   useCaseNumber = <usecase>
   ```

   where *<usecase>* is one of these use cases.

   useCase_1
   Delete all of the data from a single data source or delete a specific subset of data from the specified data source. If you run this use case, locate this variable in the `delete.properties` file and specify an appropriate value:

   - `dataSourceName`: Specify the name of the data source for which you want to delete data.
   - `query`: Enter a query to delete a specific subset of data. For example, if you enter `severity:E`, all the log records in the data source that contain this severity status are deleted. You must use the Apache Solr query syntax to specify the query.

   useCase_2
   Delete all of the data from a single Collection or delete a specific subset of data for the specified collection. If you run this use case, locate this variable in the `delete.properties` file and specify an appropriate value:

   - `collectionName`: Specify the name of the Collection for which you want to delete data.
   - `query`: Enter a query to delete a specific subset of data. For example, if you enter `severity:E OR severity:W`, all the log records in the

collection that contain either or both of these severity statuses are deleted. You must use the Apache Solr query syntax to specify the query.

useCase_3

Use this use case to delete data from all the IBM Operations Analytics - Log Analysis collections for a time period that you specify. You must specify a start and end time in the same time zone as the time specified in the `unitysetup.properties` file.

Note:

The utility deletes data for whole multiples of the time window value that occur during the specified time only. The time window value is defined in `unitysetup.properties` file. For example, if the time window value is one day and you enter the following start and end times, the utility deletes three complete days worth of data from all IBM Operations Analytics - Log Analysis collections.

```
startTime = <11/21/2013 14:00:00>
endTime = <11/25/2013 09:00:00>
```

The deleted days are 22, 23, and 24 November. Data from 14:00 to 00:00 on 21 November is not deleted nor is data from 00:00 to 09:00 on 25 November.

Note:

If you specify an `endTime` that is in the future, IBM Operations Analytics - Log Analysis displays a message that warns you that this can delete data from the current, active collection. Data is only removed from the current collection if the future date exceeds the value for the collection window. For example, if the collection window value is one day and the `endTime` is specified as `23:00:00` on the same day as the utility is run, data is not deleted from the active collection. However, if the collection window value is one day and the `endTime` is specified as `00:30:00` on the day after the current day, data is deleted from the active collection.

3. Save the `delete.properties` file.
4. Run the tool in stand-alone mode:
   - To run the utility in stand-alone mode, run the command:

     ```
     <Path to Python> deleteUtility.py <password>
     ```

     where *<Path to Python>* is the location to which python is installed and the *<password>* and the associated user name are defined in the `delete.properties` file.
   - To run the utility in cron mode:
     a. Update the `<HOME>/IBM/LogAnalysis/utilities/deleteUtility/callDeleteUtility.sh` file. Ensure that the password specified here is correct.
     b. Enter the following command to run the utility:

        ```
        sh ./createCron.sh
        ```
     c. After the script runs, an entry is created in the cron file. To view the delete cron job that you created, enter the following command:

        ```
        crontab -l
        ```

Deletion tool

To delete data from IBM Operations Analytics - Log Analysis, use the deletion tool.

The tool is available in the `<HOME>/utilites/deleteUtility` directory.

Use the tool to:

- Delete all of the data from a single data source or delete a specific subset of data from the specified data source.
- Delete all of the data from a single Collection or delete a specific subset of data from the specified Collection.
- Delete data from all the IBM Operations Analytics - Log Analysis collections for a time period that you specify.
- Delete data that is older than the specified retention period at regular intervals.

The tool is configured to run use case 4 by default. If you do not change the default value for the retention period, the default value of 24 hours is used.

Administering reference

Read to get reference information about the utilities and tools that you can use when you are administering Log Analysis.

`export_statistics` command

Use the `export_statistics` command in the <HOME>/IBM/LogAnalysis/utilities directory to export statistics data to a table or comma-separated values (CSV) file line.

Syntax

This command has the syntax:

```
export_statistics <base_uri> [-u -p]| -f | -o | -t | -s | - h
```

where `<base_uri>` is the location of the IBM Operations Analytics - Log Analysis for which you want to export data. If you do not specify a value for the `<base_uri>` parameter, the `$UNITY_BASE_URI` environmental variable is used.

Parameters

These parameters are also available:

-u The user name for a user with administrative privileges. The format for this parameter is `--u=username`. Where `username` is a user with administrative privileges. If you do not specify a value for the `-u` parameter, the `$UNITY_USER` environmental variable is used.

-p The password for the user that you specified. The format for this parameter is `--p=password`. Where `password` is the password for the user name that you specified. If you do not specify a value for the `-p` parameter, the `$UNITY_PASS` environmental variable is used.

-f The format in which you want to export the data. The format for this parameter is `--f format`. Where `format` is `table` or `CSV`. The default value is `Table`.

-o Outputs the statistics to a file. The format for this parameter is `--o path`. Where `path` is the file name and path for the export file. The default value is `stdout`.

-t Specifies the type of data that you want to export. The format for this parameter is `--t type`. Where `type` is `summary`, `daily`, or `thirtydayavg`. The results are presented in table format with each column named. For example, `Data Source`. The default value is `daily`.

-s Use this parameter to separate records when outputting to a CSV file. The format for this parameter is `--s separator`. Where `separator` is the separator that you want to use in the CSV file. The default value is a comma (**,**).

-h (Optional) Displays help for this command. The <*base_uri*> parameter is not required for this parameter.

Note: A discrepancy might occur between the file size of the file that is ingested and the value that is returned by the `export_statistics` command. The value returned is less than or equal to the file size of the ingested file. This difference occurs because incomplete records in the log file are discarded by the Splitter reducing the file size.

The command outputs different information, depending on the type parameter that you specify, `summary`, `daily`, or `thirtydayavg`. The column headings for daily data are:

Data Source
The Data Source with which the ingested data is associated.

Collection
The Collection with which the Data Source containing the data is associated.

Date The date on which the data was ingested.

Ingested Bytes
The volume of data that is ingested, in bytes.

Billable
The volume of data that is used for billing purposes.

Log Path
The path to the data source.

Hostname
The host name that indicates the source of the log file.

The column headings for the 30 day rolling average are as follows:

Current Thirty Day Rolling Avg
The current 30 day, rolling average.

Thirty Day Rolling Avg High-Water Mark
Shows the average peak amount of data that is used over the last 30 days.

Date of Thirty Day Rolling Avg High-Water Mark
Shows the date when the amount of data ingested peaked.

The column headings for the 30 day average are as follows:

Date Shows the date for each average.

Thirty Day Average
Shows the average amount of data that is used over 30 days.

Example

This example downloads the daily data source statistics from `http://unityserver.example.com:9988/Unity` using the user name `unityadmin` and the password `secretpassword`. The statistics are output to a CSV file: `~/Desktop/LogStats.csv`.

```
export_statistics http://unityserver.example.com:9988/Unity
--username=unityadmin --password=secretpassword --f csv --o ~/Desktop/LogStats.csv
```

To export the daily statistics, use the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

The command outputs the following information:

```
Data Source|Collection |Date      |Ingested Bytes|Billable|Log Path             |
------------+-------------+--------------+-----------------+------------+--------
localtest  |  localtest|2013-11-14|22640         |True    |/alogtest/SystemOut.log
localtest  |  localtest|2013-11-13|396200        |True    |/alogtest/SystemOut.log
--------------------------------------------------------------------------------|
| Hostname                |
-----------+--------------
| <hostname1>.example.com
| <hostname2>..example.com
```

To export a summary, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

This command outputs the following information:

```
Current Thirty Day Rolling Avg| Thirty Day Rolling Avg High-Water Mark|
----------------------------------+-----------------------------------
              13961           | 13961                                 |
----------------------------------------------------------------------
|Date of Thirty Day Rolling Avg High-Water Mark
--------+--------------------------------------
|2013-11-14 00:00:00 -0500
```

To export the 30 day average data, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t thirtydayavg
```

This command outputs the following information:

```
     Date     |  Thirty Day Average
--------------+--------------------
  2013-11-14  |        13961
  2013-11-13  |        13206
```

delete.properties

To remove data from IBM Operations Analytics - Log Analysis, use the deletion tool.

The `delete.properties` file is in the `<HOME>/IBM/LogAnalysis/utilities/deleteUtility` directory.

The deleteUtility supports three use cases. The use cases require general information, and information specific to each use case.

The hostName, port, userName, and delayDuration parameters are required.

The query parameter is optional and can only be used with use cases 1 and 2.

Table 85. General parameters

Parameter	Value
useCase	The use case number that you want to run.
hostName	The host name that corresponds to the data source defined.
port	The https port. The default port is 9987.
userName	The user name for a user with administrative access rights.
delayDuration	The delay interval (in milliseconds) between two consecutive rest (post) calls.
query	If you want to delete a specific type of data in a specified data source or collection, use this parameter to specify a query in the Apache Solr syntax. For example, entering Severity:E deletes all the log records for the specified severity.

Use Case 1

dataSourceName: Specify the name of the data source for which you want to delete data.

query: Enter a query to delete a specific type of data. For example severity:E

Use Case 2

collectionName: Specify the name of the Collection for which you want to delete data.

query: Enter a query to delete a specific type of data. For example severity:E or severity:W

Use Case 3

```
startTime = <11/21/2013 14:00:00>
endTime = <11/25/2013 09:00:00>
```

API guide

Use this guide to help you to integrate Log Analysis with other applications.

Search REST API overview

The Search REST interface can be used to execute search queries. Search can be executed through a HTTP POST request on https:<host>:<port>;/Unity /Search.

The Search REST interface is the primary search method that is used to execute search queries.

Searching over data sources

The input search request and the response back from USR are both modeled as JSON objects. In the following section, the structure of these input and output JSON objects are described.

JSON Format for Search Request

The input JSON for a search request is a single JSON object.

The input JSON object has the structure:

```
{
      "logSources": ["logsource1", "logsource2", ....],
      "query":  "the search query string entered by the user",
      "filter":  //filter query (see section 4.1 below)
      "queryLang": "the language of the search query",
      "start": 0,
      "results": 10,
      "getAttributes": ["attribute1", "attribute2", .....],
      "sortKey": ["key1", "key2", ......],
      "facets": {
            "facet1D1":  // facet request (see section 4.2 below)
            "facetID2":  // facet request (see section 4.2 below)
                  ......
      }
      "advanced": {
            "rankingHint": "a query expression that provides a ranking hint
             for Gumshoe",
            "explain": false,
"interpretations": true,
"catchallOnly": false,
"rules": {
                  "allRules": true,
                  "categoryRules": true,
                  "rewriteRules": true,
                  "ignoreCategories": ["cat1", "cat2", ....]
 }
            "grouping":  ["groupingType1", "groupingType2", ..],
            "highlighting": true,
      }
}
```

1,000 or more results and Custom Apps: When a query in a Custom App returns more than 1000 records, you get only 1000 results back. The search result returned includes a field `totalResults` which shows total number of matching results. Another field `numResults` gives the number of records returned. You can check these values in the Custom App script and handle the results accordingly.

The following table lists the semantics of the remaining attributes

Table 86. Search request input parameters

Name	Description	Default Value	Comments
logsources	logsources against which the search request must be performed	Required	A list of logsources. This should be a JSON array. Each entry can be a logsource name or tag name. If a tag name is specified , all logsources under the tag will be included in the search `"logsources": [ {"type":"tag","name":"/ipo"}, {"type":"logSource", "name":"/DayTraderLogSource"} ]`
query	Search query	Required	Typically, the value of this parameter is whatever is entered by the end-user in a search box. However, any valid query string as per the Velocity query syntax (excluding range queries) is permitted.

Table 86. Search request input parameters (continued)

Name	Description	Default Value	Comments
filter	Application filter	No filter	Valid JSON object as described in section 4.1. This parameter is intended for applications to pass in filters in addition to the user search query. Conceptually, the overall query processed by USR is "query AND filter". The separation into two parts is to allow for additional query manipulation of the "query" part when we implement advanced search capabilities.
start	Offset of the first result to return (Integer)	0	If specified value is negative, the value will be defaulted to 0; if the specified value is greater than the number of results, no results will be returned.
results	Number of results desired (Integer)	10	Min value is 1 (values <= 0 default to 1); maximum value is 1000.
getAttributes	Attributes to be returned for each result entry	Not required	When this parameter is not specified in the request, the engine will return only the set of attributes marked as `retrievebyDefault` in the indexing configurations associated with the logsources in question. If this parameter is specified and is a non-empty array, then the attributes listed in the array are fetched. Finally, if the parameter is specified but is an empty array, then ALL retrievable attributes across all logsources will be returned for each result entry.
sortKey	One or more fields on which to sort the result	Relevance order	A valid value for this parameter is a comma-separated list of field names, with each field name prefixed by "+" or "-". Each field name appearing in the list must have been declared to be a "sortable" field at index build time. The first field in the list is treated as the primary sort key, the second field (if present) as the secondary sort key, and so on. The "+" (resp. "-") prefix is an instruction to sort the corresponding field values in ascending (resp. descending) order. For queries involving multiple logsources, sort keys must exist in all logsources involved in the query, and sort keys must have the same type across logsources.

Table 86. Search request input parameters (continued)

Name	Description	Default Value	Comments
outputTimeZone	Time zone in which DATE field results should be formatted	Collection time zone (for single logsource queries and multi-logsource queries where logsources have identical time zones). Server time zone (for multi-logsource queries where logsources have different time zones)	
outputDateFormat	SimpleDateFormat string that specifies how DATE field results should be formatted	UNITY_DATE_ DISPLAY_FORMAT Read from unitysetup.properties	

Filter query

A filter query is a Boolean query specified as a nested JSON record. In its simplest form a Boolean query consists of a basic query. A basic query can be a term query, wildcard query, phrase query or range query. Basic queries can also be combined using arbitrarily nested conjunctions (*AND* queries), disjunctions (*OR* queries) and negations (*NOT* queries) to form complex Boolean queries.

4.1.1 Basic filter queries

Term query

A term query is specifies a field name and a term. It matches all documents for which the field contains the term. A term query is specified as follows:

```
{ "term":
{"myField": "termValue"}
}
```

Wildcard query

A wildcard query specifies a field name and a wildcard expression. It matches all documents for which the field matches the wildcard expression. A wildcard query is specified as follows:

```
{ "wildcard":
{"myField": "wildcardExpression"}
}
```

Phrase query

A phrase query specifies a field name and a phrase. It matches all documents for which the field contains the phrase. A phrase query is specified as follows:

```
{ "phrase":
{"myField": "phraseValue"}
}
```

Range query

A range query specifies a field name along with a lower bound (inclusive) and an upper bound (exclusive) for the field value. A range query is applicable only to numeric and date fields. For date fields, an additional date format must be provided. A range query is specified as follows:

```
{"range":
    {"myField":
      { "from": "lower-bound", // value will be included in the search
          "to": "upper-bound", // value will be excluded in the search
          "dateFormat": "date-format" // only for date fields
       }
    }
}
```

4.1.2 Complex filter queries

Complex filter queries can be constructed by combining one or more queries using AND, OR or NOT queries.

AND query

An AND query consists of two or more sub-queries. Sub-queries can be either a basic query or another complex query. A document satisfies an AND query only if it satisfies all of its sub-queries. An AND query is specified as follows:

```
{"and":[
        {"query1": ...},
        {"query2": ...},
        ...
        {"queryN": ...}]
            }
```

OR query

An OR query consists of two or more sub-queries. Sub-queries can be either a basic query or another complex query. A document satisfies an OR query if it satisfies at least one of its sub-queries. An OR query is specified as follows:

```
{"or":[
        {"query1": ...},
        {"query2": ...},
        ...
        {"queryN": ...}]
}
```

NOT query

A NOT query consists of a single sub-query. The sub-query can be either a basic query or a complex query. A document satisfies a NOT query if it does not satisfy the contained sub-query. A NOT query is specified as follows:

```
{"not": {"query": ....}}
```

Facet requests

Different types of facet requests are supported by USR, along with the JSON format used to specify each type of facet request.

Each facet request is specified as a JSON key-value pair with the key being the facetID and the value being a JSON record. The type of facet being computed determines the structure of this JSON record. The supported facet types and their corresponding JSON request format are described here.

Term Facets

```
"myTermFacet": {
      "terms": {
            "field": "myField",
            "size": N
       }
}
```

Facet counting is performed on the field *myField* and the top-N most frequent facet values (for some positive integer N) is returned.

The next two facets (histogram and statistical) apply only to numeric fields. In other words, the field on which these facets are being computed must have been configured with a dataType=LONG or dataType=DOUBLE in the indexing specification associated with the IBM Operations Analytics collection(s) over which the facet request is being processed.

Histogram Facets

```
"myHistoFacet": {
      "histogram": {
            "field": "myField",
            "interval": 100
      }
}
```

Performs facet counting with buckets determined based on the *interval* value.

Statistical Facets

```
"myStatFacet": {
      "statistical": {
            "field": "myField",
            "stats": ["min", "max", "sum", "avg", "count", "missing",
       "sumOfSquares", "stddev"]
      }
}
```

Performs simple aggregate statistics on a facet field. Eight statistics are supported - maximum, minimum, summation, average, count, missing, sum of the squares, and standard deveiation. The "stats" attribute specifies which of these statistics should be computed for the given facet request.

Date Histogram Facets

```
"myDateHistoFacet": {
 "date_histogram": {
  "field": "myField",
  "interval": "day",
  "outputDateFormat": "yyyy-MM-dd
   'T' HH:mm:ssZ"
 }
}
```

A version of the histogram facet specialized for date fields. The value of the *interval* attribute can be one of the string constants *year*, *month*, *week*, *day*, *hour*, or *minute*. The value of the *outputDateFormat* is any valid date format string as per the Java SimpleDateFormat class. This format string is used to represent the histogram boundaries in the response JSON coming out of USR.

For single collection data histogram facets, boundaries are based on the collection time zone (either from the index configuration, or from unitysetup.properties if missing in the index configuration). For multi-collection facets, boundaries are based on collection time zone if the time zone of all collections is identical. For multi-collection facets where collection time zones differ, boundaries are based on the server time zone.

Note: The results returned from the date histogram facet are not sorted. If you are plotting the resulting time intervals in a chart, you need to sort the JSON returned by the date histogram facet. For example, in python, if your search request is the following:

```
request = {
        "start": 0,
        "results": 1,
        "filter": {
              "range": {
                        "timestamp":{
                          "from":"01/01/2013 00:00:00.000 EST",
                          "to":"01/01/2014 00:00:00.000 EST",
                          "dateFormat":"MM/dd/yyyy HH:mm:ss.SSS Z"
                        }
                }
       },
        "logsources": [{"type": "logSource", "name": "MyTest" }],
        "query": "*",
        "sortKey":["-timestamp"],
        "getAttributes":["timestamp","perfMsgId"],
        "facets":{
            "dateFacet":{
                "date_histogram":{
                    "field":"timestamp",
                    "interval":"hour",
                    "outputDateFormat":"MM-dd HH:mm",
                    "nested_facet":{
                        "dlFacet":{
                            "terms":{
                                "field":"perfMsgId",
                                "size":20
                            }
                        }
                    }
                }
            }
        }
    }
```

First, retrieve the `dateFacet` from the JSON returned by the http request and then call the `dateSort()` function .

```
    response = connection.post(
      '/Search', json.dumps(request),
      content_type='application/json; charset=UTF-8');
    content = get_response_content(response)

    #convert the response data to JSON
    data = json.loads(content)

    if 'facetResults' in data:

         # get the facet results
         facetResults = data['facetResults']

         if 'dateFacet' in facetResults:
             # get the dateFacet rows
             dateFacet = facetResults['dateFacet']

             # the results of the dateFacet are not sorted,
             # so call dateSort()
             dateSort(dateFacet)
```

where dateSort() is defined as follows:

```
#------------------------------------------------------
# dateSort()
#------------------------------------------------------
def dateSort(dateFacet):
    # This function parses the UTC label found in the dateFacet in
    # the format "mm-hh-DDD-yyyy UTC"
    # and returns an array in the form [yyyy, DD, hh, mm]
    def parseDate(dateLabel):
        aDate = map(int, dateLabel.split(" ")[0].split("-"))
        aDate.reverse()
        return aDate

    # call an in-place List sort, using an anonymous function
    # lambda as the sort function
    dateFacet.sort(
      lambda facet1, facet2: cmp(parseDate(facet1['label']),
      parseDate(facet2['label'])))
    return dateFacet
```

Nested Facets

A facet request can be nested inside another facet request, by specifying a nested_facet key. You can nest facets to any number of levels.

The following is a valid nested facet query, with a termsfacet query nested inside a date_histogram facet query:

```
"facets":
  {"dateFacet":{
     "date_histogram":{
         "field":  "timestamp","interval":"hour",
         "outputDateFormat":"MM-dd HH:mm",
         "nested_facet":{
          "severityFacet":{
             "terms":{
                "field":"severity",
             "size":10
                    }
               }
            }
       }
    }
  },
```

JSON Format for Search Response

The search results from USR are also packaged as a single JSON object .

The JSON object has the structure:

```
{
   "searchRequest": // copy of the entire input JSON object that
  generated this response
   "totalResults": // integer value representing total number of
  results for the query
   "numResults":  // number of top-level results sent back within
  this JSON ("top-level"
             // because a grouped/clustered result is counted as 1
   "executionInfo": {
      "processingTime":
       // time (in ms) measured from the receipt of the search
             // request by USR to point when USR begins to construct the result
            "interpretations":  // for advanced search post
            "rules": // for advanced search post
      }
   "searchResults": [
               // Array of JSON objects one per top-level result entry.
               // The size of this array will be the value of the "numResults"
        attribute
                  ]
   "facetResults": {
         "facetID1":  { // Object with facet information for facetID1 }
         "facetID2": {  // Object with facet information for facetID2 }
                        ......
      }
}
```

Each element of the "searchResults" array will have the following structure:

```
{
"resultIndex":  // a number that denotes the position of this result in the
                       // overall result set for this query
"attributes": {
            "field1": "value1",
            "field2": "value2",
             ....
              // one key-value pair for each field of the result entry;
        the set of fields
              // will be determined by the semantics of the getAttributes
       parameter
```

The JSON structure for the facet results depends on the specific type of facet request.

Term Facets

```
        "facetID": {
              "total": // total number of distinct terms in the field
                  used for generating this facet
              "counts": [
                          { "term": "term1",  "count":  10},
                          { "term": "term2",  "count": 5},
                          ...
                    ]
        }
```

Histogram Facets

```
        "facetID": [
                    { "low": 50, "high": 150, "count": 10},
                    { "low": 150, "high": 250, "count": 25},
                     ...
                   ]
```

Statistical Facets

```
        "facetID": {
              "max": // max value
              "min": //min value
              "sum": // sum value
             "avg": // avg value
             "count": // count value
             "missing": // missing values
             "sumOfSquares": // sumOfSquares value
             "stddev": // stddev value
        }
```

In general, all three aggregates do not have to be present. Only the aggregates listed in the "stats" attribute of the corresponding facet request will be included.

Date histogram Facets

Identical to the output of the histogram facets, except that the "low" and "high" attributes will be represented according to the format string specified in the input date histogram facet request. For example, the output may be something that looks like the following:

```
"facetID": [
            { "low": "2012-01-01 10:00:00", "high":
        "2012-01-01 11:00:00,"count": 10},
            { "low": "2012-01-01 11:00:00", "high":
       "2012-01-02 12:00:00","count": 10
       "label": "9-22-188-2012 UTC" },
...
      ]
```

Nested Facets

If the outermost facet is a term facet, the response will be as follows:

```
     "total": // total number of distinct terms in the field used for
      generating this facet
    "counts": [
     { "term": "term1",  "count":  10, "nested_facet":
    {nested facet result...}},
     { "term": "term2",  "count": 5, "nested_facet":
    { nested facet result...}}]
                 ...
            ]
```

Search query API

The /query API works like the /Search API, with the exception of the structure of the response JSON. This API returns the data in tabular format instead of hierarchical format.

Search Request

Search request structure is the same as the /Search API.

The only extra field is name, which is an optional field. The name is used as the ID of the data set in the results. If the name is not specified,

```
searchResults
```

is used as the ID.

```
{
"name":"AllErrors",
"start": 0,
"results": 10,
"filter": { },
"logsources": [ ... ],
"query": "*",
"getAttributes": [ ... ],
"facets": { "facetId1" :{...}, ...}
}
```

Other search parameters, such as **search filters**, **facets**, **nested facets**, **logsources**, **query**, **getAttributes**, **start**, and **results** are described in section 4.

Search Results

The search results are in a tabular format, which can be ingested by custom applications. The key 'data' in results points to an array of data sets. Each data set has ID, fields, and rows.

Results include one data set for search results and one for each facet that is specified in the request.

Search results data set uses the 'name' specified in the request as the ID. If not specified **searchResults** is used as ID.

Facet results use the facet ID used in the request as the ID for the data set. In case of term, histogram and date-histogram facets, 'count' is added to the fields along with the specified fields.

For statistical facets (max, min, sum, avg, count, missing, sumOfSquares, and stddev), field ID is generated by combining field name and the function name. For example, for 'min' it is `fieldname-min` where `fieldname` is the field included in the statistical facet. Similarly, for max it is `fieldname-max`.

```
{
"data": [
{
"id": "AllErrors",
"fields": [
{"label": "fieldLabel1", "type": "TEXT", "id": "fieldId1" },
{ "label": "fieldLabe2", "type": "LONG", "id": "fieldId2"}
],
"rows": [
{
"fieldId1": "value1",
"fieldId2": "value2"
} ]
},
{
"id": "facetId1",
"rows": [
{
"fieldId1": "value1",
"fieldId2": "value2",
"count": "value3"
},
{
"fieldId1": "value1",
"fieldId2": "value2",
"count": "value3"
}
],
"fields": [
{ "label": "fieldLabel1", "type": "LONG", "id": "fieldId1" },
{ "label": "fieldLabel2", "type": "LONG", "id": "fieldId2"},
{ "label": "Count", "type": "LONG", "id": "count" } ]
}
]
}
```

Search request and results

This example shows a search request and response with sample data.

Search request

```
{
"start": 0,
"results": 100,
"name":"AllErrors",
"logsources": [
{
"type": "tag",
"name": "*"
}
],
"query": "*",
"facets": {
```

```
"termFacet01": {
"terms": {
"field": "msgclassifier",
"size": 419
     }
   }
  }
}
```

Search results

```
{
"data": [
{
"fields": [
{
"label": "msgclassifier",
"type": "TEXT",
"id": "msgclassifier"
},
{
"label": "className",
"type": "TEXT",
"id": "className"
},
{
"label": "logsource",
"type": "TEXT",
"id": "logsource"
}
],
"rows": [
{
"msgclassifier": "SRVE0250I",
"className": "com.ibm.ws.wswebcontainer.VirtualHost",
"logsource": "WASLogSource"
},
{
"msgclassifier": "SECJ0136I",
"className": "com.ibm.ws.wswebcontainer.VirtualHost",
"logsource": "WASLogSource"
}
],
"id": "AllErrors"
},
{
"rows": [
{
"msgclassifier": "SRVE0242I",
"count": 132
},
{
"msgclassifier": "CWPKI0003I",
"count": 3
}
],
"fields": [
{
"label": "msgclassifier",
"type": "TEXT",
"id": "msgclassifier"
},
{
"label": "count",
"type": "LONG",
"id": "count"
}
],
"id": "termFacet01"
  }
 ]
}
```

Using the REST API to administer the Log Analysis data model

You can use HTTP methods and the REST API to load, create, update, and delete Log Analysis artifacts and batches of artifacts.

To load, create, update, and delete a batch of artifacts such as rule sets, file sets, source types, collections, and log sources, use a `GET`, POST, PUT, or DELETE method and the following URL:

```
https://<server>:<port>/Unity/<Artifact>
```

where *<server>* is the machine that IBM Operations Analytics - Log Analysis is installed on. *<port>* is the port that you want to use for REST API requests on the same machine. *<Artifact>* is the name of the artifact that you want to process, for example rule sets.

To load, create, update and delete a specific artifact, use a `GET`, POST, PUT, or DELETE method and the following URL:

```
https://<server>:<port>/Unity/<Artifact>?id=<ruleset_id>
```

Read the following documentation to get more information about specific input parameters and returned values.

Rule sets

You can use various HTTP methods to load, create, update, and delete rule sets.

Load (GET method)

To return all the rule sets, use a `GET` method and the following URL:

```
https://<server>:<port>/Unity/RuleSets
```

To return a specific rule set, use a `GET` method and the following URL:

```
https://<server>:<port>/Unity/RuleSets?id=<ruleset_id>
```

where *<ruleset_id>* is the ID of the rule set that you want to retrieve.

The operation returns the following values from the requested rule sets in the simple JSON format:

Table 87. Returned value for rule sets

Returned value	Description
`rulesFileDirectoy`	Full path to the file that contains the rules that govern how the splitting and annotating is done. The rules must be written in the Annotated Query Language (AQL) format.

Table 87. Returned value for rule sets (continued)

Returned value	Description
type	This parameter can have a value of 1 or 0. 0 means that the rule set is used for splitting. 1 means that the rule set is used for annotation.
name	The name of the rule set.
id	The rule set identifier.

The operation returns the following value for a single rule set:

```
{
    "rulesFileDirectory": "/home/....;",       (- AQL Path)
    "type": 1,                                 (- 0 for Split 1 for Annotate)
    "name": "windowsOSEventsLS-Annotate",
    "id": 3
}
```

Create (POST method)

To create a rule set, use a POST method and the following URL:

```
https://<server>:<port>/Unity/RuleSets?
```

To specify the values for the rule set, define them in the HTTP message body in the JSON format. The input values are listed in the table.

Table 88. Input values for POST method

Input parameter	Description
rulesFileDirectoy	Enter the AQL path.
type	This parameter can have a value of 1 or 0. 0 means that the rule set is used for splitting. 1 means that the rule set is used for annotation.
name	The name of the rule set.

The specification for the new rule set is defined in the input JSON. For example:

```
{
    "name": "Test",
    "type": 0,                          (- 0 for Split 1 for Annotate)
    "rulesFileDirectory": "/home/....;"  (- AQL Path)
}
```

Update (PUT method)

To update a rule set, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/RuleSets
```

To specify the values for the rule set, define them in the HTTP message body in the JSON format. The input values are the same as those values listed in table 2.

The input JSON is the same as that which is used for the POST method.

Delete (DELETE method)

To delete a rule set, use a DELETE method and the following URL:

```
https://<server>:<port>/Unity/RuleSets
```

File sets

You can use various HTTP methods to load, create, update, and delete file sets.

Return file sets (GET method)

To return all the file sets, use a GET method and the following URL:

```
https://<server>:<port>/Unity/FileSets
```

To return a specific file set, use a GET method and the following request:

```
https://<server>:<port>/Unity/FileSets?id=<fileset_id>
```

where *<fileset_id>* is the ID of the file set that you want to retrieve.

The operation returns the following values from the requested file sets in the simple JSON format:

Table 89. Returned values for file sets

Returned value	Description
className	The name of the Java class that is used by the file set.
fileType	This parameter value can be either 0 or 1. 0 means that the file set is Java-based. 1 indicates that the file set is Python-based.
fileName	The name of the file that contains the code that does the splitting or annotating.
type	This parameter value can be either 0 or 1. 0 means that the file is used to split log files. 1 indicates that the file set is used to annotate log files.
name	The name of the file set.
id	The file set identifier.

For example, the operation returns the following values in the JSON format for a file set:

```
{
    "className": "com.ibm.tivoli..annotator.JavacoreAnnotator",
    "fileType": 0, (- 0 for Java 1 for Script)
    "fileName": "JavacoreExtractor_v1.1.0.1.jar",
    "type": 1, (- 0 for Split 1 for Annotate)
    "name": "Javacore-Annotate",
    "id": 2
}
```

Create (POST method)

To create a file set, use a POST method and the following URL:

```
https://<server>:<port>/Unity/FileSets
```

You define the parameter values in the JSON format in the HTTP message body. For example:

```
{
    "name": "Test",
    "type": 0, (- 0 for Split 1 for Annotate)
    "fileType": 0, (- 0 Java 1 for Script)
    "fileName": "db2-content.jar", (- Jar file name
    "className": "TestClass" (- Java Class name)
}
```

The input values are the same as the ones that are specified in table 1 except for id. This value is generated when the file set is created.

Update (PUT method)

To update a file set, use a PUT method and the following URL:

`https://<server>:<port>/Unity/FileSets`

You define the parameter values in the JSON format in the HTTP message body. The input values are the same as the ones that are specified in table 1 except for id. This value is generated when the file set is created.

Delete (DELETE method)

To delete a file set, use a DELETE method and the following URL:

`https://<server>:<port>/Unity/FileSets?id=<fileset_id>`

where <*fileset_id*> is the ID of the file set that you want to delete.

Source types

You can use various HTTP methods to load, create, update, and delete source types.

Load (GET method)

To load a source type, use a GET method and the following URL:

`https://<server>:<port>/Unity/SourceTypes`

To load a single source type, use a GET method and the following URL:

`https://<server>:<port>/Unity/SourceTypes?id=<sourcetype_id>`

where <*sourcetype_id*> is the identifier of the source type that you want to load.

The possible input values are described in the table:

Table 90. GET method parameters

Parameter	Description
indexingConfig	Enter the valid index configuration for the JSON file.
inputType	Specify the type of file that is loaded. 0, denoting log file, is the only possible value.
splitter:fileSet	Specify the file set that the splitter uses to split log files.
splitter:ruleSet	Specify the rule set that the splitter uses to split log files.

Table 90. GET method parameters (continued)

Parameter	Description
splitter:type	Specify the type of splitter that is used. 0 means that the file is used for splitting log files. 1 means that the file is used for annotating log files.
name	Specify the name of the source type that you want to create.
id	Specify the identifier of the source type that you want to load.
annotator:fileSet	Specify the file set that the annotator uses to annotate log files.
annotator:ruleSet	Specify the rule set that the annotator uses to annotate log files.
annotator:type	Specify the type of annotator that is used. 0 means that the file is used for splitting log files. 1 means that the file is used for annotating log files.
annotator:postOnFailure	Specify whether you want to annotator to post results if the annotation process fails. The default value is false.

You define the parameter values in the JSON format in the HTTP message body. For example:

```
{
    "indexingConfig": {},    - Valid index configuration JSON
    "inputType": 0,      - 0 for log file
    "splitter": {
        "fileSet": <fileset_id>,
        "ruleSet": <ruleset_id>,
        "type": 1      - 0 for Split 1 for Annotate
    },
    "name": "Javacore",
    "id": <id>,
    "annotator": {
        "fileSet": <fileset_id>,
        "ruleSet": <ruleset_id>,
        "type": 1,      - 0 for Split 1 for Annotate
        "postOnFailure": false
    }}
```

Create (POST method)

To create a source type, use a POST method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes
```

You define the parameter values in the JSON format in the HTTP message body. The values are the same as the ones that are listed in table 1. For example:

```
{
    "name": "Test",
    "indexingConfig": {},    - Valid index configuration JSON
    "inputType": 0,      - 0 for log file
    "splitter": {
        "ruleSet": <ruleset_id>,
    "fileSet": <fileset_id>
    },
    "annotator": {
```

```
"ruleSet": <ruleset_id>,
    "fileSet": <fileset_id>,
    "postOnFailure": true
}}
```

Update (PUT method)

To update a source type, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes?id=<sourcetype_id>
```

where *<sourcetype_id>* is the identifier of the source type that you want to update.

You define the parameter values in the JSON format in the HTTP message body. The values are the same as the ones that are listed in table 1. The input JSON is the same as that described for POST method.

Delete (DELETE method)

To delete a source type, uses a DELETE method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes?id=<sourcetype_id>
```

where *<sourcetype_id>* is the identifier of the source type that you want to delete.

Collections

You can use various HTTP methods to load, create, update, and delete collections.

Load (GET methods)

To load a single collection, use a GET method and the following URL:

```
https://<server>:<port>/Unity/Collections
```

To return a specific collection, use a GET method and the following URL:

```
https://<server>:<port>/Unity/Collections?id=<collection_id>
```

where *<collection_id>* is the ID of the collection that you want to retrieve.

The method returns the information in the JSON format. The returned values are listed in the table.

Table 91. Parameters for GET method

Parameter	Description
indexingConfig	The valid index configuration value for the JSON file.
sourceType	The source type ID of the returned collection.
name	The name of the returned collection.
id	The identifier of the returned collection.
annotator	This parameter value can be either 0 or 1. 0 means that the source type is used to split log files. 1 indicates that the source type is used to annotate log files.

For example:

```
{
    "indexingConfig": null,
    "sourceType": 15,
    "name": "Javacore-Collection1",
    "id": 1,
    "annotator": 0
}
```

Create (POST method)

To create a collection, use the `POST` method and the following URL:

`https://<server>:<port>/Unity/Collections`

You define the parameter values in the JSON format in the HTTP message body. The parameter values are:

Name Specify the name for the collection.

sourceType
Specify the number that represents the new source type.

For example:

```
{
    "name": "Test",
    "sourceType": 6
}
```

Update (PUT method)

To update a collection, use a `PUT` method and the following URL:

`https://<server>:<port>/Unity/Collections?id=<collection_id>`

where *<collection_id>* is the identifier of the collection that you want to update.

You define the updated parameter values in the JSON format in the HTTP message body. The parameter values and input JSON are the same as those that are used by the `POST` method.

Delete (DELETE method)

To delete a collection, use a `DELETE` method and the following URL:

`https://<server>:<port>/Unity/Collections?id=<collection_id>`

where *<collection_id>* is the identifier of the collection that you want to delete.

Log sources

You can use various HTTP methods to load, create, update, and delete data sources.

Data sources are called log sources in the REST API files.

Load (GET method)

To load all the data sources and associated tags, use a `GET` method and the following URL:

`https://<server>:<port>/Unity/Logsources/ChildTags`

The data sources are returned in a JSON array. The returned values for each data source are listed in the table.

Table 92. Returned values for `GET` *method*

Parameter	Description
`type`	Identifies whether the entry is a tag or a log source.
`name`	The name of the tag or log source.
`id`	The ID of the tag. This only applies to tags.
`logsource_pk`	The ID of the log source. This only applies to log sources.

For example:

```
[
    {
        "type": "tag",
        "name": "day trader",
        "id": <tag id>
    },
    {
        "type": "logSource",
        "logsource_pk": <logsource id>,
        "name": "WASLS04"
    }
    .....
]
```

To list all the data sources and tags that are defined beneath a specified tag, use the following URL:

`https://<server>:<port>/Unity/Logsources/ChildTags?tag_id=<tag_id>`

where *<tag_id>* is the ID of the tag that you want to specify.

This method returns an array in the JSON format that is the same as that returned for the previous method.

To load the data sources and tags for a specific data source name, use the following URL:

`https://<server>:<port>/Unity/Logsources?name=<logsource_name>`

where *<logsource_name>* is the name of the log source that you want to load.

You can also use the log source ID to specify the data source:

`https://<server>:<port>/Unity/Logsources?logsource_pk=<logsource_id>`

where *<logsource_id>* is the identifier of the log source that you want to load.

Both of these methods return the data source and tag information as a JSON file. The returned values are listed in the table.

Table 93. Returned values for `GET` *method (single log source)*

Parameter	Description
`log_path`	The full path and log file name.

Table 93. Returned values for GET method (single log source) (continued)

Parameter	Description
tag_id	The identifier for any tags that are associated with the data source.
userid	The identifier for the user who created the data source.
time_format	The time format used by the data source.
name	The name of the data source.
state	The status of the data source.
hostname	The host name that is specified when the data source is created. This value is either local, that is the same as the machine where IBM Operations Analytics - Log Analysis is installed or it is remote, that is different to the machine where IBM Operations Analytics - Log Analysis is installed.
rolling_pattern	Identifies whether the data source uses rolling patterns for log files. NULL signifies that rolling pattern log files are not used.
encrypted_password	The encrypted password for the specified user ID.
config_type	The type of data source. This can be local, remote or custom.
collectionId	The identifier of the associated collection.
service_topology	The associated service topology, if any.
logsource_pk	The log source ID.
description	The description entered during log source creation.

For example:

```
{
    "log_path": "/home/...../SystemOut.log",
    "tag_id": <tag id>,
    "userid": <user id>,
    "time_format": "",
    "name": "WASLS01",
    "state": 0,
    "hostname": "nc9118041071",
    "rolling_pattern": null,
    "encrypted_password": null,
    "config_type": "custom",
    "collectionId": <collection id>,
    "service_topology": "",
    "logsource_pk": <logsource id>,
    "description": ""
}
```

Create (POST method)

To create a new data source, use a POST method and the following URL:

```
https://<server>:<port>/Unity/DataSourceConfig
```

You define the parameter values in the JSON format in the HTTP message body. The input parameters are listed in the table.

Table 94. Input parameters for POST method

Parameter	Description
config_type	The type of data source. This can be local, remote, or custom.
hostname	The host name of the machine where the data that the data source collects resides.
userid	The user name that is associated with the data source. This parameter is optional.
encrypted_password	The password that is associated with the user name. This parameter is optional.
datasourcepath	The full path and log file name of the log file that is loaded by this data source.
sourceType	The identifier of the source type that is associated with the data source.
rollingfile	If you want to use the data source to collect rolling log files, set this parameter to True. If you do not, set it to False.
filepattern	The file pattern that is associated with the data source. This parameter is optional.
name	The name of the data source. This parameter is optional.
description	The description for the data source. This parameter is optional.
group	The name of the group that is associated with the data source. This parameter is optional.

For example:

```
{
    "configType": "local",
    "hostname": "nc9118041071.in.ibm.com",
    "username": null,
    "password": null,
    "datasourcepath": "/home/...../SystemOut2.log",
    "sourceType": <sourcetype id>,
    "rollingfile": false,
    "filepattern": null,
    "name": "Test",
    "description": "",
    "group": ""
}
```

Update (PUT method)

To update a data source, use a PUT method and the following URL:

`https://<server>:<port>/Unity/Logsources?logsource_pk=<logsource_id>`

where *<logsource_id>* is the log source that you want to update.

You define the parameter values in the JSON format in the HTTP message body. The input values are the same as the ones that are specified in table 3.

Delete (`DELETE` method)

To update a data source, use a `DELETE` method and the following URL:

```
https://<server>:<port>/Unity/Logsources?logsource_pk=<logsource_id>
```

where *<logsource_id>* is the identifier of the log source that you want to delete.

Alerting REST API

Standard

Use the Alerting REST API to create and manage templates, base conditions, composite conditions, and alert actions.

Alerting REST API for Templates

Standard

Use the Alerting REST API to create, get, and delete base condition, composite condition, and alert action templates.

You cannot use the Alerting REST API to update templates.

To complete an action, enter the following URL and add the action:

```
https://<server>:<port>/Unity/Alerts/1.0/<Relative-URL>
```

To create a template, use a POST method and the URL to specify the required attributes.

Create a template

To create a base condition template, use the `BaseConditionTemplate` relative URL.

To create a composite condition template, use the `CompositeConditionTemplate` relative URL.

To create an alert action template, use the `AlertActionTemplate` relative URL.

To specify the required template attributes, use a POST method and the URL with JSON in the following format in th POST body:

```
{
"name": "<template_name>",
"description": "<template_description>",
"implLanguage": "JAVA",
"className": "alert.def.AlertTemplate1",
"implArtifact": base-64-encoded-jar-contents",
"parameters": { "<parameter_1>": {"description": "<parameter_1_description>",
"optional": false, "multivalued": false},
"<parameter_2>": {"description": "<parameter_2_description>",
"optional": true, "multivalued": true}
"type": "<parameter_2_type>"
}
}
```

Multivalued parameters must be JSON arrays. If the template does not contain any parameters, you cannot create more than one instance of the template.

Return details for a template

Use the HTTP GET method for the following actions

- To return the details for a base condition template, use the `BaseConditionTemplate/<template_name>` relative URL.
- To return the details for a composite condition template, use the `CompositeConditionTemplate/<template_name>` relative URL.
- To return the details for an alert action template, use the `AlertActionTemplate/<template_name>` relative URL.
- *<template_name>* is the name of the template that you want to return information on.

Return details for all templates

Use the HTTP GET method for the following actions.

- To return the details for all base condition templates, use the `BaseConditionTemplate` relative URL.
- To return the details for all condition templates, use the `CompositeConditionTemplate` relative URL.
- To return the details for all alert action templates, use the `AlertActionTemplate` relative URL.

Delete a template

Use the HTTP DELETE method for the following actions.

- To delete a base condition template, use the `BaseConditionTemplate/<template_name>` parameter.
- To delete a condition template, use the `DeleteCompositeConditionTemplate/<template_name>` parameter.
- To delete an alert action template, use the `DeleteAlertActionTemplate/<template_name>` parameter.
- *<template_name>* is the name of the template that you want to return information on.

Alerting REST API for base conditions

Standard

Use the Alerting REST API to create, get, update, and delete baseline conditions.

To complete an action, enter the following URL and add the action:

```
https://<server>:<port>/Unity/Alerts/1.0/<Relative-URL>
```

Create a base condition

To create a base condition, use the `BaseCondition` relative URL and the HTTP POST method.

To specify the base condition attributes, use the URL and a POST method with the following JSON in the POST body:

```
{ "name": "<base_condition_name>",
"description": "<base_condition_description>",
"baseConditionTemplateName": "<template_name>",
"datasourceName": "<datasource_name>",
"parameterValues": { "<parameter_1>" : "<value_1>",
"<parameter_2>" : "<value_2>"},
"actions": ["<action_1>", "<action_2>", "<action_3>"]
}
```

- *<base_condition_name>* is the name.
- *<base_condition_description>* is the description.
- *<template_name>* is the template that you want to use as the basis for the condition instance.
- *<datasource_name>* is the name of the data source that the condition monitors.
- *<parameter_1>* and *<value_1>* are the parameters and values that you want to monitor.
- *<action_1>* is the action that is triggered when the condition is met.

For example:

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

Return details for a base condition

To return the details for a base condition, use the `BaseCondition/`*`<base_condition_name>`* relative URL. Use the HTTP GET method for this action.

Return details for all base conditions

To return the details for all base conditions, use the `BaseCondition` relative URL. Use the HTTP GET method for this action.

Delete a base condition

To delete a base condition, use the `BaseCondition/`*`<base_condition_name>`* relative URL. Use the HTTP DELETE method for this action.

Update a base condition

To update a base condition, use the `BaseCondition/`*`<base_condition_name>`* relative URL. You can only update the description and the parameter values.

To specify the description or the parameter values that you want to update, use the URL and a PUT method with JSON in the following format in the PUT body:

```
{ "name": "<base_condition_name>",
"description": "<updated_base_condition_description>",
"parameterValues": { "<parameter_1>" : "<new_value_1>",
"<parameter_2>"": "<new_value_2>"
}
```

- *<base_condition_name>* is the name of the base condition that you want to update.
- *<updated_base_condition_description>* is the new description that you want to use to update the base condition.
- *<parameter_1>* and *<new_value_1>* is the parameter name and value that you want to update.

Enable a base condition

To enable a base condition, use the `BaseCondition/<base_condition_name>/enable` relative URL. Use the HTTP POST method for this action.

Disable a base condition

To disable a base condition, use the `BaseCondition/<base_condition_name>/disable` relative URL. Use the HTTP POST method for this action.

Add an action to a base condition

To add an action to a base condition, use the `BaseCondition/<base_condition_name>/AddAction/<action_name>` relative URL. Use the HTTP POST method for this action.

Remove an action from a base condition

To remove an action from a base condition, use the `BaseCondition/<base_condition_name>/RemoveAction/<action_name>` relative URL. Use the HTTP POST method for this action.

Alerting REST API for composite conditions

Standard

Use the Alerting REST API to create, get, update, and delete composite conditions.

To complete an action, enter the following URL and add the action:

```
https://<server>:<port>/Unity/Alerts/1.0/<Relative-URL>
```

Create composite condition

To create a composite condition, use the `CompositeCondition` relative URL and a POST method.

To specify the composite condition attributes, use the URL and a PUT method with JSON in the following format in the POST body:

```
{
"name": "<composite_condition_name>",
"description": "<composite_condition_description>",
"compositeConditionTemplateName": "<composite_condition_template_name>",
"parameterValues": { "<parameter_1>" : "<value_1>",
"<parameter_2>" : "<value_2>"},
"inputConditions": ["<condition_1>", "<condition_2>",
"<condition_3>"],
"actions": ["<action_1>", "<action_2>", "<action_3>"
]
}
```

- *<composite_condition_name>* is the name of the composite condition.
- *<composite_condition_description>* is the description of the composite condition.
- *<composite_condition_template_name>* is the name of the composite condition template that you want to base this instance on.
- *<parameter_1>* and *<value_1>* are the parameters and values that you want to monitor.
- *<action_1>* is the action that is triggered when the conditions are met.

Return the details for a composite condition

To return the details for a composite condition, use the `CompositeCondition/<Composite_condition_name>` relative URL.

Return the details for all composite conditions

To return the details for all the composite conditions, use the `CompositeCondition` relative URL.

Delete a composite condition

To delete the details for a composite condition, use the `CompositeCondition/<Composite_condition_name>` relative URL.

Update a composite condition

To update a composite condition, use the `CompositeCondition/<Composite_condition_name>` relative URL. You can only update the description and parameter values.

To specify the description or the parameter values that you want to update, use the URL and a PUT method with JSON in the following format in the POST body:

```
{ "name": "<composite_condition_name>",
"description": "<updated_composite_condition_description>",
"parameterValues": { "<parameter_1>" : "<new_value_1>",
"<parameter_2>"": "<new_value_2>"
}
```

- *<composite_condition_name>* is the name of the composite condition that you want to update.
- *<updated_composite_condition_description>* is the new description that you want to use to update the composite condition.
- *<parameter_1>* and *<new_value_1>* is the parameter name and value that you want to update.

Enable a composite condition

To enable a composite condition, use the `CompositeCondition/<Composite_condition_name>/enable` relative URL. Use the HTTP POST method for this action.

Disable a composite condition

To disable a composite condition, use the `CompositeCondition/<Composite_condition_name>/disable` relative URL. Use the HTTP POST method for this action.

Add an action to a composite condition

To add an action to a composite condition, use the `CompositeCondition/<Composite_condition_name>/AddAction/<action_name>` relative URL. Use the HTTP POST method for this action.

Remove an action from a composite condition

To remove an action to a composite condition, use the `CompositeCondition/<Composite_condition_name>/RemoveAction/<action_name>` relative URL. Use the HTTP POST method for this action.

Alerting REST API for alert actions

Standard

Use the Alerting REST API to create, get, update, and delete alert actions.

To complete an action, enter the following URL and add the action:

```
https://<server>:<port>/Unity/Alerts/1.0/<Relative-URL>
```

<Parameter> is the name of the parameter that you want to use.

Create an alert action

To create an alert action, use the `AlertAction` relative URL and a POST method.

To specify the alert action attributes, use the URL and a POST method with JSON in the following format in the POST body:

```
{
"name": "<alert_action_name>",
"description": "<alert_action_description>",
"alertActionTemplateName": "<template_name>",
"parameterValues": {"<parameter_1>" : "<value_1>",
"<parameter_2>" : "<value_2>"},
}
```

- *<alert_action_name>* is the name of the alert action that you want to create.
- *<alert_action_description>* is the description of the alert action.
- *<template_name>* is the name of the template that you want to use to create the alert action.
- *<parameter_1>* and *<value_1>* are the parameters and values that you want to add to the action.

Retrieve the details for an alert action

To return the details for an alert action, use the `AlertAction/<alert_action_name>` parameter.

Retrieve the details for all alert actions

To return the details for all the alert actions, use the `AlertAction` parameter.

Delete an alert action

To delete an alert action, use the `AlertAction/<alert_action_name>` parameter.

Update an alert action

To update an alert action, use the `AlertAction/<alert_action_name>` parameter. You can only update the description and parameter values.

To specify the description or the parameter values that you want to update, use the URL and a PUT method with JSON in the following format in the PUT body:

```
{ "name": "<alert_action_name>",
"description": "<alert_action_description>",
"parameterValues": { "<parameter_1>" : "<new_value_1>",
"<parameter_2>"": "<new_value_2>"
}
```

- *<alert_action_name>* is the name of the alert action that you want to update.
- *<alert_action_description>* is the new description that you want to use to update the alert action.
- *<parameter_1>* and *<new_value_1>* is the parameter name and value that you want to update.

Enable an alert action

To enable an alert action, use the `AlertAction/<alert_action_name>/enable` relative URL. Use the HTTP POST method for this action.

Disable an alert action

To disable an alert action, use the `AlertAction/<alert_action_name>/disable` relative URL. Use the HTTP POST method for this action.

REST API for asynchronous searches

You can use HTTP methods such POST, GET, and DELETE to customize your asynchronous searches.

GET method

You can use the GET method to return the details of all current searches.

The user ID is implied during authentication when the method is called.

The GET method returns the information that is contained in the search request properties.

You can use the currentState property to determine the status of a search.

Request values

Table 1 outlines the values that you must specify in the request.

Table 95. Request values for GET

Name	Type	Default value	Description
Size	Number	30	Indicates the maximum number of records that are returned. To return all records, specify -1.
Start	Number	0	Index number for the first record that is returned.

Returned values

Table 2 outlines the values that are returned by the function.

Table 96. Returned values for `GET`

Attribute	Description
`currentState`	Returns the status of a search. Possible values are `QUEUED`, `PARSING`, `RUNNING`, `PAUSED`, `FINALIZING`, `FAILED`, `DONE`.
`completionProgress`	A number from 1 - 100 that indicates an approximation of the progress of the search.
`isDone`	Indicates that the search is finished.
`isFailed`	Indicates that the search failed with an unrecoverable error. For example, if the search string uses an incorrect syntax.
`isPaused`	Indicates that the search is paused.
`messages`	Contains the error and debugging messages that are generated during the search.
`priority`	A number from 1 - 10 that indicates the priority of the search.
`searchRequest`	Lists the JSON file that is used to define the search.

`POST` method

You can use the `POST` method to start a search and return the search request ID.

You use the `search` parameter to specify a JSON file. The file specifies a search query that contains information about the search string, time filter, facets, and more.

The function returns the search request ID. You can add these amendments to the search URL to view and manage the search. Table 1 outlines these amendments.

Table 97. Search URL amendments

URL amendment	Description
`search/<search_request_ID>`	View the status of the search request.
`search/<search_request_ID>/<action>`	Run the action commands that you specify in the `<action>` parameter. Some possible actions are pause, cancel, and preview.
`search/<search_request_ID>/results`	View the search results.

Request values

Table 2 outlines the values that you must specify in the request.

Table 98. Request values for `POST` function

Name	Type	Default value	Description
`search`	JSON	n/a	JSON file that specifies the search query details such as the search string, time filter, facets and more. The JSON specification is required for this function.
`mode`	Enumeration	normal	The valid values are `blocking` and `async`. If the mode is set to `async`, the search runs asynchronously. If the mode is set to `blocking`, the function returns the search request ID when the search finishes.
`timeout`	Number	86400	The number of seconds that the search is retained for after processing stops.

Returned values

If it is successful, the function returns the status and the service request ID. If it is not successful, the function returns an error code.

`DELETE/search/<search_request_id>` method

You can use the `DELETE/search/<search_request_id>` method to delete the search that is specified by the `GET` method.

Request values

You do not need to specify any parameters for the request. The operation deletes the search request that is specified in the `GET` operation.

Response codes

The two possible response codes are outlined in table 1.

Table 99. Response codes

Status code	Description
200	Search deleted successfully.
404	The search request does not exist.

Returned values

The operation returns the same values as the `GET` operation. For more information, see "`GET` method" on page 397.

`GET/search/<search_request_id>` method

You can use the `GET/search/<search_request_id>` method to return details about the search request that is associated with the user who runs the call.

The method returns information about the search request properties such as the time taken to complete the search.

The parameters to `POST /search` provides details on search request properties when creating a search.

Request values

There are no request values as the operation uses the user ID of the user who runs the operation to identify the search request ID.

Response codes

The response codes are outlined in table 1.

Table 100. Response codes

Status code	Description
`200`	Search request was retrieved successfully.
`403`	Cannot retrieve search request. User does not have the authorization to view the search request.
`404`	The search request does not exist.

Return values

The operation also returns a JSON file that contains information about the search request.

`POST/search/<search_request_id>/action` method

You can use the `POST/search/<search_request_id>/action` method to request an action for a specific search request.

Request values

Table 1 outlines the request values that you can use with this operation.

Table 101. Request values

Name	Type	Default	Description
`name`	Enumeration	n/a	The action that is carried out. Valid values are `pause`, `unpause`, `cancel`, `setpriority`

The following values are valid for the `name` request:

pause Suspends the current search.

unpause
Resumes the current search if it is paused.

cancel Stops the current search and deletes the cached results.

setpriority
Sets the priority of the search request. This value can be any value from 1 - 10.

Response codes

The response codes are outlined in table 2.

Table 102. Response codes

Status code	Description
200	Search updated successfully.
403	User does not have the authorization that is required to edit the search request.
404	The specified search request does not exist.

Returned values

The operation returns the status of the search request and any related messages.

GET/search/<*search_request_id*>/results method

You can use the GET/search/<*search_request_id*>/results method to return the results of the specified service request at the time the request is made.

Response codes

The response codes are outlined in table 1.

Table 103. Response codes

Status code	Description
200	Results that are returned successfully.
204	Search exists but the search results are not ready. Run the request again.
403	User does not have the authorization that is required to display the specified search request.
404	The specified search request does not exist.

Returned values

The request returns the status and the related values. For example, a successful request returns the JSON file that contains the search results and related facets.

Loading and streaming data

Before you can perform a search on log or other data, you must first load the data into IBM Operations Analytics - Log Analysis. When the file is loaded the data is indexed and is then available to be searched.

There are two main scenarios for loading data:

- Batch loading historic data. For example, you may want to ingest historic log data in a single batch for analysis or for testing.
- Streaming data from a monitored application. You may want to load data that is streamed from a local or remote server.

You can load or stream data from local or remote servers. However, each tool is designed for a particular scenario. This is explained in the *Intended uses of data loading components* table. IBM Operations Analytics - Log Analysis is installed with an internal version of the IBM Tivoli Monitoring Log File Agent. However, IBM Operations Analytics - Log Analysis can also load data from a separate installation of the IBM Tivoli Monitoring Log File Agent, known as an external IBM Tivoli Monitoring Log File Agent.

Table 104. Intended uses of data loading components

	Load batch of historic data		**Stream data**	
Component	**Local**	**Remote**	**Local**	**Remote**
Data Collector client	Yes	Yes	No	No
Internal IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
External IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
logstash	No	No	No	Yes
Generic Receiver	Yes	Yes	No	No

Note: You must create a Data Source before you configure data loading. For information about creating a Data Source, see the *Administering IBM Operations Analytics - Log Analysis* section of the Information Center. For an overview of the process that you must follow to configure and use IBM Operations Analytics - Log Analysis, see the *Steps to get started with IBM Operations Analytics - Log Analysis* topic in the *Overview of IBM Operations Analytics - Log Analysis* section of the Information Center.
You can load log data into IBM Operations Analytics - Log Analysis using a number of different methods:

Data Collector client

Use the Data Collector client to ingest data in batch mode. This is the easiest method if you want to ingest a large log file for historic analysis if you want to test your IBM Operations Analytics - Log Analysis configuration before attempting the more complex IBM Tivoli Monitoring

Log File Agent configuration. The Data Collector client is not designed for ingesting data from remote sources. If you want to ingest a batch of historical data from a remote source, use the IBM Tivoli Monitoring Log File Agent.

For a video that demonstrates how to batch upload a WebSphere Application Server or DB2 file using the Data Collector client, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Videos. For information about batch uploading alternative log file types such as Oracle alert logs, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Batch%20uploading%20Oracle%20Alert%20logs.

IBM Tivoli Monitoring Log File Agent

Use the IBM Tivoli Monitoring Log File Agent for scenarios where you want to stream log data from your production environment or to stream data from a remote server.

For a video that demonstrates how to upload a WebSphere Application Server or DB2 file using the IBM Tivoli Monitoring Log File Agent, see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Videos.

logstash

logstash can be used as a method to collect and load data into IBM Operations Analytics - Log Analysis using the logstash Integration Toolkit. For information about the logstash Integration Toolkit, including information about how to download and install it, see logstash Integration Toolkit.

Generic Receiver

Use the Generic Receiver to load data from the REST interface into IBM Operations Analytics - Log Analysis.

You cannot use IBM Operations Analytics - Log Analysis to index log records that contain non-ASCII characters. If your log records contain non-ASCII characters, the records are not added when you use the IBM Tivoli Monitoring Log File Agent or the Data Collector client. When you use the Data Collector client errors that relate to non-ASCII characters are added to the Generic Receiver log.

Example scenarios

The following table outlines a number of example scenarios to help illustrate how you use the different components for different scenarios.

Table 105. Example data loading scenarios

Example	Component
I want to load a batch of historic log data to test the environment.	Data Collector client
I want to monitor an application on a remote server.	IBM Tivoli Monitoring Log File Agent
I want to use logstash to monitor log files on a remote server.	logstash
I want to load a batch of historic log data in the JSON format.	Generic Receiver

Supported operating systems

The supported operating systems that you can install IBM Operations Analytics - Log Analysis are listed in the *Installing* Guide. In addition to these, you also need to know what operating systems are supported by the data streaming and loading scenarios. For example, if you want to use the internal to stream data from a remote source, you need to know the supported operating systems.

Table 106. Supported operating systems for data loading

Scenario	**Feature**	**Supported operating systems**
Use the Data Collector to load a batch of historic data	Data Collector	• Red Hat Enterprise Linux Server Edition Version 5 or Version 6 (64 bit) • SUSE Linux Enterprise Server 11 (64 bit)
Use the internal IBM Tivoli Monitoring Log File Agent to stream data	Internal IBM Tivoli Monitoring Log File Agent	See the *Requirements for the monitoring agent* topic in the documentation for your version of IBM Tivoli Monitoring at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring
Use an external IBM Tivoli Monitoring Log File Agent to stream data	External IBM Tivoli Monitoring Log File Agent	See the *Requirements for the monitoring agent* topic in the documentation for IBM Tivoli Monitoring 6.2.3.1 at:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring

Configuring data streaming

Before you can stream data, you must configure the tools that you use to send the data to IBM Operations Analytics - Log Analysis.

IBM Tivoli Monitoring Log File Agent configuration scenarios

You can use the internal IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis or you can use an external IBM Tivoli Monitoring Log File Agent to stream data from local or remote servers.

You can also use the IBM Tivoli Monitoring Log File Agent to upload a batch of historic data. For more information, see “Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent” on page 265.

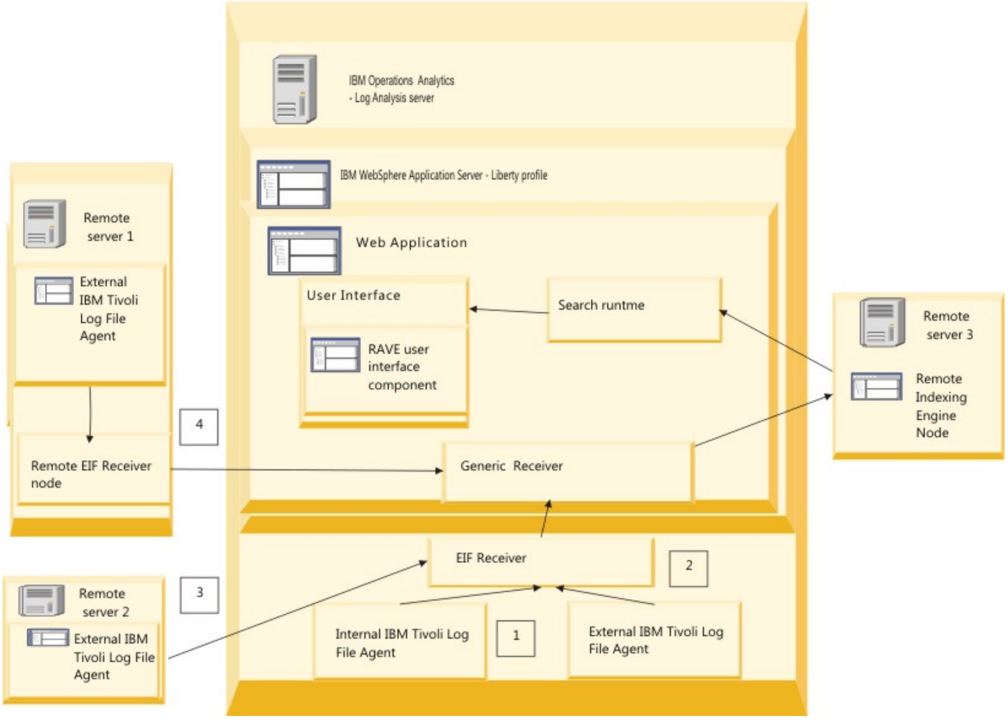
You can integrate the IBM Tivoli Monitoring Log File Agent with IBM Operations Analytics - Log Analysis in two ways.

You can use it with the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis. This is known as the internal IBM Tivoli Monitoring Log File Agent.

You can also use it with an IBM Tivoli Monitoring Log File Agent that has been installed separately as part of another installation.

You can use local and remote versions of both types of IBM Tivoli Monitoring Log File Agent.

The following graphic illustrates these possibilities:


IBM Operations Analytics - Log Analysis server
IBM WebSphere Application Server - Liberty profile
Web Application
User Interface
Search runtme
RAVE user interface component
Generic Receiver
EIF Receiver
Internal IBM Tivoli Log File Agent
External IBM Tivoli Log File Agent
1
2
3
4
Remote server 1
External IBM Tivoli Log File Agent
Remote EIF Receiver node
Remote server 2
External IBM Tivoli Log File Agent
Remote server 3
Remote Indexing Engine Node

The following possible scenarios are illustrated in the graphic:

1. Internal IBM Tivoli Monitoring Log File Agent on a local server
In this scenario, you use the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis to load data from the local installation of IBM Tivoli Monitoring to IBM Operations Analytics - Log Analysis.

2. External IBM Tivoli Monitoring Log File Agent on a local server
In this scenario, you use a version of the IBM Tivoli Monitoring Log File Agent that was not installed with IBM Operations Analytics - Log Analysis but that is installed on the same server as IBM Operations Analytics - Log Analysis.

3. External IBM Tivoli Monitoring Log File Agent on a remote server
In this scenario, you use an installation of an external IBM Tivoli Monitoring Log File Agent to push data to IBM Tivoli Monitoring Log File Agent. To facilitate this integration, you modify the properties of the IBM Tivoli Monitoring Log File Agent.

4. Remote instance of the internal IBM Tivoli Monitoring Log File Agent

In this scenario, you use a the remote installer tool to install a remote instance of the internal IBM Tivoli Monitoring Log File Agent.

The following table summarizes the different configurations required for the scenarios.

Table 107. Configuration for data streaming scenarios

Data streaming scenario	IBM Tivoli Monitoring Log File Agent type	Log file location	Required parameters in `.conf` file
1	Internal and local	Local	`Datasources`
2	Internal and remote. You use the remote installer to create the remote instance.	Remote	`Datasources`, `ServerLocation`, `ServerPort`, `BufEvtMaxSize`.
3	Local and external	Local	`Datasources`
4	Remote and external	Remote	`Datasources`, `SshAuthType`, `SshHostList`, `SshPassword`, `SshPort`, `SshPrivKeyfile`, `SshPubKeyfile`, `SshUserid`.

Configuring IBM Tivoli Monitoring Log File Agents for use with IBM Operations Analytics - Log Analysis

You can configure IBM Tivoli Monitoring Log File Agents to start IBM Operations Analytics - Log Analysis.

About this task

For more information about how configuration and format files are used, see “IBM Operations Analytics - Log Analysis configuration and format files” on page 234.

For more information about the required parameters in the configuration file, see “Configuration file parameters” on page 202.

CAUTION:

You cannot use non-ASCII characters in the installation path. The installation path cannot exceed 80 characters.

For more information, about this and about how to configure the monitoring agent in step 3 see:

http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.2/com.ibm.itm.doc_6.3fp2/install/unixconfig_ma.htm?lang=en

Procedure

1. To configure IBM Tivoli Monitoring Log File Agent, run the command:

```
./itmcmd config -A pc
```

where pc is the product code for your agent. For example: ./itmcmd config –A lo.

2. You are prompted to supply the following information:

   Enter instance name (default is:):
   Enter the instance name. For example, *rhelagent*.

   Conf file (default is:):
   Enter the configuration file path. For example, /unity/IBM/ITM/config/lo/.

   Format File (default is:):
   Enter the format file path. For example, /unity/IBM/ITM/config/lo/.

   Note: All fields must be completed. Blank fields cause IBM Tivoli Monitoring Log File Agent to fail.
3. Where prompted, provide the monitoring agent configuration information.
4. To start the IBM Tivoli Monitoring Log File Agent, run the command

   ```
   ./itmcmd agent  -o instance name start lo
   ```

Configuring IBM(r) Tivoli(r) Monitoring Log File Agents

Before you use the IBM Tivoli Monitoring Log File Agent, you may want to modify the configuration and format files.

About this task

For more information about how configuration and format files are used, see "IBM Operations Analytics - Log Analysis configuration and format files" on page 234.

For more information about the required parameters in the configuration file, see "Configuration file parameters" on page 202.

Procedure

1. Open the configuration file that you want to use.
2. Define the required parameters in the configuration file. The required parameters are different depending on the data loading scenario.
   - If you want to stream data from a local server, specify the data sources in the DataSources parameter.
   - If you want to push data from a remote directory, you must specify values for the Datasources, ServerLocation, ServerPort, and BufEvtMaxSize parameter.
   - If you want to use an external IBM Tivoli Monitoring Log File Agent that is not installed as part of IBM Operations Analytics - Log Analysis, you must specify values for the Datasources, SshAuthType, SshHostList, SshPassword, SshPort, SshPrivKeyfile, SshPubKeyfile, and SshUserid parameters.
3. Define the format file as required.
4. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the .fmt file for each rotating log type to allow for the appropriate name change. Open the .fmt file, and edit the line:

   ```
   -file FILENAME
   ```

   to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the FILENAME must read:

   ```
   -file SystemOut*.log
   ```

5. Save your changes.

Example

For example:

```
===============
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PASSWORD
SshPassword=<password>

====================
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PUBLICKEY
SshPrivKeyfile = <SshUserid_Private_Key_File_Path>
(Or)
SshPubKeyfile = <SshUserid_Private_Key_File_Path>

=====================
```

where <*password*> is the password that you want to use.

<*SshUserid_Private_Key_File_Path*> is the full path for the file that contains the private key of the user that is specified in the SshUserid user. For example, if you save the password to a file called `password.txt` in the `<HOME>/utilities` directory, the full path is as follows:

```
SshPrivKeyfile = <HOME>/utilities/password.txt
```

Configuring IBM Tivoli Monitoring Log File Agent subnodes

Create an IBM Tivoli Monitoring Log File Agent subnode to group an explicit set of configurations that the IBM Tivoli Monitoring Log File Agent uses to identify and process a log event.

About this task

The subnode consists of a format (`.fmt`) file and a configuration (`.conf`) file. A single instance of the IBM Tivoli Monitoring Log File Agent can have multiple subnodes. Each subnode behaves like a single thread running in the same instance of the IBM Tivoli Monitoring Log File Agent.

You can create subnodes for the following use cases:

Improve performance by making generic format settings more specific
To improve overall performance, you can create specific configurations to replace more generic ones. For example, you can specify the same regular expression (REGEX) in a generic `.fmt` file to parse both WebSphere Application Server (WAS) and DB2 log files. However as the content of the log files differs, this is inefficient. To improve performance, replace the single `.fmt` files with 2 new files containing 2 specific REGEXs for WAS and DB2 in 2 new subnodes.

Improve performance by making generic configuration settings more specific
Similarly, you can improve performance by replacing generic configurations with more specific ones. For example, you can specify the same roll over behaviour in a generic `.conf` to process both WAS and DB2 log files. However as the roll over behaviour in the log files differs, this

configuration results in some of the logs not being processed correctly and is inefficient. To improve performance, replace the single `.conf` with 2 new files in 2 new subnodes.

Improve performance for many data sources
If you use a large number of data sources to monitor the log events, you can create subnodes to spread the workload.

Remote monitoring with IBM Tivoli Monitoring Log File Agent 6.3
With IBM Tivoli Monitoring Log File Agent 6.3, you can modify the `.conf` file to monitor logs from multiple remote sources. However, the user credentials may not be the same for the remote machines. You can only maintain 1 set of user credentials in the IBM Tivoli Monitoring Log File Agent configuration file. In this case, you create multiple subnodes with different user credentials in each. This allows you to monitor multiple remote sources from a single IBM Tivoli Monitoring Log File Agent node with multiple subnodes.

There is also a limitation on the naming of subnodes. For more information, see "Character limits for IBM Tivoli Monitoring Log File Agent subnodes names" on page 232.

Procedure

1. Go to the directory where the IBM Tivoli Monitoring Log File Agent is installed. For example, if you are using the internal IBM Tivoli Monitoring Log File Agent that is delivered with IBM Operations Analytics - Log Analysis, the directory is `<Add_path>`.
2. To open IBM Tivoli Monitoring Log File Agent configuration window, run the following command:
   ```
   bin/CandleManage
   ```
3. Right click on the **Tivoli Log File Agent** service and click **Configure**.
4. Click on the instance that you want to configure and click **OK**. The **Configure Tivoli Log File Agent** window is displayed.
5. On the **Log File Adapter Configuration** tab, ensure that the **Conf file** and **Format File** fields are blank.
6. Click on the **Log File Adapter Global Settings** tab and note the directory that is specified in the **Configuration file autodiscovery directory**. This is the directory where you will save the subnode configuration files. Click **OK**.
7. In the subsequent window, you can ignore the other changes and click **Save** to save your changes.
8. Enter the root user password when prompted to implement your changes.
9. Copy the subnode configuration files to the directory that you noted in step 6. The IBM Tivoli Monitoring Log File Agent automatically detects the changes. You do not need to restart the IBM Tivoli Monitoring Log File Agent instance.

Results

The procedure describes how to configure subnodes in the IBM Tivoli Monitoring Log File Agent UI. You can also use the command line. To use the command line to configure the subnodes:

1. Go to the directory where the IBM Tivoli Monitoring Log File Agent is installed.
2. Run the following command:
   ```
   itmcmd config -A lo
   ```

3. Follow the onscreen instructions.
4. Specify the configuration file autodiscovery directory.
5. Complete the configuration.
6. Save the subnode configuration file to the directory that you specified in step 4.

Character limits for IBM Tivoli Monitoring Log File Agent subnodes names:

When you name a subnode, ensure that you are aware of the character and naming limitations.

32 character limitation

The IBM Tivoli Monitoring Log File Agent uses msn to name and identify the subnode. IBM Tivoli Monitoring limits the length of this name to 32 characters. The limit includes the identifier, the dash, and the semi-colon. This leaves 28 new characters for the host name, subnode and configuration file name.

The subnode name is specified in the following format:

```
LO:<Hostname>_<Subnode>-<Conffilename>
```

where `LO` is an identifier that is assigned to all subnodes. <*Hostname*> is the host name of the machine where the subnode is installed. <*Subnode*> is the name of the subnode.<*Conffilename*> is the name of the subnode configuration file.

For example:

```
LO:nc1234567890_WASInsightPack-lfawas
```

However, IBM Tivoli Monitoring limits the length of this name to 32 characters. The example name is 35 characters long. The limit includes the identifier, the dash, and the semi-colon, leaving 28 characters for the host name, subnode and configuration file name. To work around this limitation, IBM Tivoli Monitoring renames the subnode as:

```
LO:nc1234567890_WASInsightPack-l
```

This name is 32 characters long. The host name uses 12 characters. The subnode uses 14 characters.

This limitation can cause an issue if you use similar names for the configuration files. For example, after you name the first subnode, you create another subnode called:

```
LO:nc1234567890_WASInsightPack-lfawas2
```

IBM Tivoli Monitoring renames this subnode as:

```
LO:nc1234567890_WASInsightPack-l
```

As you can see, due to the truncation, both subnodes now have the same name, meaning that the IBM Tivoli Monitoring Log File Agent will not detect the new configuration.

Increasing the limit

The host name is used for integrations with Tivoli Endpoint Manager, where it helps you to identify subnodes. However, this is not required for IBM Operations

Analytics - Log Analysis. You remove the host name from the default naming convention so that you can use all 28 characters for the configuration file name.

To change the host name setting:

1. Stop the IBM Tivoli Monitoring Log File Agent.
2. Open the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/lo_default_workload_instance.conf` file.
3. Change the default value for the following property from `Y` (Yes) to `N` (No):

   ```
   CDP_DP_USE_HOSTNAME_IN_SUBNODE_MSN='N'
   ```
4. Save the updated file.
5. Restart the IBM Tivoli Monitoring Log File Agent.

After you change this configuration setting, the subnode name no longer includes the host name. For example, the subnodes in the previous example are now named `LO:WASInsightPack-lfawas` and `LO:WASInsightPack-lfawas2`.

Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data

If you use a IBM Tivoli Monitoring Log File Agent that is installed on a remote server to stream data to IBM Operations Analytics - Log Analysis, you must update the configuration and format files for the IBM Tivoli Monitoring Log File Agent.

Before you begin

You must create a custom data source before you configure data loading. For information about creating a data source, see “Data Source creation” on page 339.

About this task

You can use the configuration files in the `Unity_HOME/IBM-LFA-6.30/config/lo` directory as a basis for the configuration files on your remote server. However, you must ensure that the configuration files that you create:

- contain a line separator between each property that you define in the `.conf` file.
- use the `.conf` file extension and that the format file uses the `.fmt` extension.

To enable the IBM Tivoli Monitoring Log File Agent configuration, complete the following procedure:

Procedure

1. Specify a value or values for the `DataSources` property. If you have multiple locations, you can list the locations and use a comma as a separator. Ensure that you do not leave any spaces. For example, you specify the following values to represent 2 data sources:

   ```
   DataSources=/opt/IBM/WAS1/logs/SystemOut.log,/opt/IBM/WAS2/logs/SystemOut.log
   ```

   When you create a data source for a remote machine, you must enter the correct version of the host name for that machine. To find the correct host name, run the following command on the remote machine:

   ```
   uname -a
   ```

   Enter the name that is returned by this command in the host name parameter for the data source.

2. Specify the server location for the EIF receiver server. For example, for a server that is located at 111.222.333.444, specify the following value:

   ```
   ServerLocation=111.222.333.444
   ```
3. Specify the port that the EIF receiver uses. For example:

   ```
   ServerPort=5529
   ```
4. Specify the `BufEvtPath` for the LFA. The cached events are stored in this file. For example:

   ```
   BufEvtPath=/opt/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache
   ```
5. Specify the maximum buffer size for the LFA. This is the maximum size that the cache is allowed to be. If the cache is full, events are dropped and performance can decline. The value that you enter here is in kilobytes. For example:

   ```
   BufEvtMaxSize=102400
   ```
6. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the `.fmt` file for each rotating log type to allow for the appropriate name change. Open the `.fmt` file, and edit the line:

   ```
   -file FILENAME
   ```

   to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the `FILENAME` must read:

   ```
   -file SystemOut*.log
   ```
7. Save your changes.

What to do next

Allow time for the log data to be ingested and then search for a value contained in your log file to validate that the configuration has succeeded.

IBM Operations Analytics - Log Analysis configuration and format files

If you use an internal or external IBM Tivoli Monitoring Log File Agent, you can edit the configuration and property files to suit your specific installation.

The IBM Tivoli Monitoring Log File Agent configuration for a particular data source is defined in the following files:

- A `<name>.conf` file that contains the properties that are used by the IBM Tivoli Monitoring Log File Agent for processing the log files.
- A `<name>.fmt` file that contains an expression and format that is used by the agent to identify matching log file records and to identify the properties to include in the Event Integration Format (EIF) record. The EIF is sent from the agent to the receiving server. The receiving server is the server where the IBM Operations Analytics - Log Analysis server is installed. The `<name>.fmt` file uses a regular expression to determine matching records in the log file and to send each matching record to the IBM Operations Analytics - Log Analysis server in an EIF event.

If you want to use the IBM Tivoli Monitoring Log File Agent to send your log files to IBM Operations Analytics - Log Analysis server, you must customize the regular expression and define your own stanza in the `<name>.fmt` file to capture the log records that are to be sent. The event record format must include the host name, file name, log path, and text message. The IBM Operations Analytics - Log

Analysis server uses these values to process the logs. For more information about the IBM Tivoli 6.3 Log File Agent and the configuration files and properties, see Tivoli Log File Agent User's Guide.

The file names must be identical for both files. For example, `WASContentPack_v1.1.0-lfawas.conf` and `WASContentPack_v1.1.0-lfawas.fmt`.

After you modify the configuration files as required, you use the IBM Tivoli Monitoring Log File Agent to load the data into IBM Operations Analytics. For a general description of how to do this, see "Using the IBM Tivoli Monitoring Log File Agent" on page 237

If you use an external instance of the IBM Tivoli Monitoring Log File Agent to load data into the IBM Operations Analytics - Log Analysis server, you must install the configuration files into the agent. This configuration ensures that the agent knows where the log files for a data source are located, how to process the records in the log file, and the server to which records are sent.

LFA configuration file examples

The following example shows the files that are installed as part of the WebSphere Insight Pack that is included as standard with IBM Operations Analytics - Log Analysis.

The `WASContentPack_v1.1.0-lfawas.conf` file contains many properties, including the following examples:

```
# Files to monitor.  The single file /tmp/regextest.log, or any file like
/tmp/foo-1.log or /tmp/foo-a.log.
      LogSources=/home/unityadm/IBM/LogAnalysis/logsources
   /WASInsightPack/*

      # Our EIF receiver host and port.
      ServerLocation=<EIF Receiver host name>
      ServerPort=5529
```

The `WASContentPack_v1.1.0-lfawas.fmt` file contains the following regular expression that matches any record within a monitored log file. In this example, the regular expression matches all the log records in the file and to the Operations Analytics server as an EIF event. The EIF event contains the host name where the agent is running, the file name of the log file, the log file path of the log file, and the log file record itself.

```
 // Matches records for any Log file:
     //

     REGEX AllRecords
     (.*)
     hostname LABEL
     -file FILENAME
     logpath PRINTF("%s",file)
     text $1
     END
```

Configuration file parameters

The IBM Tivoli Monitoring Log File Agent uses the information that is specified in the configuration file to process log file information.

Table 1 explains that parameters that you can modify in this file.

Table 108. Parameter summary

Parameter	Description
`DataSources`	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA conf file, the directory you specify must not contain any subdirectories.
`SshAuthType`	You must set this value to either `PASSWORD` or `PUBLICKEY`. If you set this value to `PASSWORD`, IBM Operations Analytics - Log Analysis uses the value that is entered for `SshPassword` as the password for Secure Shell (SSH) authentication with all remote systems. If you set this value to `PUBLICKEY`, IBM Operations Analytics - Log Analysis uses the value that is entered for `SshPassword` as pass phrase that controls access to the private key file.
`SshHostList`	You use the `SshHostList` value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system. If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.
`SshPassword`	If the value of the `SshAuthType` parameter is `PASSWORD`, enter the account password for the user that is specified in the `SshUserid` parameter as the value for the `SshPassword` parameter. If the value of the `SshAuthType` parameter is `PUBLICKEY`, enter the pass phrase that decrypts the private key that is specified in the `SshPrivKeyfile` parameter.
`SshPort`	You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to `22`.

Table 108. Parameter summary (continued)

Parameter	Description
SshPrivKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshPubKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

Using the IBM Tivoli Monitoring Log File Agent

You can use the log file agent to load log file information into IBM Operations Analytics - Log Analysis.

Before you begin

Consider the size of the log files that you want to load. If a log file is in the region of 50 MB, or more, in size, increase the size of the log file agent cache. In the appropriate configuration file, set BufEvtMaxSize=102400. For WAS log files, update <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf. For DB2 log files, update <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf.

You must delete the appropriate existing cache file. For WAS log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache and for DB2 log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache

For very large log files, update the cache size of the EIF receiver. In the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf file, increase the value of the BufEvtMaxSize property.

Lines in a log that are longer than 4096 characters are, by default, ignored by the IBM Tivoli Monitoring Log File Agent. To force it to read lines longer than 4096 characters, add the EventMaxSize=<*length_of_longest_line*> property to the .conf file that will be used while loading the log.

For WAS update $UNITY_HOME/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf file. DB2 update $UNITY_HOME/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the `<HOME>/IBM/LogAnalysis/utilities` directory, run the following commands:

- `<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop`
- `<HOME>/IBM/LogAnalysis/utilities/unity.sh -start`

About this task

The IBM Tivoli Monitoring Log File Agent might be on the same server as IBM Operations Analytics - Log Analysis and monitoring a local directory. In this scenario, the installation of IBM Operations Analytics - Log Analysis completes all of the configuration required.

If the IBM Tivoli Monitoring Log File Agent is on the same server as IBM Operations Analytics - Log Analysis, but monitoring remote directories, some additional configuration is required. If you want to monitor log files on remote servers, you must make some specific settings changes. For more information about these specific settings, see the *Configuring remote monitoring that uses the predefined configuration files* topic under *IBM Tivoli Log File Agent Configuration* in the *Extending IBM Operations Analytics - Log Analysis* section.

If your configuration requires it, you can use a remote IBM Tivoli Monitoring Log File Agent. In this scenario, install and configure the IBM Tivoli Monitoring Log File Agent based on the your requirements. For more information, see the IBM Tivoli Monitoring documentation: http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/welcome.htm

Procedure

To use the log file agent to load log information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. Ensure that the log file you want to add is in the appropriate directory. For WAS logs, place the log file in the following directory:

   ```
   <HOME>/IBM/LogAnalysis/logsources/WASInsightPack
   ```

   For DB2 logs, place the log file in the following directory:

   ```
   <HOME>/IBM/LogAnalysis/logsources/DB2InsightPack
   ```

   For Generic annotator log files, place the log file in the following directory:

   ```
   $UNITY_HOME/logsources/GAInsightPack
   ```

   The log file is automatically picked up and analyzed. Depending on the size of the log file, processing it could take some time.
3. Optional: To monitor progress, check the following log files:
   - `<HOME>/IBM/LogAnalysis/logs/GenericReceiver.log`
   - `<HOME>/IBM/LogAnalysis/logs/UnityEifReceiver.log`

   When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `$UNITY_HOME/logs` directory to ensure that the data ingestion has completed correctly.

   This example illustrates the addition of a batch of log records. The result is indicated in the `RESPONSE MESSAGE` section of the log file:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO  - LogEventPoster : -
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 2078,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    ++++++++++++++++++++++++++++++++++++
2013-04-2 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    EIF event delivery to Generic Receiver -- SUCCESS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

In this log, the number of log records processed is indicated in the line:

```
{    "batchSize": 2078,    "failures": [    ],    "numFailures": 0 }
```

2078 log records were successfully ingested. The `numFailures` value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the `numFailures` value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a `TIMEOUT FLUSH` event. These events are added to the log file at the end of the session of data collection:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

2013-04-20 04:54:26,341 [pool-4-thread-1] INFO  - LogEventService :
 TIMEOUT FLUSH for logsource:nc9118041070::
  /home/example/LogAnalytics/logsources/
WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO  - LogEventPoster : ---
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 1714,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO  - LogEventService :
   POST RESULT:
{"failures":[],"batchSize":1714,"numFailures":0}

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

To calculate the number of events that have been processed, calculate the sum of all of the `batchSize` values. To calculate the number of events ingested, calculate the sum of all of the `batchSize` values and deduct the total sum of `numFailure` values.

If the ingestion fails, an error message is recorded in the `UnityEIFReceiver.log`:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster : Not Found
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
"CTGLA0401E : Missing log source ","RESPONSE_CODE":404}
```

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   FAILURE -  ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413 Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the $UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties.

500 Internal Server Error: Displayed when there is any issue withIBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404 Not Found: Displayed when a Log Source is not found for a hostname and log path combination in the request.

409 Conflict: Displayed if the data batch is posted for a Log Source that is an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the inputType field in the request JSON does not match the inputType field in the Collection for the requested hostname and log path combination.

200 OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400 Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Results

After the task completes, the log file is indexed and can be searched using the **Search** field on the IBM Operations Analytics - Log Analysis Dashboard.

Considerations when using the IBM Tivoli Monitoring Log File Agent

Before you configure the IBM Tivoli Monitoring Log File Agent to ingest data, update the IBM Tivoli Monitoring Log File Agent to ensure that the configuration is appropriate to the log file that you are likely to ingest.

Log file size

If your log files are likely to exceed 50 MB, increase the size of the IBM Tivoli Monitoring Log File Agent cache: In the appropriate configuration file, set BufEvtMaxSize=102400. For WAS log files, update <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf. For DB2 log files, update <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf.

You must delete the appropriate existing cache file. For WAS log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache and for DB2 log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache

For very large log files, update the cache size of the EIF receiver. In the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf file, increase the value of the BufEvtMaxSize property.

For WAS, update `<HOME>/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf` file. DB2 update `<HOME>/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf` file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the `<HOME>/IBM/LogAnalysis/utilities` directory, run the following commands:
- `unity.sh -stop`
- `unity.sh -start`

Maximum log line length

The IBM Tivoli Monitoring Log File Agent monitors each log file line. The default maximum line length that can be processed by the IBM Tivoli Monitoring Log File Agent is 4096 bytes. This is equivalent to 4096 ASCII characters. This limitation is related to the log line and not the log record. If a log record consists of multiple log lines, such as in the case of a stack trace, the limit applies to each line. This is a limitation of the IBM Tivoli Monitoring Log File Agent and does not apply if you use an alternative data collection mechanism.

Performance implications of using the IBM Tivoli Monitoring Log File Agent

Loading logs using the IBM Tivoli Monitoring Log File Agent is a CPU bound process. If your system does not meet the minimum requirements you will need to increase the `MaxEventQueueDepth`. On some systems, altering this value may produce a noticeable impact on performance. This will buffer additional IBM Tivoli Monitoring Log File Agent events while they are waiting to be processed. The required value for `MaxEventQueueDepth` may vary depending on the size of the rolled log and the number/speed of your CPU's. If you choose not to increase this value, then older events may be replaced on the event queue by newer events and not sent to the IBM Operations Analytics - Log Analysis server.

To minimize the chance of data loss due to CPU bottlenecks, and to reduce the latency between when a log record is written to the file and when it is loaded, we recommend that the maximum size of a log be small enough so that you system does not fall behind while processing the logs.

Common IBM Tivoli Monitoring Log File Agent configuration conflicts

When you create a remote IBM Tivoli Monitoring Log File Agent (LFA) node and a custom data source and both use the same log path, you can create a conflict.

When you create a custom data source and use it monitor a directory on a remote LFA subnode and you later create another data source, like a remote data source, that monitors the same directory, you can create a conflict in the LFA configuration. These conflicts may cause errors in the Log Analysis log files and reduce the performance of Log Analysis.

The following example is provided to help you to understand this situation.

To avoid these conflicts, you need to avoid monitoring the same directory with different data sources. If you want to monitor two files in the same directory, include the file name in the **Log Path** field when you create the data source.

Example

For example, you are an administrator and you want to monitor files from an LFA that is installed on a remote server as described in the Knowledge Center documentation. See "Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data" on page 233. In this case, the LFA is not part of the Log Analysis product.

First, you must create a custom data source called `Customdatasource` to load data from remote instance of the LFA. In the Data Source creation wizard, you specify the host name and the following log path:

```
/opt/WAS/WAS_logs/myLogFile.log
```

Next, you need to create the configuration and format files for the LFA sub nodes. You create two files, `lfa1.conf` and `lfa1.fmt`. In the `lfa1.conf` file, you specify the following data source:

```
Datasources=/WAS/WAS_logs/some_dir/*
```

Logs that are subsequently generated or appended are ingested by the `Datasource1` data source.

After some time, you create another data source to load data from the same remote server. The new log file is called `newLogFile.log` and it is located in the same directory as the file that you created the `Customdatasource` data source for. You create a remote data source called `Remotedatasource` and specify the log path as:

```
/opt/WAS/WAS_logs/newLogFile.log
```

Finally, you push the log files into Log Analysis.

However, after you push the log file, you notice some strange behaviour in the Log Analysis log files. The `GenericReceiver.log` log file shows that the data is being ingested for `/opt/WAS/WAS_logs/newLogFile.log`. However, it also says that the `/opt/WAS/WAS_logs/newLogFile.log` log file is not a valid data source.

This occurs because the same log file is being monitored by both data sources. As a result, it is monitored by two different LFA sub nodes and in two different streams. The data is loaded but this can waste resources and decrease the overall performance.

To avoid this situation, you must be aware of any possible conflicts especially when you create a custom data source that monitors a directory rather than a file.

Regular expression support for the LFA

The IBM Tivoli Monitoring Log File Agent (LFA) supports specific implementations of regular expressions.

Single-line unstructured data

If you want to use the DSV toolkit to extract and export the data in the comma-separated value (CSV) format for use with the DSV toolkit, you can use a regular expression to extract and export the data.

For example, consider the following log file record:

```
10453072 23460 E5D27197E653C548BDA744E8B407845B AOBEAI1 /EAI      I H R SACP9002
BPUSRSYS/612   23460 - XGNEA108:662:000042:06:E036977:WWS00003:7000:16:1:REV=N
Proc Time=000.03
```

You can configure Log Analysis to use a regular expression to extract and export the data in the comma-separated value (CSV) format. For example, here is an example of a regular expression that is defined in the `.fmt` file:

```
REGEX EAILOG
⌂([0-9]*)(.*)SACP9002(.*):([0-9]*):([0-9]*):([0-9]*):([a-zA-Z0-9]*):
([a-zA-Z0-9]*):([a-zA-Z0-9]*):
(.*)Proc Time=([0-9]*.[0-9]*)
timestamp $1 CustomSlot1
discard $2
SACP9002 $3
bankID $4 CustomSlot3
branchID $5 CustomSlot4
discard3 $6
tellerSID $7 CustomSlot5
workstationID $8 CustomSlot6
transactionTypeID $9 CustomSlot7
discard4 $10
responseTime $11 CustomSlot8
msg PRINTF("%s,%s,%s,%s,%s,%s,%s",timestamp,bankID,branchID,tellerSID,workstationID,
transactionTypeID,responseTime)
END
```

Manipulating date time information for the Generic Annotation Insight Pack

If you use the Generic Annotation Insight Pack or the date time rule set from the Generic Annotation Insight Pack in a custom Insight Pack, you can use some limited regular expressions that you can use to parse time and date information.

The second delimiter, which is a colon (`:`), is not supported. The regular expression replaces the second delimiter with a period (`.`), which is supported. For example, to change a date from `15/12/2014 12:12:12:088 GMT` to `15/12/2014 12:12:12.088 GMT`, you can add the following regular expression to the `.fmt` file:

```
// Matches records for any Log file:
// Log Analytics Data Source chas_access.log

REGEX nongr
([0-9][0-9])/([0-9][0-9])/([0-9][0-9]) ([0-9][0-9]):([0-9][0-9])
:([0-9][0-9]):([0-9][0-9][0-9]) ([A-Z][A-Z][A-Z])
(.*Batch Status for.*)
month $1
day $2
year $3
hour $4
minute $5
second $6
ms $7
zone $8
message $9
hostname example.com
-file /opt/la/IBM/LogAnalysis/logs/GenericReceiver.log
RemoteHost ""
logpath PRINTF("%s",file)
text PRINTF("%s/%s/%s %s:%s:%s.%s %s %s", month, day, year, hour, minute,
second, ms, zone, message)
END
```

Troubleshooting data loading

When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `<HOME>/logs` directory to ensure that the data ingestion has completed correctly.

This example illustrates the addition of a batch of log records. The result is indicated in the RESPONSE MESSAGE section of the log file:

```
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO  - LogEventPoster : -----------
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 2078,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    ++++++++++++++++++++++++++++++++++++
2013-04-2 04:43:24,273 [pool-5-thread-1] INFO  - LogEventPoster :
    EIF event delivery to Generic Receiver -- SUCCESS
```

In this log, the number of log records processed is indicated in the line:

```
{    "batchSize": 2078,    "failures": [    ],    "numFailures": 0 }
```

2078 log records were successfully ingested. The numFailures value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the numFailures value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a TIMEOUT FLUSH event. These events are added to the log file at the end of the session of data collection:

```
2013-04-20 04:54:26,341 [pool-4-thread-1] INFO  - LogEventService :
 TIMEOUT FLUSH for datasource:nc9118041070::
  /home/yogesh/IBM/LogAnalysis/logsources/WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO  - LogEventPoster : -----------
Posting Event to UNITY DATA COLLECTOR -
   https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   {    "batchSize": 1714,
"failures": [    ],    "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   ++++++++++++++++++++++++++++++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO  - LogEventPoster :
   EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO  - LogEventService :
   POST RESULT:
{"failures":[],"batchSize":1714,"numFailures":0}
```

To calculate the number of events that have been processed, calculate the sum of all of the batchSize values. To calculate the number of events ingested, calculate the sum of all of the batchSize values and deduct the total sum of numFailure values.

If the ingestion fails, an error message is recorded in the UnityEIFReceiver.log:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++ RESPONSE MESSAGE +++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster : Not Found
```

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
"CTGLA0401E : Missing data source ","RESPONSE_CODE":404}
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   +++++++++++++++++++++++++++++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO  - LogEventPoster :
   FAILURE -  ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413 Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the `$UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`.

500 Internal Server Error: Displayed when there is any issue withIBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404 Not Found: Displayed when a data source is not found for a hostname and log path combination in the request.

409 Conflict: Displayed if the data batch is posted for a data source that is an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the `inputType` field in the request JSON does not match the `inputType` field in the Collection for the requested hostname and log path combination.

200 OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400 Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Configuring the EIF Receiver

How to configure remote or local installations of the Tivoli Event Integration Facility (EIF) receiver to work with IBM Operations Analytics - Log Analysis.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Configuring receiver buffer size and timeout

When collecting data using the IBM Tivoli Monitoring Log File Agent (LFA) and Tivoli Event Integration Facility (EIF) Adapter flow, you might need to change the rate at which events are flushed to the generic receiver for indexing. Incoming events are buffered at the EIF receiver side.

About this task

To improve overall IBM Operations Analytics - Log Analysis performance, you can configure the 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. When the event rate decreases, decrease the buffer size and keep the timeout interval at the default value or increase it, depending on the event rate.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the buffer size and timeout parameters:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```

   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder used for the specific remote EIF instance.
3. Change the Timeout and Buffer Size parameters to suit your operating environment:

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

4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```

   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see "`eifutil.sh` command" on page 70.

Results

With higher buffer sizes, notice that it takes a longer time to fill the buffer with events and for batches to be posted to the receiver.

Configuring the EIF receiver user account

The Tivoli Event Integration Facility (EIF) receiver uses the default `unityuser` user account to access the generic receiver. You can change the user account or the default user password in the `unity.conf` configuration file.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the default EIF user or password:

1. Stop IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

  ```
  <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
  ```
- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder that is used for the specific remote EIF instance.
3. Change the following `userid` and `password` parameters to suit your operating environment:

   ```
   unity.data.collector.userid=unityuser

   unity.data.collector.password=password
   ```

   To encrypt the password, use the `unity_securityUtility.sh` command. For more information, see "Changing the default password for the Data Collector and EIF Receiver" on page 266.
4. Save your changes.
5. Restart IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to restart IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -restart` command to restart the instances. For more information, see "`eifutil.sh` command" on page 70.

Results

The EIF receiver uses the new credentials to access the generic receiver.

Configuring the number of events in the EIF Receiver

You can configure the number of events that the EIF Receiver stores for each internal queue. If you intend to ingest a large quantity of data and at a high rate, configure these values to larger values. However, increasing this value also increases the memory requirements for EIF Receiver.

About this task

Ensure that you have sufficient memory to support the number of events in the queue.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change this setting:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.
2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the <*remote_deployment_location*>`/LogAnalysis/DataForwarders/EIFReceivers/`<*eif_inst_#*>`/config/unity.conf` directory. Where <*remote_deployment_location*> is the directory on the remote machine where you deployed the EIF instance. <*eif_inst_#*> is the folder used for the specific remote EIF instance.
3. Locate these lines and change the value to reflect your requirements:

   ```
   unity.data.collector.eif.consumer.num.events=1000000
   unity.data.collector.event.manager.num.events=20000
   ```

   The following settings are applicable per data source:

   ```
   unity.data.collector.event.service.num.events=20000
   unity.data.collector.event.poster.num.events=500
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

Configuring the EIF Receiver memory clean up interval

IBM Operations Analytics - Log Analysis ensures that the memory used for data collection with the Log File Agent using a property in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf` file. The EIF Receiver uses this value to manage the memory usage. The configuration cycle is set to a value in minutes with a default value of 2 minutes.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To configure this property:

1. Stop IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see "`eifutil.sh` command" on page 70.

2. Open the configuration file for editing:
   - If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
   - If you use a remote installation of the EIF, the `unity.conf` file is in the `<`*`remote_deployment_location`*`>/LogAnalysis/DataForwarders/EIFReceivers/<`*`eif_inst_#`*`>/config/unity.conf` directory. Where *<remote_deployment_location>* is the directory on the remote machine where you deployed the EIF instance. *<eif_inst_#>* is the folder that is used for the specific remote EIF instance.
3. Change the parameters to suit your operating environment:
   ```
   #gc interval is in minutes
   unity.data.collector.gc.interval=2
   ```
4. Save your changes.
5. Start IBM Operations Analytics - Log Analysis:
   - If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:
     ```
     <HOME>/IBM/LogAnalysis/utilities/unity.sh -start
     ```
   - If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see "`eifutil.sh` command" on page 70.

Configuring scalable data streaming from multiple, remote sources

To facilitate dynamic data streaming that is scalable across multiple remote sources, you must configure IBM Operations Analytics - Log Analysis after you install it.

To enable data collection from remote hosts, you must complete the following steps:

1. Install Apache Solr on the remote machine.
2. Set up Secure Shell (SSH) communication.
3. Configure SSH to work with the remote installer utility.
4. Use the remote installer utility to install instances of the Event Integration Facility (EIF) or the IBM Tivoli Monitoring Log File Agent (LFA) on remote machines.
5. Configure the EIF so that it is compatible with the remote instances that your create. If you use the LFA, you do not have to configure the local installation. However, you do have to manually configure the sub nodes.

You can also maintain and administer these connections after you set them up.

As an alternative to streaming data, You can batch load data. For more information, see "Loading and streaming data" on page 223.

Installing Apache Solr on remote machines

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

About this task

If no local instances of Apache Solr exist, then you need to install the instances on the remote machine as soon as you install IBM Operations Analytics - Log Analysis. If there is a local instance of Apache Solr, you can install the remote instances whenever you want.

You must use a non-root user to run the script.

You cannot use the installer to install Apache Solr on a local machine.

You cannot use the installer to install multiple Apache Solr nodes on a single remote machine.

To install Apache Solr on multiple remote machines, run the script separately for each remote machine. You cannot use the installer to install instances of Apache Solr simultaneously or in parallel.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:

   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```

2. To run the `remote_deploy.sh` script, enter the following command:

   ```
   ./remote_deploy_solr.sh -install
   ```

3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password-less SSH authentication is disabled. If password-less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the <HOME>/IBM/LogAnalysis/utilities/config directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

   SSH Port
   Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

   Top-level Installation Directory
   To use the default value, which is `<HOME>`, press enter. Alternatively, you can enter the path to the directory where you want to install the DE.

   Apache Solr Search Port
   To use the default value, 9989, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

Apache Solr Query Service Port
To use the default value, 7205, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

4. To start the installation, press enter. In most cases, the installation takes about 5 minutes to complete.

Results

The results of the installation are output in the log file in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs/ManageSolrnodes.log` file.

To view the status for the instances of Apache Solr that are installed remote machines, run the `unity.sh -status` command.

Example

Here is an example script output:

```
Remote Hostname in FQDN format:12345.example.com
username:unity
password:*********
SSH port: [22]
Top-level Installation Directory: [/home/unity]
Solr Search Port: [9989]
Solr Query Service Port: [7205]

Script is ready for remote installation of Solr:
Review the following inputs ....
-----------------------------------------------------------------------------
Remote Host Name: 12345.example.com
Remote User Name: unity
Remote SSH Port: 22
Top-level remote installation directory: /home/unity
Solr v9.0 - remote installation directory:
/home/unity/IBM/LogAnalysis
Solr - remote ports: 9989, 7205
------------------------------------------------------------------------
['q' - Abort]['Enter' - Install]

Sat Nov 16 03:08:38 CST 2013 Starting remote installation of Solr
, this will take couple of minutes to complete  ....
Sat Nov 16 03:08:38 CST 2013 Waiting for remote installation to complete ....
Sat Nov 16 03:11:47 CST 2013 Successfully installed Solr
Solr on remote host:12345.example.com ....
```

Removing Apache Solr instances:

Before you remove an installation of IBM Operations Analytics - Log Analysis, you must remove Apache Solr.

About this task

Note: Do not remove Apache Solr if IBM Operations Analytics - Log Analysis is still being used. IBM Operations Analytics - Log Analysis does not function properly when any instances of Apache Solr are removed. For this reason, only remove Apache Solr when you are about to uninstall IBM Operations Analytics - Log Analysis.

If you installed Apache Solr locally and remotely, remove the local instance first, then remove the remotely installed instances.

This process uses Installation Manager to remove Apache Solr instances. You can also do so silently. To run the silent removal, run following `imcl -c` command, enter 3 to modify the installation, and remove the instance.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:

   ```
   cd <HOME>/IBM/LogAnalysis/solr_install_tool
   ```

2. To run the `remote_deploy.sh` uninstall script, enter the following command:

   ```
   ./remote_deploy.sh -uninstall
   ```

3. The script prompts you for the following information:

   Remote Hostname in FQDN format
   Enter the Fully Qualified Domain Name (FQDN) of the remote host.

   Username
   Enter the user name.

   Password
   Enter the password if password less SSH authentication is disabled. If password less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the `<*UNITY_HOME*>/utilities/config` directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

   SSH Port
   Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

   Top-level Installation Directory
   To use the default value, which is `<HOME>/IBM/LogAnalysis`, press enter. Alternatively, you can enter the path to the directory where Apache Solr is installed.

4. To start the removal, press enter. You can view the logs in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs` directory.

Results

When all the remote nodes are removed, you can safely uninstall IBM Operations Analytics - Log Analysis.

Setting up Secure Shell to use key-based authentication

Secure Shell (SSH) is a cryptographic network protocol for secure data communication between different computers. You set up key-based authentication between the IBM Operations Analytics - Log Analysis servers and the remote computers to which it connects.

About this task

Benefits of using key-based authentication:

- Data is transferred across a secure channel.
- The administrator is no longer concerned about the password changes for the remote servers.
- The passphrase is independent of the individual server password policy.

- One passphrase is used for multiple servers. Only the public key file must be copied to the client server.

For more information you can view the man pages for **ssh-keygen** by running this command:

```
man ssh-keygen
```

Procedure

1. To generate public and private keys, enter the following command:

   ```
   ssh-keygen -t rsa
   ```

   or either of the following commands:

   ```
   ssh-keygen
   (This command generates the same results as ssh-keygen -t rsa.)
   ```

   ```
   ssh-keygen -t dsa
   (If you specify dsa, the generated keys include _dsa in their file names.)
   ```

   The following example shows what a valid output might look like:

   ```
   bash-3.2$
   bash-3.2$ ssh-keygen -t rsa
   Generating public/private rsa key pair.
   Enter file in which you want to save the key (/home/unity/.ssh/id_rsa):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /home/unity/.ssh/id_rsa.
   Your public key has been saved in /home/unity/.ssh/id_rsa.pub.
   The key fingerprint is:
   4a:ef:d5:7a:d8:55:b3:98:a1:1f:62:be:dd:c4:60:6e unity@<variable>.example.com
   The key's randomart image is:
   +--[ RSA 2048]----+
   |                 |
   |                 |
   |                 |
   |             . ..|
   |      . S   .o+.o|
   |     . o   =o++. |
   |      . . +o+E.o |
   |       . ..o=.o  |
   |        . .o.. . |
   +-----------------+
   bash-3.2$
   ```

   Enter the passphrase. (The **Enter passphrase** field can remain blank to specify an empty passphrase.)

2. To view the contents of the public key file, run the following commands:

   ```
   cd ~/.ssh
   ls -l id_rsa*
   cat id_rsa.pub
   ```

   The command output is:

   ```
   bash-3.2$
   bash-3.2$ cat .ssh/id_rsa.pub
   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDg0/GGoxGzyC7Awjbwnp0hCaeztIRt6yhAg
   GKdwM7nb7Iiv0RgwT4/48E26K1Ur9HrI1W/j0K0JHQw
   vaAFibqeLmqLdK9ctCE901ywTOPFcYeBYPUF9vp/MgaypgGxVwDbW/e0SNPb7YAtZpjRoqeUq
   oYoKzFXXspQkxdhcQfpx0RYMbQdGGg03hDCM2wr2KP
   VuTVniF2IvDu1C4fcRkUPr8aQNMiuEcJgV3VHhlau/0Uo0YpH53NXKhn/sx8xdyTVsKQ1rhW8
   g07HIVc2Tf9ZF2gYXn/HbjE509xK/APu2nztt0h+Air
   JyT5jYMi/IvSI0zbPyc0p9WijPeG8r/v unity@<variable>.in.ibm.com
   bash-3.2$
   ```

3. Create a directory called `.ssh` on the remote server. Use this to store the public key.

4. Copy the public key file (`id_rsa.pub`) to the `.ssh` directory on the remote client:

```
scp /home/unity/.ssh/id_rsa.pub
<username>@<remotehostname>:/
<HOME>/.ssh/id_rsa.pub
```

where <*hostname*> is the system host name and <*username*> is the system user name.

5. Add the content of the public key to the `authorized_keys` file on the remote host.

   ```
   bash-3.2$ ssh <username>@<remotehostname>
   bash-3.2$ cd ~/.ssh
   bash-3.2$ cat id_rsa.pub >> authorized_keys
   bash-3.2$ rm id_rsa.pub
   bash-3.2$ exit
   ```

6. Ensure that there are no duplicate keys for the same client in the authorized_keys file.
7. Log in to the remote computer to ensure that key-based SSH is working:

   ```
   ssh <username>@<hostname>
   ```

   Enter the passphrase, if prompted.

   ```
   bash-3.2$ bash-3.2$ ssh <username>@<remotehostname>
   Enter passphrase for key '/home/unity/.ssh/id_rsa':
   Last unsuccessful login: Mon Jul 15 14:22:37 2013 on ssh from <variable>.example.com
   Last login: Mon Jul 15 14:26:54 2013 on ssh from <variable>.example.com
   $
   ```

   Configuration of key-based authentication is complete.

Results

The steps may not work because different versions of SSH are supported by the operating systems that are used by the remote servers. For more information about how to solve this issue, see the *Secure Shell (SSH) configuration does not work* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Configuring secure shell (SSH) communication for multiple remote hosts:

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

Before you begin

Before you configure SSH for multiple remote hosts, you must configure SSH between IBM Operations Analytics - Log Analysis and the remote hosts. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the Information Center.

About this task

By default, the SSH properties file, `ssh-config.properties` file, is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory. If you save the file to another location, the utility requests that the user enters values for the remote host, user, and password. In this case, the utility does not use the values specified in the file.

If you save the `ssh-config.properties` file in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory, the `eif_remote_install_tool` utility uses the properties specified in the file.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

If you specify values for both the password and the private key file path, the utility uses the file to create a password-less SSH connection.

If you do not specify a value for the password or the private key file path, IBM Operations Analytics - Log Analysis cannot create a connection and instead generates an error message in the log:

```
    ERROR:
    example.unity.remote.SshConfigException:
Property file config/ssh-config.properties must contain at least one of:
PASSWORD, PATH_OF_PASSWORD_LESS_SSH_KEY
    Correct SSH configuration OR reconfigure and retry
    Installation Aborted....!
```

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory and open the `ssh-config.properties` file.
2. Specify values for the following properties for each remote host:
   - Remote host
   - Remote user ID
   - Port
   - Connection timeout in milliseconds. The default is 6000.

   For example:

   ```
   REMOTE_HOST=<REMOTE_HOST>
   PORT=<PORT>
   TIME_OUT=60000
   USER=<REMOTE_USER>
   ```
3. For password-based authentication, you also need to specify the password in the configuration file. For example:

   ```
   PASSWORD=password1
   ```
4. For public key based authentication, specify the path to the directory that contains the private key file. For example:

   ```
   PATH_OF_PASSWORD_LESS_SSH_KEY=/home/pass/.ssh/id_rsa
   ```
5. If your installation of SSH requires a passphrase, specify the passphrase. For example:

   ```
   PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=passphrase1
   ```

Configuring data collection for scalability on multiple remote nodes

To facilitate scalable data collection on multiple remote nodes, use the `install.sh` command to install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server.

Before you begin

Before you run the command, you must configure secure shell (SSH) communication between the local installation of IBM Operations Analytics - Log Analysis and the remote host. For more information about how to do so, see "Configuring secure shell (SSH) communication for multiple remote hosts" on page 66.

About this task

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis.

You can use the remote installer in the following scenarios:

- If you have a high rate of data ingestion on multiple data sources. For example, if you have 100 or more events per second and 20 or more data sources.
- If you require improved throughput performance on the remote server.
- If the hardware resources on the remote server are restrained.
- If you want to optimize performance according to the conditions described on the Performance developer works page here: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM Log Analytics Beta/page/Performance and tuning

You can use the command to deploy up to 20 instances of the Tivoli Event Integration Facility Receiver or a single instance of the IBM Tivoli Monitoring Log File Agent on a remote node. The command deploys and configures IBM Java 1.7. The command also configures the deployed Tivoli Event Integration Facility Receiver instance to communicate with the IBM Operations Analytics - Log Analysis Data Collector interface.

However, this command does not configure the IBM Tivoli Monitoring Log File Agent subnode. You must configure this setting manually. Both the remote and local instance of the IBM Tivoli Monitoring Log File Agent can monitor remote data sources. For more information about configuring IBM Tivoli Monitoring Log File Agent, see "Configuring a remote IBM Tivoli Monitoring Log File Agent instance to stream data" on page 233.

To ensure that the remote instances of the Tivoli Event Integration Facility work with the local Data Collector interface, you must create the remotely deployedTivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances as part of the same installation. This is because the encryption configuration and signature generation is done during the main installation. If you install IBM Operations Analytics - Log Analysis after you set up the remote nodes, you must install the remote Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent instances again. However, you can remove remote instances of the Tivoli Event Integration Facility or IBM Tivoli Monitoring Log File Agent without installing IBM Operations Analytics - Log Analysis again.

Note: If you use the script to install the remote instance on a server that uses the SUSE Linux Enterprise Server 11 operating system, the script fails. To resolve this issue, see the *Cannot install remote EIF instance on SUSE* topic in the *Troubleshooting* IBM Operations Analytics - Log Analysis guide.

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory and run the `install.sh` command. You are prompted for a series of inputs.
2. Enter the remote installation directory. This value must be the location where the deployed artifacts are installed on the remote host.
3. If you want to deploy the Tivoli Event Integration Facility Receiver, select it. If you do, enter the Tivoli Event Integration Facility Receiver instances that you want to deploy.
4. If you want to deploy the IBM Tivoli Monitoring Log File Agent instance on the remote node, select it.

Results

After you complete the procedure, you can now collect data from the remote hosts.

What to do next

After the initial setup, you will want to periodically change the configuration. IBM provides two commands to start and stop the instances so that you can update the configuration.

To administer Tivoli Event Integration Facility Receiver instances, use the `eifutil.sh` command.

To administer IBM Tivoli Monitoring Log File Agent instances, use the `lfautil.sh` command.

`eifutil.sh` command:

To administer EIF Receiver instances, use the `eifutil.sh` command.

Syntax

The `eifutil.sh` command has the following syntax and is in the *<USER_HOME_REMOTE>*`/DataForwarders/EIFReceivers/utilities` where *<USER_HOME_REMOTE>* is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where <*Inst_ID*> is the ID for the specific EIF instance.

Parameters

`-status`
Displays the status for the installed instances. For example:

```
============================================================================
COMPONENT             Instance         PID            PORT            STATUS
============================================================================
EIF Receiver          eif_inst_1      13983           6601              UP
EIF Receiver          eif_inst_2      14475           6602              UP
EIF Receiver          eif_inst_3      14982           6603              UP
EIF Receiver          eif_inst_4      15474           6604              UP
EIF Receiver          eif_inst_5      15966           6605              UP
============================================================================
```

-start <*Inst_id*>
Starts the specified instance.

-stop <*Inst_id*>
Stops the specified instance.

-startAll
Starts all instances.

-stopAll
Stops all instances.

-restart<*Inst_id*>
Restarts the specified instance.

-restartAll
Restarts all the instances.

lfautil.sh command:

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the `lfautil.sh` command.

Syntax

The `lfautil.sh` command has the following syntax and is in the <*USER_HOME_REMOTE*>`/utilities/` directory on the remote host where <*USER_HOME_REMOTE*> is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start Starts all the LFA instances on the remote host.

-stop Stops all the LFA instances on the remote host.

-status
Displays the status for the LFA instances on the remote host. For example:

```
==========================================
COMPONENT            PID            STATUS
==========================================
Log File Agent         23995            UP
==========================================
```

-restart
Restarts the LFA instances on the remote host.

Loading batches of data

In addition to streaming data directly, you can also load batches of historic data for test or other purposes.

Generic Receiver

The Generic Receiver is a component of IBM Operations Analytics - Log Analysis that supports the REST interface for loading data into IBM Operations Analytics - Log Analysis. The REST API uses JSON (JavaScript Object Notation) as an input and returns JSON as an output after the incoming logs are processed. If an error occurs, the API returns an error code and a message.

Processing a batch

Invoking the Generic Receiver API initiates the processing of a batch that is contained in the Input JSON. Buffer a set of log records to create a batch and send data in batches to IBM Operations Analytics - Log Analysis. The batches must be sent in the order in which logs are generated for a specific data source. The size of each batch must be less than the batch size (500000 bytes) supported by IBM Operations Analytics - Log Analysis. At the minimum, you can send data for a single log record in a batch. The Generic Receiver processes a batch by:

- Splitting the batch into multiple log records by using the Splitter that was specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Annotates every log record that is found by the Splitter by using the Annotator that is specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Indexing the annotated log record in the back-end search engine

As special cases, split and annotated steps are skipped if the Splitter or Annotator is specified as null in the SourceType. Even if there is no data to split, you need to send an empty string in the text field of the Input JSON.

Batching of data at the client might lead to an incomplete log record at the end of the batch. This incomplete log record gets buffered in IBM Operations Analytics - Log Analysis and stitched with the remaining data in the subsequent batch to form a complete log record. This stitching assumes that you are maintaining the log record order of the data that is sent to IBM Operations Analytics - Log Analysis. If the order is not maintained, then logs are not correctly split into log records.

Input JSON

The basic structure of an Input JSON file is:

```
{
"hostname":  ,    (String)
"logpath":" ,   (String)
"batchsize": ,  (String)
"inputType":    // Optional (String) "LOGS";
"flush":   // Optional (boolean)
"payload":   // (JSONObject)
{
"name1":"value1",    // Optional
...
...
"nameN":"valueN" ,      // Optional
text : "log record 1 log record 2 ..."  (String)
 }
}
```

The following parameters in the Input JSON are mandatory:

hostname
The host name that corresponds to the data source for which you want to ingest data.

logpath
The log path that corresponds to the data source for which you want to ingest data.

batchsize
The number of BYTES of logs that are sent in one batch to IBM Operations Analytics - Log Analysis (less than 500,000).

inputType
The type of input data: `LOGS`.

flush flag
A flag that indicates to the Generic Receiver whether the last record in the batch is a complete log record. Typically, this flag would be set to true in the last batch upon reaching the end of file.

payload.txt
This text contains the actual log records to be split, annotated, and indexed into IBM Operations Analytics - Log Analysis. The text portion is split into log records by the Splitter, annotated by the Annotator, and then indexed. If you do not have any log records, but want to index only structured (name-value pairs) data, you can specify this mandatory field as an empty string.

More metadata (optional) to be indexed with every log record of the batch can be specified as name-value pairs in the input JSON or the payload within the input JSON. This metadata is applicable at the batch level. For posting distinct metadata for each log record, send 1 log record at a time in the batch.

Post the input JSON to the following URL:

```
http://<UNITY_HOST_NAME>:<UNITY_PORT>/Unity/DataCollector
```

where `<UNITY_HOST_NAME>` is the machine on which you installed IBM Operations Analytics - Log Analysis and `<UNITY_PORT>` is the port on which it is running. The default port is 9988. The client (Java or Script) sending data into IBM Operations Analytics - Log Analysis needs to authenticate by using the form-based mechanism that is implemented in IBM Operations Analytics - Log Analysis before the Data Collector API is invoked. Refer to the authentication and security design document for details.

Output JSON

The output that is sent by the Generic Receiver after indexing logs contains the count and detailed information on the failure cases in a JSON Array. The details include the actual logRecord, specific error message, and any exception. The basic structure of an Output JSON file is:

```
{
"batchSize" : ,   // (int)
"numFailures" : ,  // (int)
"failures" :    // (JSONArray)
  [
  {
   "logRecord" : ,  // (JSONObject)
   "errorMessage": ,  // (String)
   "exception" : ,  // (JSONArray)
  },
  .
  .
  .
  {
  }
 ]
}
```

Serviceability

As you send data into IBM Operations Analytics - Log Analysis, you might encounter errors that occur before the incoming batch gets processed or errors that occur during processing of batch and indexing log records.

If errors occur before the incoming batch gets processed, the Generic receiver returns an error code and message. To correct the problem, process the error code, make any required changes, and resend the data.

Possible causes for error code 400 (HttpServletResponse.SC_BAD_REQUEST) include:
- Invalid input JSON
- Input batch size is greater than what is supported (500000 bytes)
- No data source is configured from the Admin UI for the host name and log path combination that is sent in the input JSON
- The input type (LOGS) specified in the batch does not match the value that is specified in the logsource that is configured from the Admin UI

Possible causes for error code 500 (HttpServletResponse.SC_INTERNAL_SERVER_ERROR) include:
- An exception that is encountered in any of the steps of the ingestion pipeline (for example, during splitting of a batch).
- An internal IBM Operations Analytics - Log Analysis database-related error.
- Any other exception in IBM Operations Analytics - Log Analysis.

If errors occur during processing of batch and indexing log records, the output JSON provides details of indexing failure. To correct the problem, process the error code, make any required changes, and resend only the affected log records. Sending the same log record twice to IBM Operations Analytics - Log Analysis results in duplicate records in the back-end index and duplicate records in the search results.

Batch loading historic log data with the Data Collector client

Use the Data Collector client to ingest data in batch mode. Use this method to review historic log data. This is the easiest method if you want to ingest large log files for historic analysis.

Before you begin

If you want to use the Data Collector client to load data from remote sources, you must configure the data collector on the remote host before you can configure the local data collector as described here. For more information, see "Configuring the Data Collector client to ingest data from remote hosts" on page 263.

About this task

If you want to load a log file that does not include time stamp information, ensure that the values for `timestamp` and `timestampFormat` are configured in `javaDatacollector.properties`. IBM Operations Analytics - Log Analysis cannot index log files without a time stamp, but if no time stamp information is found in a log file, the value that is configured in javaDatacollector.properties is used.

Procedure

To use the Data Collector client to load log file information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. At the command line, navigate to the <HOME>/utilities/datacollector-client directory.
3. Update the configuration file that is used by the Data Collector client, javaDatacollector.properties. Set the following properties, as appropriate:

   logFile
   The full path of the file you want to ingest.

   servletURL
   The URL of the Data Collector service.

   userid The user ID for the Data Collector service.

   password
   The password for the Data Collector service.

   datasource
   The datasource that you want to use to load data.

   timestamp
   The time stamp to use if a time stamp is not found in the log file.

   batchsize
   The number of BYTES of logs that are sent in one batch. The default value is 500,000.

   keystore
   The full path to the keystore file.

   inputType
   The valid input types are: LOGS, CONFIGFILES, SUPPORTDOCS. The default value is LOGS.

   flush flag
   If the default true is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to false, no flush signal is sent when the end of file is reached.

   The following sample javaDatacollector.properties file displays the configuration for loading the SystemOut.log log file.

```
#Full path of the file you want to read and upload to Unity
logFile = SystemOut.log
#The URL of the REST service. Update the host/port information if required
servletURL = https://hostname:9987/Unity/DataCollector
#The user ID to use to access the unity rest service
userid=unityuser
#The password to use to access the unity rest service
password=password
datasource=Systemout
#Time stamp to use if your content can not find a time stamp in log record.
The same time stamp would be used for all records
timestamp = 01/16/2013 17:27:23:964 GMT+05:30
#The number of BYTES of logs sent in one batch to Unity
batchsize = 500000
#The full path to the keystore file
keystore = /home/unity/IBM/LogAnalysisTest/wlp/usr/servers/Unity/
```

```
keystore/unity.ks
#input data type - LOGS, CONFIGFILES, SUPPORTDOCS
inputType = LOGS
#flush flag:
#true : (default) if the client should send a flush signal to the Generic
 Receiver for the last batch of this file
#false : if no flush signal to be sent upon reaching eod-of-file
flushflag = true
#Other properties (name/value pairs, e.g. middleware = WAS) that you want
 to add to all json records
#These properties need to be appropriately added to the index configuration
```

4. Ensure that the Data Collector client JAR file, `datacollector-client.jar`, has execute permissions.
5. Use the following command to run the Data Collector client with the correct inputs:

   ```
   <HOME>/ibm-java/bin/java
   -jar datacollector-client.jar
   ```

Results

After the task completes, the log file is indexed and can be searched in the **Search** workspace.

Configuring the Data Collector client to ingest data from remote hosts

If you want to use the Data Collector client to collect data from a remote server and return it to the local machine, you must configure the data collector on the remote host.

Before you begin

You must use the instance of IBMJava Runtime Engine (JRE) 1.7 that is installed by the remote installer. Before you configure the data collector, you must use the remote installer to install at least one instance of IBM Tivoli Monitoring Log File Agent or the EIF Receiver on a remote machine. For more information, see the *Configuring data collection for scalability on multiple remote nodes* topic in the Installation Guide.

About this task

To configure the Data Collector on the remote host, copy the data collector client files from your local version of the data collector files to the remote host.

Procedure

1. Copy the `<HOME>/utilities/datacollector-client` directory and all the files that are contained in it from the local installation of IBM Operations Analytics - Log Analysis to the remote machine.
2. Add the location of the log and keystore files to the `javaDatacollector.properties` file in the directory that you copied the data collector to in the previous step. The keystore file is named `unity.ks` and it is available in the <*`Remote_install_dir`*>`/LogAnalysis/store/` directory on the remote machine. Where <*`Remote_install_dir`*> is the directory where you installed the remote instance as described in the *Prerequisites* section here.

Results

After you complete the configuration, you must complete the Data Collector configuration. For more information about how to do this, see “Batch loading historic log data with the Data Collector client” on page 262. You must ensure that the remote installation uses the IBMJava Runtime Engine (JRE) 1.7 that is installed by the remote installer. IBM Java Runtime Engine (JRE) 1.7 is stored in the `<Remote_install_dir>/LogAnalysis/ibm-java/` directory.

Data Collector properties

Before you can use the data collector to stream data or load a batch of historic data, edit the `javaDatacollector.props` file.

The `javaDatacollector.props` file is in the `<HOME>/IBM/LogAnalysis/utilitiesdatacollector-client` folder.

The `logFile`, `hostname`, `logpath`, and `keystore` parameters are required.

The `userid`, `password`, and `keystore` parameters are automatically populated with the default values that are created during the installation. If you want, you can change these but you do not need to.

Table 109. Data Collector properties

Parameter	Value
`logFile`	The full path of the file you want to load.
`servletURL`	The URL of the Data Collector service.
`userid`	The user ID for the Data Collector service.
`password`	The password for the Data Collector service.
`datasource`	The datasource that you want to use to load data.
`timestamp`	The time stamp to use if a time stamp is not found in the log file.
`batchsize`	The number of BYTES of logs sent in one batch. The default value is 500,000.
`keystore`	The full path to the keystore file.
`inputType`	The valid input type is `LOGS`.
`flush flag`	If the default `true` is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to `false` no flush signal is sent when the end-of-file is reached.

Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent

You can use the IBM Tivoli Monitoring Log File Agent to load batches of historic data for testing and other purposes.

Procedure

1. Copy the log files that you want to load to a temporary directory on the IBM Operations Analytics - Log Analysis server. For example, to upload a batch of

log files from an installation of WebSphere Application Server, you copy the `SampleSystemOut.log` file to the `/tmp/logs/` directory.
2. Create a custom data source.
3. Copy the log file to the directory that you specified in the `logpath` parameter when you created the data source.

Extending storage space available to Apache Solr

You can add more Apache Solr storage directories outside the initial IBM Operations Analytics - Log Analysis Apache Solr installation location if the disk on which Apache Solr was installed reached maximum capacity.

Before you begin

Ensure that the Apache Solr storage directories are present on all Apache Solr servers and are writable.

About this task

Switching to a new Apache Solr directory is not instantaneous. Therefore, it is to monitor the disk usage of your Apache Solr directory to ensure that extra directories are added before the current storage directory reaches maximum capacity.

Procedure

To enable the storage extension capability, complete the following steps.

1. Stop IBM Operations Analytics - Log Analysis with the following command.
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
   ```
2. Open the `unitysetup.properties` file in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
3. Add the following property to the directory
   ```
   ENABLE_SOLR_RELOCATION=true
   ```
4. Create the following properties file
   ```
   <HOME>/solrConfigs/storageConfig.properties
   ```

   For example,
   ```
   /home/unity/IBM/LogAnalysis/solrConfigs/storageConfig.properties
   ```
5. Open the `storageConfig.properties` file and add the following property to the file.
   ```
   SOLR_STORAGE_DIR=storage-path-on-solr-nodes
   ```

   For example,
   ```
   SOLR_STORAGE_DIR=/opt/scala/ext_storage
   ```
6. Restart IBM Operations Analytics - Log Analysis with the following command.
   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Results

The new IBM Operations Analytics - Log Analysis configuration file enables the specification of custom data storage locations. The new locations are written to when IBM Operations Analytics - Log Analysis crosses the default boundary of 1 day.

Changing the default boundary for creating Apache Solr collections

You can change the default boundary that is associated with extending Apache Solr storage space depending on your business needs.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file.
2. Locate and modify the value of the `COLLECTION_ASYNC_WINDOW` property from the default value of `1d` (1 day).

   Note: The minimum property size is `6h`.

   The boundary size can be specified in minutes (`m`), hours (`h`), or days (`d`).
3. Restart IBM Operations Analytics - Log Analysis with the following command.

   ```
   <HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
   ```

Changing the default password for the Data Collector and EIF Receiver

If you want, you can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics. This is optional.

Changing the default EIF Receiver or Data Collector password

You can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

About this task

After you install IBM Operations Analytics - Log Analysis, the EIF Receiver and the Data Collector are configured to use the default user name and password to connect to IBM Operations Analytics - Log Analysis. The encrypted passwords are defined in the following files:

- Data Collector client is named `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`.
- EIF Receiver is named`<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`.

IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to encrypt and decrypt passwords for your installation, in the following format:

```
password={aes}<Unique_string_of_alphanumeric_characters>
```

For example, the `javaDatacollector.properties` file uses the `unityuser` user ID to access the Data Collector server. In this example, IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to generate the following password:

```
{aes}7DB629EC03AABEC6C4484F160FB23EE8
```

The encrypted password is replicated to the configuration files for the Data Collector and the EIF Receiver.

Procedure

1. To change the default password, use the `unity_securityUtility.sh` command.

For more information about this command, see "unity_securityUtility.sh command" on page 205.

2. Update the configuration files for the Data Collector or the EIF Receiver.
3. Optional: If you want to change the password on remote instances of the EIF Receiver, complete the previous steps and copy the unity.conf file from the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/ directory on the local machine to the *<remote_deployment_location>*/LogAnalysis/DataForwarders/EIFReceivers/*<eif_inst_#>*/config/unity.conf directory on the remote machine. Where *<remote_deployment_location>* is the directory on the remote machine where you deployed the EIF instance. *<eif_inst_#>* is the folder that is used for the specific remote EIF instance.

Example

For example, you want to change the default password for the default user that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis to myNewPassword. Complete the following steps:

1. Go to the IBM/LogAnalysis/utilities directory.
2. Run the unity_securityUtility.sh command as follows:
```
 [utilities]$ ./unity_securityUtility.sh encode myNewPassword
 Using keystore file unity.ks
 <HOME>/IBM/LogAnalysis/utilities/../wlp/usr/servers/Unity/
keystore/unity.ks
 {aes}E6FF5235A9787013DD2725D302F7D08
```
3. Copy the AES encrypted password to the relevant configuration files, for example copy it to the Data Collector file. You must copy the complete, encrypted string from the command output, including the {aes} prefix. For example:
```
{aes}E6FF5235A9787013DD2725D302F7D088
```

unity_securityUtility.sh command

You can use the unity_securityUtility.sh command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Syntax

The unity_securityUtility.sh command is in the <HOME>/IBM/LogAnalysis/utilities directory and it has the following syntax:

```
unity_securityUtility.sh encode [textToEncode] [unity.ks]
```

Parameters

The unity_securityUtility.sh command has the following parameters:

encode

The encode action returns an AES encrypted version of the text that you enter as the text to encrypt.

[*textToEncode*]

Use the [*textToEncode*] parameter to enter the password that you want to encrypt. If you do not specify a password for this parameter, IBM Operations Analytics - Log Analysis prompts you for one.

[unity.ks]

The unity.ks file is the default keystore that is generated automatically during installation. It controls how the password is encrypted and decrypted.

The unity.ks file is used to encrypt and decrypt passwords for the following features:

- Java data collector client in the <HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties file.
- EIF Receiver in the <HOME>/IBM/LogAnalysis/utilities/UnityEIFReceiver/config/unity.conf file.

For an example of how to use this command, see “Changing the default EIF Receiver or Data Collector password” on page 266.

Installing logstash

Installing logstash on a remote node extends IBM Operations Analytics - Log Analysis functions so it can ingest and perform metadata searches against log data that is acquired by logstash.

logstash 1.4.2 is bundled and installed with IBM Operations Analytics - Log Analysis. You can install logstash on a local host but to improve system performance, install logstash on a remote node.

This document describes the version of logstash that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of logstash might have been published after this version of IBM Operations Analytics - Log Analysis. To download the most up-to-date logstash versions and updated documentation, see https://www.elastic.co/downloads/logstash.

logstash is an open source tool for managing events and logs. It can be used to collect logs, parse them, and send them to another tool such as IBM Operations Analytics - Log Analysis to store them for later use.

The logstash agent is an event pipeline that consists of three parts:

1. Inputs
2. Filters
3. Outputs

Inputs generate events. Filters modify events. Outputs send the event somewhere. For example, events can be sent to storage for future display or search, or to the IBM Operations Analytics - Log Analysis framework. Events can have a type, which is used to trigger certain filters. Tags can be used to specify an order for event processing as well as event routing to specific filters and outputs.

logstash can be used as a "pre-processor" to analyze sources and provide a semi-structured or structured feed to IBM Operations Analytics - Log Analysis for the purposes of searching and potential usage within custom analytics applications.

For more information on logstash events, see the section *the life of an event in logstash* at https://www.elastic.co/guide/en/logstash/current/index.html.

Dependencies

Supported version of logstash and its dependencies.

Supported logstash version

The supported version of logstash is 1.4.2 .

DSV Toolkit requirement

DSV Toolkit v1.1.0.1 or higher for generating IBM Operations Analytics - Log Analysis Insight Packs. The Insight Packs are used to index log records that have been annotated using logstash. You only require the DSV toolkit if you want to use logstash to perform ingestion, splitting, annotating or for when the data being read by logstash is in DSV format. For more information on this user scenario, see "Configuring logstash for rapid annotation and pre-indexing processing" on page 273.

Generic Annotation Insight Pack

Generic Annotation v1.1.0, or v1.1.1 (refresh 1) is recommended for the normalized timestamp splitter function, which recognizes a variety of timestamps.

Installing logstash on a remote node

You can install logstash on a remote node to improve system performance.

Before you begin

Ensure that the SSH user has the correct permissions for installation. For more information on SSH configuration, see "Secure Shell (ssh) configuration for remote logstash" on page 270 in the *Loading and streaming data guide*.

About this task

logstash is processor and system resource intensive. logstash can be installed on the local host but to improve system performance, install logstash on a remote node.

Procedure

1. To install logstash, run the following command:
   ```
   <HOME>/IBM/LogAnalysis/remote_install_tool/install.sh
   ```
2. The installation script installs logstash, and provides options to install the EIF receivers and log file Agent. To select each option, including logstash, select `y` or `Y`.
3. Provide the path to the installation location on the remote host.

Results

logstash is installed in the `<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/` directory. To confirm the installation, logon to the remote node as the configured SSH user and go to the installation location.

Example

The following are example deployments:

logstash example:

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/
```

Output plug-in configuration path:

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/logstash-scala/logstash/
config/logstash-scala.conf
```

Output plug-in jar directory

```
<install-dir>/LogAnalysis/Logstash/logstash-1.4.2/logstash-scala/logstash/outputs
```

Secure Shell (ssh) configuration for remote logstash

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

The `ssh_config.properties` file is in the <HOME>/IBM/LogAnalysis/remote_install_tool/config directory. Configure the parameter values as outlined in Table 1.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password-based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file-based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

To set up command-line authentication, rename the ssh-config properties file or move the properties file to a new location. By default the configurations are selected from the properties file. If the file is unavailable, the user is prompted for command-line input.

Table 110. ssh_config parameters

Parameter	Value
`REMOTE_HOST=`	*<REMOTE SERVER IP/FQ HOSTNAME>*
`PORT=`	*<SSH PORT>* THE DEFAULT VALUE IS 22
`USER=`	*<SSH_USER>*
`PASSWORD=`	*<SSH PASSWORD>*

logstash configuration

logstash can be configured as a log file agent to ingest logs from a number of different sources.

About this task

There are two established use cases for using logstash with IBM Operations Analytics - Log Analysis, these are:

- Configuring logstash as an alternative to ITM LFA
- Configuring logstash for rapid annotation and pre-indexing processing

Both use cases are described in this section.

Configuring logstash as an alternative to ITM LFA

logstash can be used as a log file agent to ingest logs from a number of different sources. It can also support integration with numerous alternative log file agents such as Lumberjack, Minuswell, Beaver, and Syslog.

About this task

Log records are written to the IBM Operations Analytics - Log Analysis that then sends the message to the IBM Operations Analytics - Log Analysis server for annotating and indexing.

Procedure

1. Update the logstash sample configuration file, `logstash/config/logstash-scala.conf`, with your configuration information.
   a. Define the input in the logstash configuration file.

      For example:

      ```
      input {
        file {
         type => "http"
         path => ["/tmp/myhttp.log"]
        }
       }
      ```

      Note: For Windows, the logstash file plug-in requires a drive letter specification for the path, for example:

      ```
      path => ["c:/tmp/myhttp.log"]
      ```

   b. Modify the logstash configuration file to add the `scala` output plug-in.

      The `scala` output plug-in buffers and sends the logstash event to the IBM Operations Analytics - Log Analysis server by using the Log Analysis server ingestion REST API. The logstash configuration file can contain one or more `scala` output plug-ins. The output plug-ins can be configured to write to different Log Analysis servers or to the same Log Analysis server with a different set of configurations.

      Every event that is sent to the `scala` output plug-in must contain at least the `host` and `path` fields. The values of these fields are used by the `scala` output plug-in to determine the target data source for the event. Any event that does not contain either of these fields is dropped by the output plug-in.

      The following are the default parameters, with sample values, for the IBM Operations Analytics - Log Analysis `scala` output plug-in:

      ```
      output {
        scala {
          scala_url => "https://<la_server>:<port>/Unity/DataCollector"
          scala_user => "<LA_user>"
          scala_password => "<encrypted_pwd>"
          scala_keystore_path => "<install-dir>/LogAnalysis/store/unity.ks"
          batch_size => 500000
          idle_flush_time => 5
          sequential_flush => true
          num_concurrent_writers => 20
          use_structured_api => false
          disk_cache_path => "<install-dir>/LogAnalysis/Logstash/cache-dir"
          scala_fields =>
            {
             "host1@path1,host2@path2"
                => "event_field11,event_field12,...,event_field1N"
             "host3@path3"
                => "event_field21,event_field22,...,event_field2N"
      ```

```
    }
   date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
   log_file => "<install-dir>/LogAnalysis/Logstash/logs/scala_logstash.log"
   log_level => "info"
  }
```

Where:

- **scala_url** is the REST endpoint for the Log Analysis ingestion REST API.
- **scala_user** is the Log Analysis user name.
- **scala_password** is the Log Analysis user password.
- **scala_keystore_path** is the path to the Log Analysis keystore on the file system.
- **batch_size** is the maximum number of bytes that can be buffered for a data source before transmitting to the Log Analysis server. The default is *500000* bytes.

  Note: Significantly decreasing the batch size impacts on throughput. Increasing the batch size requires more heap memory.
- **idle_flush_time** is the maximum time between successive data transmissions for a data source.
- **sequential_flush** defines whether batches for each data source are sent sequentially. It is set to *true* to send the batches sequentially.

  Note: Sequential sending is required when the input contains multi-line records that are combined in an Insight Pack in the Log Analysis server.
- **num_concurrent_writers** is the number of threads that concurrently transmit batches of data to the Log Analysis server.
- **use_structured_api** determines whether data is transmitted to the Log Analysis server in the JSON format. It is set to *true* to transmit data in the JSON format.

  Note: The target Log Analysis data source must be associated with a source type that uses the Log Analysis structured API.
- **disk_cache_path** is the path on the file system that temporarily buffers data. The scala output plug-in writes data to this path before transmission. The available disk space under the path must be large enough to store bursts of input data that is not immediately handled by the Log Analysis server.
- **scala_fields** is the map that specifies the names of fields that must be retrieved from the incoming logstash event and transmitted to the Log Analysis server. The keys for the map are a comma-separated list of host and path names that correspond to a Log Analysis data source.

  The scala plug-in extracts the host and path fields from each event before consulting the **scala_fields** map for a host and path combination entry. If there is an entry with field names, the scala plug-in extracts the corresponding field values from the event. The values are transmitted to the Log Analysis server. If the host and path entries are not in the **scala_fields** map, the scala plug-in extracts the contents of the message field from the event and transmits it to the Log Analysis server.
- **date_format_string** is the string value that all fields are transformed to before transmission to the Log Analysis server. The scala plug-in uses the **date_format_string** parameter to convert date values to the appropriate string value.

- **log_file** is the file that is used for logging information from the scala output plug-in.
- **log_level** is the level of logging information. The supported levels are fatal, error, warn, info, and debug.

2. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

   Ensure that the **File Path** matches the path that is specified in the logstash configuration file, logstash-scala.conf.

   Ensure that the **Type** matches the type of log file that is being ingested, for example **DB2Diag**.

   For example, if you specified /tmp/myhttp.log as an input file, then create a custom data source with path set to /tmp/myhttp.log.

What to do next

Start logstash as described in Starting logstash

Configuring logstash for rapid annotation and pre-indexing processing

logstash can be used to split log records and do basic annotation. For log types not currently supported by IBM Operations Analytics - Log Analysis, this is an alternate approach to writing AQL to annotate log files.

About this task

logstash includes a broad list of filtering, manipulation, and processing capabilities, for example, the grok filter can be used to parse text into structured data. It allows you to match text without the need to master regular expressions. There are approximately 120 grok patterns shipped by default, though you can add more. It also includes patterns for known log file formats, such as Apache's combined access log format.

In this scenario, logstash is basically used as the splitter/annotator of the log file by leveraging the grok filter. The scala_custom_eif output plugin sends a single log record to the IBM Operations Analytics - Log Analysis EIF Receiver, with the annotations in a delimiter separated value (DSV) format. Then, using the DSV Toolkit, the user must create and install an insight pack that matches the DSV format so that IBM Operations Analytics - Log Analysis can index the annotations. Please follow these steps:

Procedure

1. Update the logstash sample configuration file, logstash/config/logstash-scala.conf, with your configuration information.
   a. Define the input in the logstash configuration file.

      For example:

```
input {
  file {
   type => "apache"
   path => ["/tmp/logs/myapache.log"]
  }
 }
```

      Note: For Windows, the logstash file plugin requires a drive letter specification for the path, for example:

```
path => ["c:/tmp/myapache.log"]
```

b. Modify the logstash configuration file to add the `scala_custom_eif` output plugin.

c. Add a filter or filters to the logstash configuration file to identify the pattern of the log file format. This also creates the annotations. To trigger the filter, the type must match the input type.

   For example:

```
filter {
    if [type] == "http" {
        grok {
            match => ["message", "%{IP:client} %{WORD:method}
%{URIPATHPARAM:request}  %{NUMBER:bytes} %{NUMBER:duration}"]
                      }
    }
}
```

   In this example, the fields client, method, request, bytes, and duration are annotated by the pattern. However, only the fields client, method and request are sent to IBM Operations Analytics - Log Analysis. Thus, those are the only three annotations that can be included in the index configuration. The output module sends the event text in DSV format as:

```
"client", "method", "request"
```

   The user can also use one of the many predefined grok log format pattern such as:

```
filter {
    if [type] == "apache" {
        grok {
            match     => ["message", "%{COMBINEDAPACHELOG}"]
        }
    }
}
```

2. Create an IBM Operations Analytics - Log Analysis DSV-generated Insight Pack in order to index the annotated data in IBM Operations Analytics - Log Analysis.

   The `lsartifact/dsvProperties` directory contains a sample property file that can be used to generate an Insight Pack that ingests delimiter separated log records that are already formatted for Apache combined access log files. Use the DSV toolkit, which is available at `<HOME>/IBM/LogAnalysis/unity_content/tools`, to generate an Insight Pack from the DSV properties file. This means the user must configure the logstash configuration file, `/lstoolkit/logstash/config/logstash-scala.conf`, with the appropriate grok filter to enable the IBM Operations Analytics - Log Analysis output plugin to generate the comma delimited logs. For example, uncomment the apache grok filter in the `logstash-scala.conf` file and generate an Insight Pack using `ApacheDSV.properties` with the DSV tooling script. The `scala` plugin will generate a comma delimited event based on the grok filter that can be ingested (annotated and split) by the generated Insight Pack.

   Note: The path to `logstash-scala.conf` is dependent on where you copied the `lstoolkit` directory on the logstash server (see step 3 of Installing the logstash Integration Toolkit).

3. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

   Ensure that the **File Path** matches the path that is specified in the logstash configuration file, `logstash-scala.conf`.

   Ensure that the **Type** matches the type of log file that is being ingested, for example **DB2Diag**.

For example, if you specified /tmp/myhttp.log as an input file, then create a custom data source with path set to /tmp/myhttp.log.

What to do next

Start logstash. For more information on starting logstash, see “logstash operations” on page 276 in the *Installing logstash* section of the *Loading and streaming data* guide.

Example - Annotating Combined Apache log files:

Using logstash to annotate Apache log files.

Procedure

1. Edit your logstash configuration file. A sample is provided in logstash-scala.conf.
   a. In the input section, specify the Apache log file to be monitored.
      ```
      input {
        file {
         type => "apache"
         path => ["/tmp/apache.log"]
        }
       }
      ```
   b. Add the logstash grok filter with the predefined COMBINEDAPACHELOG pattern to annotate the Apache log files.

      For example:
      ```
      filter {
       if [type] == "apache" {
        grok {
           match => ["message", "%{COMBINEDAPACHELOG}"]
        }
       }
      }
      ```
      The COMBINEDAPACHELOG pattern is defined as:
      ```
      COMBINEDAPACHELOG %{IPORHOST:clientip} %{USER:ident} %{USER:auth}
      \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}
      (?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response}
      (?:%{NUMBER:bytes}|-) %{QS:referrer} %{QS:agent}
      ```
      For more information about Apache log files, see http://httpd.apache.org/docs/2.4/logs.html.

      The logstash event contains annotations for clientip, ident, auth, timestamp, verb, request, httpversion, rawrequest, response, bytes, referrer, and agent.
   c. In the output section of the configuration file, specify the IBM Operations Analytics - Log Analysis output plug-in.
2. The logstash Integration Toolkit provides a properties file, lsartifact/dsvProperties/ApacheDSV.properties, which can be used with the DSV Toolkit to create an Apache Insight Pack. Edit this properties file to configure information about your IBM Operations Analytics - Log Analysis server:
   ```
   [SCALA_server]
    username: unityadmin
    password: unityadmin
    scalaHome: $HOME/IBM/LogAnalysis
   ```
3. Use the dsvGen.py script that is provided with the DSV Toolkit to generate and deploy the Apache Insight Pack:
   ```
   python dsvGen.py <path>/ApacheDSV.properties -d
   ```

4. In the IBM Operations Analytics - Log Analysis Administrative Settings UI, create a data source, which has the Apache source type that is created by the DSV toolkit in step 4, in your logstash configuration file.
5. Start logstash with the configuration file, and start ingesting Apache log files.

logstash operations

You can use the `logstash-util` script to start, stop, restart, or provide the status of logstash.

About this task

You can use the `logstash-util` script for logstash process lifecycle management.

Procedure

1. To start, stop, restart, or provide the status of logstash, run the following command:

   ```
   <install-dir>/LogAnalysis/utilities/logstash-util.sh start| stop| restart| status
   ```

   where *<install-dir>* is the name of the logstash installation location.
2. To confirm that logstash is running, run the `logstash-util` script and use the `status` option. The `status` option also displays the logstash process identifier.

logstash best practices

Best practices for logstash based on information from their user community.

For performance reasons it is recommend that logstash be installed on a different server than IBM Operations Analytics - Log Analysis. logstash is processor, memory, and disk intensive if the annotation and indexing functions are utilized.

Users who have memory constraints do not use logstash as a forwarding agent. They do not install logstash on the end client servers. They use other applications such as rsyslog to forward logs to a central server with logstash. See https://support.shotgunsoftware.com/entries/23163863-Installing-logstash-Central-Server for an example configuration.

Users with logstash at the end client who are concerned about performance have used applications such as Redis to forward logs to a central server with logstash. See the following for configuration of Redis http://www.linux-magazine.com/Online/Features/Consolidating-Logs-with-logstash .

To fine tune logstash, especially the startup time, users can tweak Java's minimum and maximum heap size with the -Xms and -Xmx flags. The -Xms parameter is the initial Java memory heap size when the JVM is started, and the -Xmx parameter is the maximum heap size.

References

Links for more information on the logstash application.

logstash website:
http://logstash.net

Getting Started with logstash Guide:
http://logstash.net/docs/1.4.2/tutorials/getting-started-with-logstash

logstash Download:
http://logstash.net (Click download button)

The logstash Book:
http://www.logstashbook.com/

IBM Operations Analytics - Log Analysis wiki:
http://www.ibm.com/developerworks/servicemanagement/ioa/log/downloads.html

IBM Operations Analytics - Log Analysis wiki: Logstash Toolkit Resources:
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Logstash%20Toolkit%20Resources

Known issues

Known issues when using logstash with IBM Operations Analytics - Log Analysis.

There are a number of known issues and their workarounds described in this section. To get the latest information on any issues or workarounds, please consult the IBM Operations Analytics - Log Analysis wiki:https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Welcome

Could not load FFI Provider

Starting logstash fails with the Ruby exception “Could not load FFI Provider”.

Symptoms

The Ruby exception “Could not load FFI Provider”.

Causes

The most common cause of this error is that /tmp is mounted with the **noexec** flag.

Resolving the problem

You can resolve this either by:

- Making /tmp mounted without the **noexec** flag
- Edit the startlogstash-scala script and amend the start command as follows:

```
LSCMD="$MYJAVA -jar -Djava.io.tmpdir=</some/tmp/dir> $LSJAR agent
--pluginpath $PLUGPATH -f $CONF"
```

  Where </some/tmp/dir> is a temporary directory.

Duplication of log records on the SCALA server

On occasion, when the logstash agent is re-started, and the log file being monitored is updated (for example, via a streaming log), logstash will ingest the entire file again rather than restarting from where it stopped monitoring.

Symptoms

The problem results in a duplication of log records on the SCALA server.

Causes

Several problems have been reported on the logstash forum (https://logstash.jira.com/secure/Dashboard.jspa) that its sincedb pointer (which tracks the last monitored position in the log file) sometimes is not updated correctly. In addition, using control-C to terminate the logstash agent does not always kill logstash. The result is a "phantom" logstash agent that is still monitoring log files. This can also result in duplicate log records.

Resolving the problem

1. A workaround to avoid duplicate log records after restarting logstash is to set the **sincedb_path** parameter in the file plugin to /dev/null, thereby telling logstash to ignore tracking the last-monitored file position, and always start monitoring from the end of the file. However, this will result in logstash ignoring any updates to the log file while the logstash agent is down. For example, in logstash-scala.conf, update:

   ```
   input {
       file {
           type => "apache"
           path => ["/tmp/logs/myapache.log"]
           sincedb_path => "/dev/null"
       }
   }
   ```

   Before re-starting logstash after making these configuration changes, you may also want to clean up any sincedb databases that were already created. By default, the sincedb database is stored in the directory $HOME, and have filenames starting with ".sincedb_".
2. When terminating the logstash agent using control-C, verify that the logstash java process was actually terminated. You can use the following command to see if logstash is still running:

   ```
   ps -ef | grep logstash
   ```

Logs do not appear in the Search UI

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Symptoms

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Causes

Log records ingested by logstash are forwarded to the IBM Operations Analytics - Log Analysis server for splitting and annotating, and indexing. If the IBM Operations Analytics - Log Analysis server goes down during this process, it is possible to lose some log records.

Notices

This information was developed for products and services that are offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered 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 www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

Privacy Policy Considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name and password for purposes of session management, authentication, enhanced user usability, and single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered 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.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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

Linux is a 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.

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

Other product and service names might be trademarks of IBM or other companies.

Product Number:

Printed in USA