

Oracle® Cloud

Using Oracle Log Analytics



E60700-64
June 2022



Oracle Cloud Using Oracle Log Analytics,

E60700-64

Copyright © 2015, 2022, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	x
Documentation Accessibility	x
Related Resources	x
Conventions	x

Part I Get Started with Oracle Log Analytics

1 Configure Oracle Log Analytics

Install the Cloud Agent and Enable Oracle Log Analytics	1-1
Perform Prerequisite Configuration Tasks	1-3
Add Entities to Oracle Log Analytics	1-6
Perform Oracle Log Analytics Specific Configuration Tasks	1-7
Set Up Syslog Monitoring	1-8
View Syslog Data	1-8
Set Up Database Instance Monitoring	1-9
Create the Database Log Source	1-9
Provide the Database Entity Credentials	1-10
Set Up Autonomous Database Audit Log Collection	1-12

2 Upload Logs to Oracle Log Analytics on Demand

Upload Log Files Using REST API	2-1
Upload Log Files Using ODU Client	2-2
Prepare to Use the ODU Client	2-2
Download the ODU Client	2-3
General Syntax of ODU Client Commands	2-3
Upload Log Files	2-4
Set Up the Properties File	2-7
Check Submission Status of an Upload	2-8
Retry an Upload	2-10

Upload Log Files Using ODU Wizard	2-11
Verify an Upload	2-14
Delete Uploaded Log Files	2-14

3 Ingest Logs from OCI Object Storage Buckets

View the Bucket Configuration	3-4
Unregister the Buckets for Log Collection	3-5

4 Use Fluentd for Log Collection

Install the Output Plug-In	4-1
Edit Fluentd Configuration File	4-2
Configure the Format of the Incoming Log Events	4-5

Part II Administer Oracle Log Analytics

5 Create a Parser

Guided Creation of the Regex Type Parser	5-2
Manual Creation of the Regex Type Parser	5-4
Create JSON Type Parser	5-6
Create XML Type Parser	5-7
Preprocess Log Events	5-8
Master Detail Function	5-9
Find Replace Function	5-13
Time Offset Function	5-15

6 Configure New Log Sources

Create a Log Source	6-1
Use Extended Fields in Log Sources	6-4
Use Data Filters in Log Sources	6-7
Use Labels in Log Sources	6-10
Create a Label	6-11
Create a Field	6-14
Create Lookups	6-15
Create a CSV Lookup	6-16
Create a Dictionary Lookup	6-16
Use the Generic Parser	6-20
Configure Field Enrichment Options	6-21

Geolocation Lookup	6-21
Use a Lookup in the Log Source	6-22

7 Administer: Other Actions

Set Up an Access Policy for a User	7-1
Create an Access Policy	7-1
Assign an Access Policy to a User	7-2
Manage Annotations	7-2
Add an Annotation to Log Records	7-3
Edit an Annotation	7-3
Manage Existing Log Sources	7-4
Edit Log Source	7-4
Create a Log Source Based on an Existing One	7-5
Work with Entity Associations	7-5
Configure New Entity Associations	7-5
Manage Existing Entity Associations	7-6
Associate Log Sources to Existing Entities	7-6
Associate Entities to Existing Log Sources	7-7
View Collection Warnings	7-7
View Warnings Summary	7-7
View Entities with Collection Warnings	7-7
Export the Content from Oracle Log Analytics	7-8
Export the Log Parsers	7-8
Export the Log Sources	7-8
Purge Log Data	7-9
Archive Log Data	7-10
Create Archive Policy	7-10
Recall Archived Logs	7-12
View Archive, Recall, and Purge Activity	7-13
Create Credential for OCI Authentication	7-14
Prerequisites for Creating Credentials	7-14
Create Credentials using UI	7-15
Create Credentials using REST API	7-16

Part III Use Oracle Log Analytics

8 Visualize Data Using Charts and Controls

Select the Visualization Type	8-3
Compare and Contrast the Data Set Using One or Two Parameters	8-3

Summarize the Data Set Using Key Parameters	8-6
Group and Drill Down to the Specific Data Set	8-8
Analyze the Data Set Using Multiple Key Parameters	8-9
Perform Advanced Analysis of the Data Set	8-10
Log Scales Visualization	8-12
View the Field Summary	8-13
Configure the Display of the Field Summary	8-16
View an Entity Card	8-17
Bar Charts Visualization	8-17
Clusters Visualization	8-19
Cluster the Log Data Using SQL Fields	8-22
Use Cluster Compare Utility	8-23
Use Dictionary Lookup in Cluster	8-25
Line Charts Visualization	8-26
Maps Visualization	8-28
Summary Tables	8-29
Word Cloud Visualization	8-30
Link Visualization	8-31
Use Dictionary Lookup in Link	8-38
Semantic Clustering Using Natural Language Processing	8-39
Generate Link Alerts	8-42
Use the Getting Started Panel	8-43
Analyze Chart Options	8-44
Additional Information in Analyze Chart	8-45
Histogram Chart Options	8-45
Compare Link Metrics Across Time	8-47
Combine and Stack Histogram Charts	8-48
Groups Table	8-49
Add URLs to Link Table	8-49
Features for Bubble Charts in Link Analysis	8-50
Change the Title of the Bubble Chart	8-50
Control the Color of the Bubbles in the Chart	8-50
Features for Fields in Link Analysis	8-51
Add More than Two Fields	8-51
Rename the Fields by Editing the Query	8-52
Add More Fields for Analysis Using Size and Color	8-53
Instant Analysis of Multiple Fields Using the Link Analyzer Chart	8-54
Mark a Field Type as Percentage or Microsecond	8-54
Features for Groups in Link Analysis	8-55
Change the Group Alias	8-55
Join Multiple Groups Using the Map Command	8-56

Create Sub-Groups Using the Createview Command	8-56
Search and Highlight Link Groups	8-57
Link by Cluster	8-58
Generate Alerts for Cluster Utilities	8-60

9 Filter and Search Through the Log Data

Typical Workflow for Troubleshooting Problems	9-1
Search Logs by Entities	9-2
Use the Filter-Out Option	9-2
Search Logs Using Keywords and Phrases	9-3
List the Recent Searches	9-4
Use the Autosuggest Feature	9-4
Filter Logs by Pinned Attributes and Fields	9-4
Filter Logs by Source Attributes	9-5
Filter Logs by Labels	9-6
Filter Logs by Data Uploaded on Demand	9-6
Filter Logs by Fields in Log Messages	9-7
Rename a Field	9-8
Filter Logs by Field Range	9-8
Filter Logs by Hash Mask	9-9
Filter Logs by Annotations	9-9

10 Save and Share Log Searches

Save a Search and Add It to a Dashboard	10-1
Create Alerts for Saved Searches	10-2
Create a Saved Search from an Existing One	10-3
Create Alerts for Existing Saved Searches	10-4
View Saved Search Anomaly Alerts and Baseline Charts	10-4
Associate Saved Search Alerts with Entities	10-5
Export the Search Results	10-5

11 Create An Alert Rule

View and Edit Alert Rules	11-3
Generate Inline Alerts	11-4
View the Entity Details for an Alert	11-4

12 Transform Logs into Operational Insight

Typical Workflow for Developing Operational Insights	12-1
Use Sample Log Data	12-1
Compare the Log Records	12-2
Use Out-of-the-Box Widgets	12-2
Create Custom Dashboards	12-3
Generate Log Metrics	12-4

Part IV Typical Use Cases

A Out-of-the-Box Log Sources

B Understand the Search Commands

C Entity Types Modeled in Oracle Log Analytics

D SQL Query Guidelines

E List of Non-Facetable Fields

F Commonly Used Oracle Log Analytics Entities

G Additional Entities in Oracle Log Analytics

H Download and Customize Oracle Log Analytics JSONs

I Write Performant Extended Field Extraction Expression

J Write Performant Regular Expressions

Sample Parse Expressions	J-3
--------------------------	-----

K Manually Specify Time Zone and Character Encoding for Files

L Add Entity by Creating a JSON File

Preface

Oracle Log Analytics provides a platform for searching and analyzing logs that're collected from entities to troubleshoot the issues encountered in them. You can also identify potential issues and plan to mitigate the errors.

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Resources](#)
- [Conventions](#)

Audience

Using Oracle Log Analytics is intended for users who want to analyze and monitor log data across the enterprise from sources such as system logs, network access logs, database logs, error logs, OS operations logs, application server logs, and many more.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Resources

For more information, see these Oracle resources:

- [Using Oracle Application Performance Monitoring](#)
- [Using Oracle IT Analytics](#)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Get Started with Oracle Log Analytics

Topics:

- [About Oracle Log Analytics](#)
- [About Oracle Log Analytics Roles and Users](#)
- [Before You Begin with Oracle Log Analytics](#)
- [Configure Oracle Log Analytics](#)
- [Upload Logs to Oracle Log Analytics on Demand](#)
- [Ingest Logs from OCI Object Storage Buckets](#)
- [Use Fluentd for Log Collection](#)

About Oracle Log Analytics

Oracle Log Analytics is a unified, integrated cloud solution that lets you monitor, aggregate, index, analyze, search, explore, and correlate all log data from your applications and system infrastructure.

Using Oracle Log Analytics, you can:

- Explore logs specific to the application that's experiencing a problem
- Analyze and explore log data efficiently
- Gain business and IT operational insight from log data
- Rapidly obtain the key values and collate them from the logs

Log events are loaded, analyzed, field-enriched, and indexed in Oracle Log Analytics. These operations can be performed either by using out-of-box parsers, by using user-defined labels, or by defining extended fields. Depending on the amount of field-enrichment done for each log event, the index size (the unit of measure for metering and billing) in Oracle Log Analytics may vary between 1.2 to 1.8 times the original log volume. While Oracle provides users with guidance on the amount of overhead that these activities will create in the indexes, the actual amount will depend on the specific operations defined or performed by the users.

 **Note:**

Oracle Log Analytics provides National Language Support (NLS) for ingesting logs that contain single-byte and double-byte character sets. NLS is available for the following nine languages:

- French
- German
- Italian
- Spanish
- Brazilian Portuguese
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

Watch this short video to get a brief overview of searching and analyzing logs.



About Oracle Log Analytics Roles and Users

Once you are an Oracle Cloud customer and you create an Oracle Management Cloud instance, the following user roles are provisioned:

Role	Tasks
Oracle Log Analytics Administrator OR Oracle Management Cloud Administrator	<ul style="list-style-type: none">• Set up Oracle Log Analytics.• Monitor targets.• Create and administer new log sources.• Create and administer new log parsers.
Oracle Log Analytics User OR Oracle Management Cloud User	<ul style="list-style-type: none">• Select targets, groups, or systems to explore.• Search and analyze logs.• Save and share log searches.• Build custom dashboards.

The Administrator and User roles created will depend on the version of the license you've purchased. If you're an existing customer **and** you've purchased the standalone Oracle Log Analytics service, then the Oracle Log Analytics Administrator and Oracle Log Analytics User roles are created.

For more information about the tasks that the users assigned with the above roles can perform, see Add Users and Assign Roles in *Getting Started with Oracle Management Cloud*.

Before You Begin with Oracle Log Analytics

Here are some of the common terms and basic concepts for Oracle Log Analytics.

Term	Definition
Agents	Oracle Management Cloud agents collect configuration, performance, availability, and log data from monitored entities and make this information available in Oracle Management Cloud.
Alerts	Information generated in response to an availability issue or when a metric crosses its thresholds. Conditions for generating alerts are defined in Alert Rules. Alerts sent to administrators by using various channels, such as, email and SMS are known as notifications .
Alert Rules	A set of rules that define the conditions under which alerts are generated and notifications sent when an alert is raised. Alert conditions consist of a metric, a comparison operator, and thresholds against which metric values are evaluated.
Associations	Associations (association instances) define a relationship between two managed entities. The association type that you define, either via the user interface or based on a written document Oracle provides, determines how data is correlated and visualized in Oracle Management Cloud. In many cases, associations are defined automatically by Oracle Management Cloud.
Cloud Agent	A Cloud agent collects the host, entity, and log data from the host where you deploy the Cloud agent. It can connect to Oracle Management Cloud directly or through a Gateway.
Data Collector	A Data Collector agent collects data from your on-premises OMR and uploads it to Oracle Management Cloud.
Entities	Entities are monitored resources such as databases, host servers, compute resources, or application servers.

Term	Definition
Entity Types	Entity types are a type of monitored resource, such as a host or database, which define where that entity fits in the Oracle Management Cloud hierarchical structure. In Oracle Management Cloud, each entity is defined by a set of characteristics, it has a parent and may have other children. For example, a generic host is an operating system (OS) independent target and it has children entities that are specific OS hosts, such as Linux and Windows. The metrics collection functionality takes advantage of this inheritance model so each monitored entity has entity-specific metrics as well as metrics inherited from each level it descended from. For example, Oracle Management Cloud collects metrics at level three that are common to all generic hosts, independent of the vendor. A Linux host, since its parent is a generic host, inherits all the metrics collected for generic hosts and its ancestors, as well as Linux-specific ones, if any.
Gateway	A Gateway agent acts as a channel between Oracle Management Cloud and other Cloud agents. Multiple Data Collector or Cloud agents can communicate with Oracle Management Cloud through a single Gateway.
JSON	JavaScript Object Notation (JSON) allows data to be concisely and precisely defined in a format that is both human and machine-readable. Oracle provides sample JSON files for defining entities. JSON files are then edited with your own custom parameters and are passed on to agents. This configuration step defines the entities with that agent and Oracle Management Cloud.
License Editions	License editions are pre-defined categories of Oracle Management Cloud offerings.
Log entity	A log entity is the host or the server from which the logs are collected.
Log source	A log source is a named group of log files. The files that belong to this group can be configured using patterns such as <code>/var/log/ssh*</code> . A log source can be associated with one or more parsers.
omcli	Oracle Management Cloud agent control command line interface utility (<code>omcli</code>) is used to interface with Cloud agents and define entities using customized JSON files.
Oracle Cloud Instance	An Oracle Cloud instance is a virtual machine, or a set of virtual machines, with CPU and memory resources, running a specific operating system and hosting a specific Cloud service offering.
Oracle home	Oracle home refers to a directory where Oracle products are installed, pointed to by an environment variable. Multiple active Oracle homes can exist on the same host.

Term	Definition
Oracle Java Cloud Service	The Oracle Java Cloud Service is a part of the platform service offerings in Oracle Public Cloud Services. Powered by Oracle WebLogic Server, it provides a platform on top of Oracle's enterprise-grade cloud infrastructure for developing and deploying new or existing Java EE applications. Optionally, you can enable Oracle Coherence within Oracle Java Cloud Service to use Coherence caching and data grid functionality.
Oracle Management Repository (OMR)	OMR is a schema in an Oracle Database where all the information collected by Oracle Enterprise Manager Cloud Control Management Agents is stored. It consists of objects such as database jobs, packages, procedures, tables, views, tablespaces, and so on.
Oracle Wallet	An Oracle Wallet is a password-protected container used to store private keys, certificates, and trusted certificates needed by secure communication between software components.
Oracle WebLogic Server	Oracle WebLogic Server is the Java EE application server, part of the Oracle Fusion Middleware suite of products, used for building and deploying enterprise applications.
Oracle WebLogic Server Cluster	An Oracle WebLogic Server Cluster consists of multiple Oracle WebLogic Server instances running simultaneously and working together to provide increased scalability and reliability.
Parser	A parser is a named entity used to define how to parse all log entries in a log source and extract field information. It uses one or multiple parse expressions and a log entry delimiter to parse all log entries in a log source. It also specifies how the parsed content is converted into fields.
Parse expression	A parse expression is the regular expression used to parse a log entry.
Security Certificate	A Security Certificate, or a Digital Certificate, is an electronic document that proves the ownership of a public key used for secure communication over a network.
WebLogic domain	A WebLogic domain is a logically related group of Oracle WebLogic Server resources. Domains include a special Oracle WebLogic Server instance called the Administration Server , which is the central point from which you configure and manage all resources in the domain. Usually, you configure a domain to include additional Oracle WebLogic Server instances called Managed Servers . You deploy web applications, EJB, and other resources onto the Managed Servers and use the Administration Server for configuration and management purposes only.

1


Configure Oracle Log Analytics

To get started with Oracle Log Analytics, follow these steps.

Topics:

- [Install the Cloud Agent and Enable Oracle Log Analytics](#)
- [Perform Prerequisite Configuration Tasks](#)
- [Perform Oracle Log Analytics Specific Configuration Tasks](#)

Skip the above steps in the following cases:

- [Set Up Autonomous Database Audit Log Collection](#)
- [Ingest Logs from OCI Object Storage Buckets](#)
- [Use Fluentd for Log Collection](#)
- [Collect Logs from Oracle Autonomous Database User Tables](#) ( [Tutorial](#)).

Install the Cloud Agent and Enable Oracle Log Analytics

You must perform the following tasks to install cloud agents.

Note:

You can access your log data on Oracle Log Analytics by:

- Installing the cloud agent that collects logs from your target host.
- Uploading log data on demand. See [Upload Logs to Oracle Log Analytics on Demand](#).
- Ingesting logs from Oracle Cloud Infrastructure Object Storage. See [Ingest Logs from OCI Object Storage Buckets](#).
- Using the open source data collector software, Fluentd to collect logs. See [Use Fluentd for Log Collection](#).

The following table lists the tasks that you must perform if you're using the cloud agent to enable Oracle Log Analytics to collect the log data.

Required Role: To complete these tasks, you must have the *Oracle Management Cloud Administrator* role. If this role isn't assigned to you or you're not sure, then ask your system administrator to ensure that the role is assigned to you in Oracle Cloud. See [About Oracle Log Analytics Roles and Users](#).

Task	Required / Optional	Description
<i>Task 1:</i> Understand the agent deployment topology.	Required	Review and understand the deployment topology of Oracle Management Cloud agents that are required to set up Oracle Log Analytics. See Understand the Architecture of Oracle Management Cloud in <i>Installing and Managing Oracle Management Cloud Agents</i> .
<i>Task 2:</i> Review the prerequisites for deploying Oracle Management Cloud agents.	Required	Review the hardware and software requirements for deploying Oracle Management Cloud agents. See Generic Prerequisites for Deploying Oracle Management Cloud Agents in <i>Installing and Managing Oracle Management Cloud Agents</i> .
<i>Task 3:</i> Grant the privileges to the cloud agent to read the log files.	Required	Make the log files readable to the Oracle Management Cloud agents. See the section <i>Requirement for Logs Collection on Unix</i> in the topic Generic Prerequisites for Deploying Oracle Management Cloud Agents in <i>Installing and Managing Oracle Management Cloud Agents</i> .
<i>Task 4:</i> Access the Oracle Management Cloud.	Required	Access the Oracle Management Cloud console and assign license editions. Ensure that the Log Collection toggle button is ENABLED . See Access Oracle Management Cloud and Enable License Editions in <i>Getting Started with Oracle Management Cloud</i> .
<i>Task 5:</i> Download the agent software.	Required	Download the agent software that contains the script required to install the Oracle Management Cloud agents. See Download Oracle Management Cloud Agent Software in <i>Installing and Managing Oracle Management Cloud Agents</i> .

Task	Required / Optional	Description
<i>Task 6:</i> Install a gateway on a host in your data center (the host should have internet access to Oracle Management Cloud.)	Optional	Install a gateway that acts as a channel between Oracle Management Cloud and all other Oracle Management Cloud agents. See <i>Install a Gateway in Installing and Managing Oracle Management Cloud Agents</i> .
<i>Task 7:</i> Install a data collector on the target host.	Optional (If you're setting up Oracle Log Analytics without an existing Oracle Enterprise Manager instance)	Install a data collector that uses a gateway to make data available to Oracle Management Cloud. See <i>Install a Data Collector in Installing and Managing Oracle Management Cloud Agents</i> .
<i>Task 8:</i> Install the cloud agent on the target host.	Required	Install the cloud agent that collects logs from a target host. See <i>Install Cloud Agents in Installing and Managing Oracle Management Cloud Agents</i> .
<i>Task 9:</i> Verify the deployment.	Required	See <i>Verify the Cloud Agent Installation, Verify the Data Collector Installation, and Verify the Gateway Installation in Installing and Managing Oracle Management Cloud Agents</i> .

Perform Prerequisite Configuration Tasks

Set up the environment to use Oracle Log Analytics by performing these prerequisite configuration tasks.

Required Role: To complete these tasks, you must have the *Oracle Management Cloud Administrator* role. If this role isn't assigned to you or you're not sure, then ask your system administrator to ensure that the role is assigned to you in Oracle Cloud. See [About Oracle Log Analytics Roles and Users](#).

Task	Required / Optional	Description
Add entities	Required	<p>After you've deployed your cloud agent, you must add the entities that your cloud agent will monitor. Review the list of entities available with Oracle Log Analytics and follow the workflow for adding new entities.</p> <p>See Adding Oracle Management Cloud Entities.</p> <p>For example, see Set Up Autonomous Database Audit Log Collection and Tutorial to Add a WebLogic Server Entity to Log Analytics and Later Infrastructure Monitoring.</p>

Task	Required / Optional	Description
Create groups	Optional	<p>After you have added entities, you can create Groups to monitor and analyze log entries of those entities. A Group is a single logical unit that can include targets of the same type (for example, all your production databases). You can view, create, and delete groups in the Administration console in Oracle Management Cloud.</p> <p>You can choose from the following types of groups:</p> <ul style="list-style-type: none"> <p>Static: A static group does not have any qualifying criteria. The membership management for a static group is typically manual or static and you must decide which entities should be included in a static group. It is best suited for a group of entities whose membership is unlikely to change frequently. Static groups are created, updated, or deleted in the Administration console in Oracle Management Cloud.</p> <p>To create a static group, see <i>Manage Groups in Working with Oracle Management Cloud</i>.</p> <p>Dynamic: A dynamic group allows you to add entities to a group based on set membership criteria. In dynamic environments where new entities come into the system frequently, entities that match the membership criteria are added automatically to a dynamic group.</p> <p>Dynamic groups can only be created using REST APIs. See All REST Endpoints in <i>Oracle Management Cloud Common REST API</i>.</p> <p>For more information about dynamic groups,</p>

Task	Required / Optional	Description
		see Manage Groups in <i>Working with Oracle Management Cloud</i> .
Create notification channels	Optional	You may want to be actively notified through email, by push notifications (mobile devices), or have a third-party application take action when an Oracle Log Analytics alert is raised. Set up notification channels and reuse the channels across different alert rules. See Set Up Notification Channels in <i>Using Oracle Infrastructure Monitoring</i> .
Create Remediation Action	Optional	You can create a remediation action that'll be performed automatically in response to an alert. You can create a Remediation Action using the Event Service API. Contact your Oracle Support or Sales Representative for more information about accessing and using the Event Service API. For an example of creating a remediation action, see Managing Incidents with Remediation Actions in <i>Using Oracle Orchestration</i> .

Add Entities to Oracle Log Analytics

Using Oracle Log Analytics, you can view and analyze the logs related to a set of entities.

Before you add entities to Oracle Log Analytics, review the list of entities whose logs are commonly analyzed using Oracle Log Analytics. See [Commonly Used Oracle Log Analytics Entities](#) and [Additional Entities in Oracle Log Analytics](#).

You can add an entity to Oracle Log Analytics by:

- Using the **Add Entities** page in the Administration console in Oracle Management Cloud. See Add Entities from the Console in *Using Oracle Infrastructure Monitoring*.
- Creating a JSON file containing the details about that entity and then running an `omcli` command. If you're adding multiple entities, you can either create separate JSONs for each entity and run the `omcli` command once for each file, or you can create a master JSON containing all the required entity details and run the `omcli` command only once.

The JSON file should contain all the mandatory properties and attributes for the entity. The mandatory association, such as association between the host and the target instance, is created automatically. However, you can also add non-mandatory associations in the JSON. For steps and example, see [Add Entity by Creating a JSON File](#).

It is recommended that you use the **Add Entities** page in the Administration console to discover the entity, than by creating a JSON file.



Note:

All hosts are automatically added as entities when a cloud agent is installed and you don't have to add them separately. However, monitoring of host entities is disabled by default. For information on how to enable host monitoring, see [Enable Host Monitoring in Using Oracle Infrastructure Monitoring](#).

Perform Oracle Log Analytics Specific Configuration Tasks

You must perform the following tasks to start viewing your log data in Oracle Log Analytics.

Required Role: To complete these tasks, you must have the *Oracle Management Cloud Administrator* or *Oracle Log Analytics Administrator* role. If one of these roles isn't assigned to you or you're not sure, then ask your system administrator to ensure that the role is assigned to you in Oracle Cloud. See [About Oracle Log Analytics Roles and Users](#).

Task	Description
<i>Task 1:</i> Create a parser.	By creating a parser, you define how the fields are extracted from a log entry for a given type of log file. See Create a Parser and Sample Parse Expressions .
<i>Task 2:</i> Create a log source.	To monitor a log file, you must create a new log source and specify the parser to extract the fields in the log data. See Create a Log Source and Out-of-the-Box Log Sources .
<i>Task 3:</i> Configure the entity association for the log source that you created.	Enable the log source that you created for a specific entity to start collecting log data. If you've enabled Oracle Infrastructure Monitoring on specific entities, then those entities are automatically available on Oracle Log Analytics to configure entity association for the log sources. See Configure New Entity Associations .

Related Topics:


- To set up syslog monitoring for your logging system event messages, see [Set Up Syslog Monitoring](#).
- To set up database instance monitoring for the database instance records extracted based on the SQL query that you provide in the log source configuration, see [Set Up Database Instance Monitoring](#).

Set Up Syslog Monitoring

Syslog is a commonly used standard for logging system event messages. The destination of these messages can include the system console, files, remote syslog servers, or relays.

Oracle Log Analytics allows you to collect and analyze syslog data from various sources. You just need to configure the syslog output ports in the syslog servers. Oracle Log Analytics monitors the output ports, accesses the remote syslog contents, and performs the analysis.

Syslog monitoring in Oracle Log Analytics lets you listen to multiple hosts and ports. The protocols supported are TCP and UDP.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click **Create source**.
Alternatively, in the **Log Sources** section, you can click the available number of log sources link and then in the Log Sources page, click **Create**.
This displays the Create Log Source dialog box.
3. In the **Source** field, enter the name for the log source.
4. From the **Source Type** list, select **Syslog Listener**.
5. Click **Entity Type** and select the required entity type such as **Host**.
6. Click **File Parser** and select **Syslog Standard Format**.
7. In the **Listener Pattern** tab, click **Add** to specify the details of the listener to which Oracle Log Analytics will listen to collect syslogs.
Enter the listener port that you specified as the output port in the syslog configuration file in the syslog server, select either **UDP** or **TCP** (recommended for heavy traffic) as the required protocol, and select **Enabled**.
Repeat this step for adding multiple listener ports.
8. Click **Save**.
9. In the Log Sources page, select the newly created syslog source (`testSyslog` in this case) and click **Associated Targets**.
10. In the Associated Targets: `<log source name>` page, click **Add**.
11. Select the host name or host names with which you want to associate the source and click **Select**.
12. In the Associated Targets: `<log source name>` page, click **Save**.

View Syslog Data

You can use the **Log Source** field in the **Fields** panel of Oracle Log Analytics to view syslog data.

1. From Oracle Log Analytics, click **Log Source** in the **Fields** panel.
2. In the **Filter by Log Source** dialog box, select name of the syslog source that you created, and click **Submit**.

Oracle Log Analytics displays the syslog data from all the configured listener ports. You can analyze syslog data from different hosts or devices.

Set Up Database Instance Monitoring

Oracle Log Analytics can extract database instance records based on the SQL query that you provide in the log source configuration. You can define a parser for the database instance log records using Oracle Log Analytics.

Currently, the supported database types are Oracle Database Instance (`omc_oracle_db_instance`), Microsoft SQL Server Database Instance (`omc_sqlserver_db_instance`), and MySQL Database Instance (`omc_mysql_db_instance`).

Overall Flow for Collecting Database Logs


The following are the high-level tasks for collecting log information stored in a database:


- Creating your log source
- Providing entity credentials
- Associating an entity with the log source

Note:


By default, after you've installed the cloud agent, it collects the database instance records for 30 days. If you want to extract records that're more than 30 days old, then update the property before the event collection from the database begins:

```
omcli setproperty agent -allow_new -name  
loganalytics.database_sql.max_oldDays -value  
<newValue_for_max_oldDays>
```

For an example of how to collect Database Audit Logs, see *Collect Database Audit Logs to Analyze Using Oracle Log Analytics* ( [Tutorial](#)).

For an example of how to collect logs from Oracle Autonomous Database user tables, see *Collect Logs from Oracle Autonomous Database User Tables* ( [Tutorial](#)).

Create the Database Log Source

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click **Create source** .
Alternatively, in the **Log Sources** section, you can click the available number of log sources link and then in the Log Sources page, click **Create**.
This displays the Create Log Source dialog box.
3. In the **Source** field, enter the name for the log source.
4. From the **Source Type** list, select **Database**.

5. Click **Entity Type** and select the required entity type. For example, **Oracle Database Instance**.
6. In the **Database Queries** tab, click **Add** to specify the details of the SQL query based on which Oracle Log Analytics instance collects database instance logs.
7. Click **Configure** to display the Configure Column Mapping dialog box.
8. In the Configure Column Mapping dialog box, map the SQL fields with the field names that would be displayed in the actual log records.

Specify a **Sequence Column**.

See [SQL Query Guidelines](#).

Note that the first mapped field with a data type of `Timestamp` is used as the time stamp of the log entry. If no such field is present, then the collection time is used as the time of the log entry.

Click **Done**.

9. Repeat Step 6 through Step 8 for adding multiple SQL queries.
10. Select **Enabled** for each of the SQL queries and then click **Save**.

Provide the Database Entity Credentials

For each entity that's used for collecting the data defined in the Database log source, you need to provide the necessary credentials to be used to connect to the entity and run the SQL query. These credentials need to be registered in a credential store that's maintained locally by the cloud agent. The credentials are used by the cloud agent to collect the log data from the entity.

Create the JSON File with Credentials Information

Create a JSON file that contains the credential information as the following:

```
[{
  "entity": "<Enter Entity Type>.<Enter Entity Name>",
  "name": "LCAgentDBCreds",
  "type": "DBCredsNormal",
  "usage": "LOGANALYTICS",
  "globalName": "AgentUserCredential",
  "description": "SSH Credential for fetching the data from db tables
via sql",
  "properties": [{
    "name": "USERNAME",
    "value": "CLEAR[username]"
  },
  {
    "name": "PASSWORD",
    "value": "CLEAR[password]"
  },
  {
    "name": "ROLE",
    "value": "CLEAR[rolename]"
  }
  ]
}]
```

For example, for a database named `avdf_instance` and user name, password, and role as `sys`, `syspasswd`, and `SYSDBA` respectively, the JSON file should contain:

```
[{
  "entity":"omc_oracle_db_instance.avdf_instance/orcl",
  "name":"LCAgentDBCreds",
  "type":"DBCredsNormal",
  "globalName":"AgentUserCredential",
  "usage":"LOGANALYTICS",
  "description":"DB Credentials",
  "properties":[{"
    "name":"USERNAME",
    "value":"CLEAR[sys]"
  },
  {
    "name":"PASSWORD",
    "value":"CLEAR[syspasswd]"
  },
  {
    "name":"ROLE",
    "value":"CLEAR[SYSDBA]"
  }
]}]
```

The name, type and usage fields should be set to `LCAgentDBCreds`, `DBCredsNormal` and `LOGANALYTICS` respectively. The `globalName` field needs to be unique within the credential store managed by the local cloud agent. The `ROLE` property is optional.

Register the Credential Information

You need to register the credential information with the cloud agent.

1. Go to the Oracle Management Cloud host computer.
2. To create a credential store if it was not created earlier,
 - a. Stop the cloud agent:

```
omcli stop agent
```
 - b. Run the following command from the `<AGENT_BASE_DIR>/agent_inst/bin` location:

```
omcli add_credential_store agent -no_password
```

See `omcli Command Options` in *Working with Oracle Management Cloud*.
 - c. Start the cloud agent:

```
omcli start agent
```
3. To register the credential information, run the following command from the `<AGENT_BASE_DIR>/agent_inst/bin` location:

```
omcli add_credentials agent -credential_file <PATH_TO_CRED_JSON_FILE>
```

See `omcli Command Options` in *Working with Oracle Management Cloud*.

 **Note:**

By default, after you've installed the cloud agent, it collects the database instance records for 30 days. If you want to extract records that're more than 30 days old, then update the property before the event collection from the database begins:


```
omcli setproperty agent -allow_new -name
loganalytics.database_sql.max_oldDays -value
<newValue_for_max_oldDays>
```

Next: To associate the entity with your log source, see [Working with Entity Associations](#).

Set Up Autonomous Database Audit Log Collection

The Autonomous Database can be discovered from the Oracle Management Cloud discovery UI. If the log collection is enabled for the tenant, then the database logs begin to collect in Oracle Log Analytics.

Required Role: To complete these tasks, you must have the *Oracle Management Cloud Administrator* role. See [About Oracle Log Analytics Roles and Users](#).

1. Access the Oracle Management Cloud. See [Access Oracle Management Cloud in Getting Started with Oracle Management Cloud](#).
2. Click **OMC Navigation**  icon > Navigate to **Administration > Entity Configuration > Licensing**. The Licensing page is displayed. Ensure that the Log Collection toggle button is **ENABLED**.

See [Enable License Editions in Getting Started with Oracle Management Cloud](#).

3. Discover Autonomous Database entity from the Oracle Management Cloud discovery UI.


See [Discover Autonomous Databases in Using Oracle Database Management for Autonomous Databases](#).

After the discovery process is complete, Oracle Management Cloud associates that Autonomous Database entity automatically with the log source *Oracle Unified Audit Trail Stored in Cloud Database*. Also, the **Standard Edition** license is auto-assigned to the Autonomous Database entity during discovery.

After the entity association, the log collection begins in Oracle Log Analytics, with the oldest logs collected first. So, ensure that the time range in the log explorer is sufficiently large to view the data from all the logs.

To disable the log collection from your Autonomous Database entity, delete the association of the entity with the log source *Oracle Unified Audit Trail Stored in Cloud Database*. To enable the log collection, you can create the association again.

To disable the log collection from all the entities for the tenant, disable the **Log Collection** toggle button in the Oracle Management Cloud **Licensing** page. This effectively stalls the log collection on the tenant from all the entities.

For an example of how to collect logs from Oracle Autonomous Database user tables, see *Collect Logs from Oracle Autonomous Database User Tables* ( [Tutorial](#)).

2

Upload Logs to Oracle Log Analytics on Demand

You can access your log data on Oracle Log Analytics by installing the cloud agent that collects logs from your target host. However, Oracle Log Analytics lets you to upload log data on demand. This is useful when you have log data from old applications that aren't supported by the Oracle Management Cloud agents, but you need to analyze them for troubleshooting. In addition, if you have applications that aren't set up to be monitored by Oracle Log Analytics, and if the applications return a large number of log entries, then you can use the on-demand upload feature to easily analyze the large volumes of log data.

The following are features of on-demand upload:

- You can upload a single log file or any compressed file (.zip, .gz, .tgz, .tar.tgz) containing multiple log files.
- The maximum file size for a single upload (single file or a ZIP file) is 1 GB.
- You can name each upload for easy reference.
- Using the named reference, you can upload files at different times to the same upload name.
- You can select a log source for the log file, and a parser for the specific log source. Oracle Log Analytics selects a default one if you don't specify a parser.



Note:

To generate any configured real time alerts for logs, make sure to specify the entity associated with the logs while uploading.

Topics:

- [Upload Log Files Using REST API](#)
- [Upload Log Files Using ODU Client](#)
- [Upload Log Files Using ODU Wizard](#)
- [Verify an Upload](#)
- [Delete Uploaded Log Files](#)

Upload Log Files Using REST API

For more information about uploading on-demand log data to Oracle Management Cloud using REST API, see [Working with Log Analytics: Upload](#) in *Oracle Management Cloud Common REST API*.

Upload Log Files Using ODU Client

You can use the **On Demand Upload** (ODU) client to upload your log files to Oracle Log Analytics through a command-line interface. This simple interface enables you to automate your uploads by integrating the ODU client into your application.

Topics:

- [Prepare to Use the ODU Client](#)
- [General Syntax of ODU Client Commands](#)
- [Upload Log Files](#)
- [Check Submission Status of an Upload](#)
- [Retry an Upload](#)

Note: If the ODU Client that you're using is no longer supported and a newer version of the client is available for download, then the message `Incompatible client version, download latest version and try again` is displayed. To download the latest ODU client, see [Download the ODU Client](#).

Prepare to Use the ODU Client

Supported Platforms:

- Oracle Linux 6.3 or later
- Oracle Linux 7.0 or later
- Microsoft Windows 7
- Microsoft Windows 10
- Microsoft Windows Server 2012
- Ubuntu 14.04 LTS
- Ubuntu 16.04 LTS
- macOS High Sierra 10.13.6
- Solaris 11

Prerequisite Authentication and Environment Information:

- **Domain URL:** Obtain OMC URL from **Agents** page:
 1. On the Oracle Management Cloud home page, click the **OMC Navigation Menu** on the top-left corner and navigate to **Administration > Agents**.
 2. On the Agents page, click the **Download** tab. The Agent Software Download page is displayed.
 3. Select `Cloud Agent` from the **Agent Type** drop-down list. The **OMC_URL** is displayed. This `OMC_URL` represents the domain URL in the ODU client.
- **User Name:** Obtain the user name that you use for accessing Oracle Management Cloud.
- **Password:** This is the password that you use for accessing Oracle Management Cloud.

- **JAVA_HOME Environment Variable:** If you've not already defined this environment variable, then set it and point it to your JDK or JRE installation.

Download the ODU Client

To start using the ODU client, download the client and store it at an appropriate location on the host.

1. From Oracle Log Analytics, navigate to **Administration Home**. Click the gear icon on the top right corner, and click **Download ODU Client**.

The client download is initiated.

2. Select an appropriate location on the host to store the `odu_client.zip` file.

3. Unzip the `odu_client.zip` file. Navigate to the folder **odu_client > bin**.

The `bin` folder contains the files `odu-client` and `odu-client.bat`. To execute the ODU client commands, use the file `odu-client` in the Linux environment and the Windows Batch File `odu-client.bat` in the Windows environment.

4. Provide `Execute` permission to `odu-client` or `odu-client.bat` file that you'll use to upload the log files.

General Syntax of ODU Client Commands

Syntax

The following is the general syntax of the commands on the ODU client:

```
odu-client upload | status | retry [<command_options>]
```

- To execute the ODU client commands, use the file `odu-client` in the Linux environment and the Windows Batch File `odu-client.bat` in the Windows environment. The above syntax is using the `odu-client` file for the example.
- The following three commands can be executed using the ODU client.
 - **upload:** Uploads the log files.
 - **status:** Provides the submission status of an upload.
 - **retry:** Retries an upload that was initiated earlier and suspended. Starts a new upload if it had failed earlier.
- **command_options** can be obtained by using the `--help` option with the command:

```
odu-client upload | status | retry --help
```


Upload Log Files

Using the **upload** command, upload the log files to Oracle Log Analytics to start analyzing the log data. Some of the file types that are currently supported for upload are `.log`, `.req`, `.xml`, `.gz`, `.zip`, `.tgz`, and `.tar.gz`.

Command Syntax

```
odu-client upload [command_parameters] [optional_arguments] file1
file2 file3
```


- `file*` are the paths of the files or directories that must be uploaded.
- Provide the file paths at the end of the command after the command parameters and arguments.
- When you provide a file path that refers to a directory, only the log files under that directory are uploaded and the sub-directories are ignored.
- Ensure to provide a minimum of one file path.

Command Parameters and Optional Arguments

The following parameters are specific to this command:

Parameter	Description
<code>-D domain_URL, --domainURL domain_URL</code>	The tenant specific URL of your Oracle Management Cloud instance. For information on how to construct the domain URL, see Prepare to Use the ODU Client .
<code>-U User_Name, --username User_Name</code>	The user name that you use for accessing Oracle Management Cloud.
<code>-P Password, --password Password</code>	The password that you use for accessing Oracle Management Cloud.
<code>-u Upload_Name, --uploadName Upload_Name</code>	The container name to which you want to upload the log files. You can upload more files at a later point using the same upload name. The upload name can be used to filter log data from the Log Explorer. See Filter Logs by Data Uploaded on Demand .
<code>-s Log_Source, --logSource Log_Source</code>	The log source that must be used for processing the files in this upload.

The following optional arguments are specific to this command:

Argument	Description
<code>-e Entity_Name, --entityName Entity_Name</code>	The name of the entity that must be associated to the uploaded logs while processing the files in this upload.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>To generate any configured real time alerts for logs, make sure to specify the entity associated with the logs while uploading.</p> </div>
<code>-t Entity_Type, --entityType Entity_Type</code>	The entity type of the given entity. This is required when the entity name is provided.
<code>-c Character_Encoding, --charEncoding Character_Encoding</code>	The character encoding of the log files that are being uploaded.
<code>-p Parser, --parser Parser</code>	The parser to use for processing the log files. Use this option to override the default parser selection that'll be automatically made by Oracle Log Analytics.
<code>-F Date_Format, --dateFormat Date_Format</code>	The format for the date information that's available in the log files. Use this argument to remove any ambiguity (like 12/10 can be December 10th or 12th October) in identifying the format of the date in the given log entry. For example, when the date is 12/10, where it can be interpreted as 12th October or 10th December, you can use DAY_MONTH or MONTH_DAY to remove ambiguity. In case the date is 12/10/08, you can use DAY_MONTH_YEAR, MONTH_DAY_YEAR, or YEAR_MONTH_DAY.
<code>-Y Date_Year, --dateYear Date_Year</code>	The year information to use for processing the log entries when the log entries do not have the year information in the time stamp.
<code>-z Time_Zone, --timeZone Time_Zone</code>	The timezone information to use for processing the log entries. By default, the timezone information in the log entry is used for processing, if available. When the information is not available in the log entry, the value of this argument is considered when you input it with the upload command. In case, the value is not available from this argument or directly from the log entry, then the timezone of the entity is considered for processing. When no information is available on the timezone, the default value is UTC.
<code>--parallel Parallel_Files</code>	The number of files that the ODU client can simultaneously upload. This parameter takes an integer value and by default is set to 3. The maximum value accepted is 10.
<code>-b Data_Directory, --datadir Data_Directory</code>	The directory with write permission where the ODU client will store the information about the processing done on the log files. The default location is <code><USER_HOME>/ .odu-client</code> .

Argument	Description
<code>--proxyhost Proxy_Host</code>	The proxy host on which the HTTP proxy is used.
<code>--proxyport Proxy_Port</code>	The proxy port on which the HTTP proxy is used. By default, the proxy port is set to 80.
<code>-prop Properties_File, --properties Properties_File</code>	<p>The path to the properties file. The default location is <code><USER_HOME>/.odu-client/odu-client.properties</code>.</p> <p>The properties file can be used to store the values of the command options for the upload command. The values of the options that you input on the command line will override the values stored in the properties file. See Set Up the Properties File.</p>
<code>-i, --interactive</code>	<p>The argument to run the upload command in the interactive mode. When you launch this mode, you can interactively select the values of the command options from the provided list of values. After you set the values of the mandatory configuration properties like upload name, file, and log source, the option to submit the upload command is enabled.</p> <p>You must input the values of domain URL, user name, and password to run the upload command in the interactive mode.</p> <p>You can run the upload command in the interactive mode without using the properties file to provide the values of the command options.</p>

Optionally, to avoid entering the values of some of these parameters on the command-line every time you run the command, you can set the parameters in the properties file.

Examples

- The command to upload the Linux syslog files `MySysLogFile1` and `MySysLogFile2`:

```
odu-client upload -D https://myDomainURL -U myUserName -P Password -
u testUpload -s "Linux Syslog Logs" -e myEntity -t "Host (Linux)"
MySysLogFile1 MySysLogFile2
```

- Domain URL: `https://myDomainURL`
- User Name: `myUserName`
- Password: `Password`
- Upload Name: `testUpload`
- Log Source: `Linux Syslog Logs`
- Entity Name: `myEntity`
- Entity Type: `Host (Linux)`

The output of this command is:

```
Validating Credentials ..
Validating Client ..
```

```
Client logs are located at: "/home/user1/.odu-client/logs/odu-client.log"
```

```
Number of files to process :      2
Starting upload of files ... . (console will be updated every 25
seconds)
    2018-11-27 16:45:40 Status RunId:1 Total Files:2 Submitted:0
Failure:0
    2018-11-27 16:46:05 Status RunId:1 Total Files:2 Submitted:2
Failure:0
```

```
File(s) submission completed.
```

```
Client logs are located at: "/home/user1/.odu-client/logs/odu-client.log"
To check status run status --runId 1
```

You can check the status of your upload request using the status command. See [Check Submission Status of an Upload](#).

- The command to upload by using a properties file `config.prop`:

```
odu-client upload --properties config.prop
```

- The command to upload the log files in the interactive mode by using the properties file `config.prop` to input the domain URL, user name, and password:

```
upload --properties config.prop -i -u TestUpload
```

An interactive menu is displayed:

```
Choose an option [1-9] OR enter q (quit)
  Option                               Value
  -----                               -----
  1. Upload Name                         TestUpload
  2. Log Directory/File
  3. Log Source
  4. Entity Name (Optional)
  5. Log Parser (Optional)
  6. Character Encoding (Optional)
  7. Time Zone (Optional)
  8. DateFormat (Optional)
  9. Year (Optional)

Choose an option [1-9] OR enter q (quit)
Enter :
```

Follow the interactive steps to fill the required details and submit.

Set Up the Properties File

To avoid entering the authentication information and other necessary parameters with every command on the command-line, you can store the parameters permanently in the properties

file. By using the `--properties` command option on the command-line, you can specify the path to the properties file while uploading the log files.

1. Store the following sample properties file with a suitable name on your local host:

```
# -----Authentication information -----
domainURL=https://myDomainURL
username=myUserName
password=Password
# -----Other mandatory information-----
uploadName=MyPropertiesDemo
logSource=Linux Syslog Logs
file=D://Logs//syslogs
# -----Additional configuration information-----
entityName=myEntity
entityType=Host (Linux)
charEncoding=GBK
parser=Syslog Standard Format
dateFormat=DAY_MONTH
dateYear=2018
timeZone=europe/berlin
parallel=5
proxyhost=proxyHost.example.com
proxyport=80
```

2. Update the parameters in the `key=value` format in the properties file. Remove those parameters from the file that aren't necessary for your application.

Check Submission Status of an Upload

You can use the **status** command to obtain the submission status of the upload that was initiated earlier. When you trigger an upload of the log files, the upload request is submitted to the processing engine on Oracle Management Cloud. The authentication information that's submitted with the upload command is stored on the local database, and is reused by ODU client for the status command.

Command Syntax

```
odu-client status [optional_arguments]
```

Optional Arguments

The following optional arguments are specific to this command:

Arguments	Description
<code>-r Run_ID, --runId Run_ID</code>	The run ID of an upload that was initiated earlier. Each upload request is provided with a unique run ID which can be later used to obtain submission status of the request.
<code>-b Data_Directory, --datadir Data_Directory</code>	The directory with write permission where the ODU client will store the information about the processing done on the log files. The default location is <code><USER_HOME>/ .odu-client</code> .

Arguments	Description
<code>--pageSize Page_Size</code>	The number of rows in a page when the ODU client display the result of the status command. By default, the page size is set to 20 rows.

Examples

- Command to show high-level submission status of all the upload requests:

```
odu-client status
```

An example output of this command is:

```
Status
Page 1 of 4 Total Records:80
RunId CreatedOn                               CreatedBy
UploadName          Source                    Total-Files      Submitted
Failure
83 Tue Apr 02 14:32:34 PST 2019                myUserName
testUpload          Linux Syslog Logs          10                10                0
82 Tue Apr 02 14:32:01 PST 2019                myUserName
testUpload          Linux Syslog Logs          10                10                0
81 Tue Apr 02 14:31:27 PST 2019                myUserName
testUpload          Linux Syslog Logs          10                10                0
...
...
64 Tue Apr 02 14:21:22 PST 2019                myUserName
testUpload          Linux Syslog Logs          10                10                0
Search for matches or Enter <runId> OR Enter n to see next items OR Enter
p to see previous items OR Enter q to quit
:
```

From the above output, you can use the `RunId` to search and navigate within pages. You can specify the `RunId` to view the corresponding details.

- Command to check the submission status of an upload request that has run ID 1:

```
odu-client status --runId 1
```

An example output of this command is:

```
Status RunId: 1

Upload Inputs:
  --uploadName      : "testUpload"
  --domainURL       : "https://myDomainURL"
  --username        : "myUserName"
  --logSource       : "Linux Syslog Logs"

Status Summary: Total-Files:2 Pending:0 Initiated:0 Submitted:2 Failure:0
```

```

Log Files

Page 1 of 1                Total Records:2
  No.
File
Status      Note
  1. \.odu-
client\bin\MySysLogFile1      Submitted
  2. \.odu-
client\bin\MySysLogFile2      Submitted

Search for matches OR Enter n to see next items OR Enter p to see
previous items OR Enter q to quit
:

```

In case of failure, the details of the failure are available under the **Note** section.

- Command to check the submission status of an upload request that has run ID 5 for a modified page size that has 30 rows:

```
odu-client status --runId 5 --pageSize 30
```

Retry an Upload

Use the **retry** command to restart an upload process that was initiated earlier but was suspended, interrupted, or failed earlier. The retry process restarts the upload from where it was suspended, thus conserving time and network bandwidth. In case of a previously failed attempt, it uploads the file again.

Command Syntax

```
odu-client retry [optional_arguments]
```

Command Parameters and Optional Arguments

The following optional arguments are specific to this command:

Arguments	Description
<code>-r Run_ID, --runId Run_ID</code>	The run ID of an upload that was initiated and suspended earlier. Each upload request is provided with a unique run ID which can be later used to obtain submission status of the request or to restart the upload.
<code>--parallel Parallel_Files</code>	The number of files that the ODU client can simultaneously upload. This parameter takes an integer value and by default is set to 3. The maximum value accepted is 10.
<code>-b Data_Directory, --datadir Data_Directory</code>	The directory with write permission where the ODU client will store the information about the processing done on the log files. The default location is <code><USER_HOME>/ .odu-client</code> .

Example

- Command to view the run IDs that can be retried:

```
odu-client retry
```

An example output of this command is:

```
retry
Following runId(s) can be retried
Page 1 of 1 Total Records:2
RunId   CreatedOn                CreatedBy   UploadName
Source   Total-Files   Pending   Failure   Submitted
48   Fri Mar 22 16:07:38 PST 2019   myUserName testUpload Linux Syslog
Logs    72             66         0          6
47   Fri Mar 22 11:53:41 PST 2019   myUserName testUpload Linux Syslog
Logs    72             72         0          0
```

```
Search for matches OR Enter n to see next items OR Enter p to see
previous items OR Enter q to quit or Enter <runId> to retry
:
```

From the above output, you can use the `RunId` to search and navigate within pages. You can specify the `RunId` to retry the upload.



- Command to retry an upload process that has the run ID 25 and to upload 6 files in parallel:

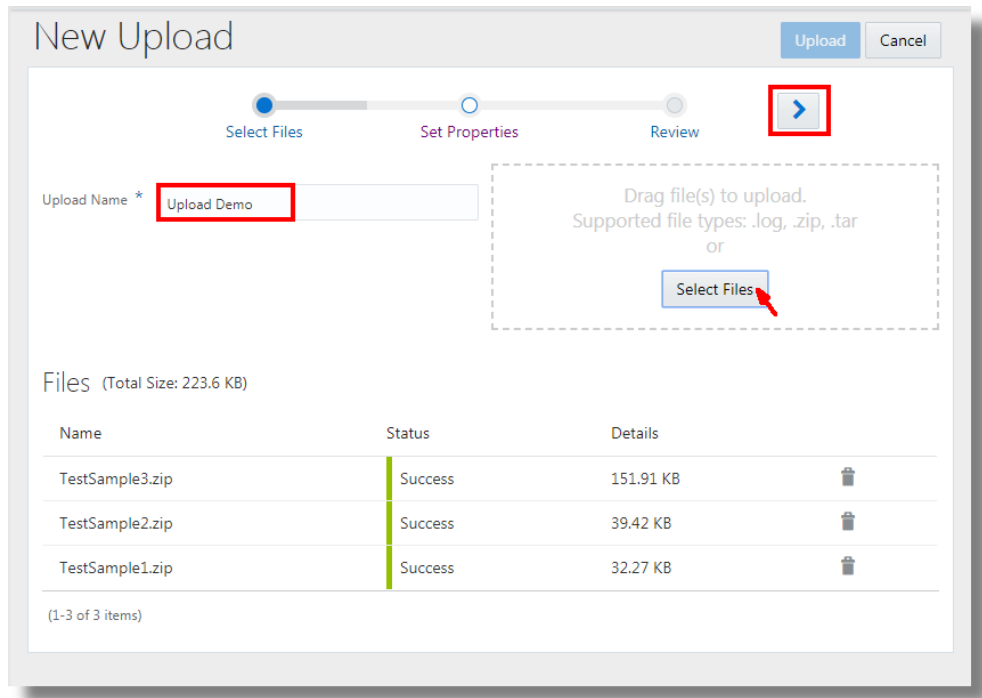
```
odu-client retry --runId 25 --parallel 6
```

Upload Log Files Using ODU Wizard

You can upload your log files using **On Demand Upload (ODU)** wizard that's available on the service console of Oracle Log Analytics.


ODU wizard is a simple and convenient tool for uploading the files through a friendly user interface. Follow the intuitive steps prompted by the wizard to select the files for upload, set the properties of the files, and review before uploading them. You can start uploading the log files by accessing the wizard in one of the following methods:

- From the OMC Navigation  icon on the top left corner of the Oracle Log Analytics interface, click **Administration Home**. In the **Uploads** section, click **New Upload**.
 - From the OMC Navigation  icon on the top left corner of the Oracle Log Analytics interface, click **Log Admin**, and click **Uploads**. In the resulting uploads page, click **New Upload** on the top left corner of the page.
1. **Select Files:** Click **Select Files** button and select the log files or archive files to upload to Oracle Log Analytics. Enter the **Upload Name**. This is the container name to which you want to upload the log files.



At this point, the files are uploaded to Oracle Management Cloud. The files will be processed after the completion of the final step of the ODU wizard.

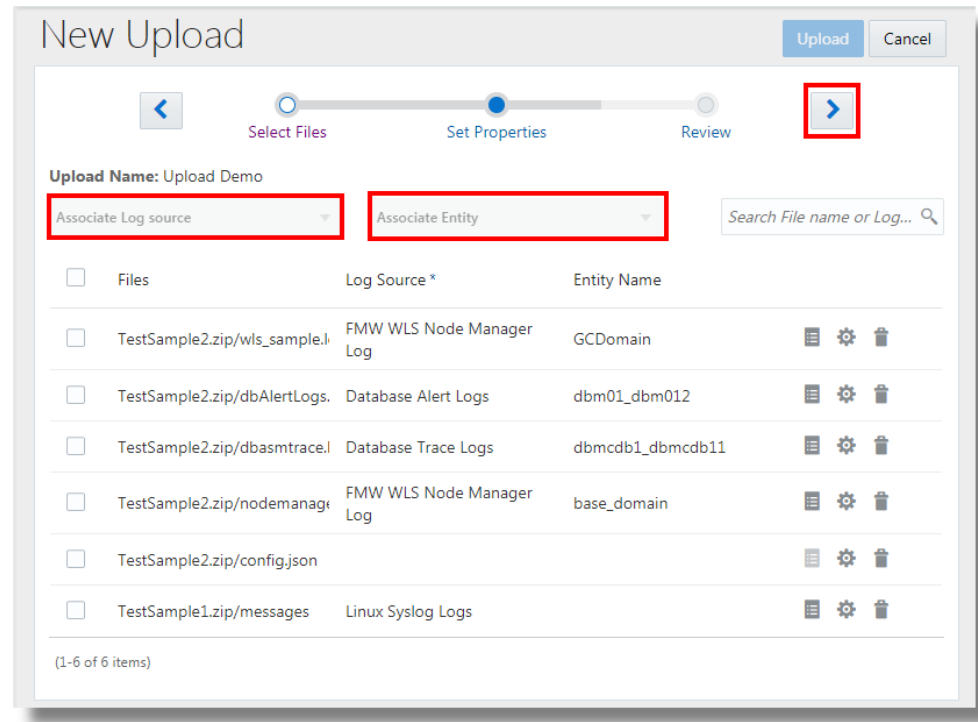
A summary of the files selected for upload is displayed. Note that the maximum individual file size or the overall upload size is 1 GB. You'll be notified if your file size or the upload size exceeds the maximum limit.



Click Next  icon. Alternatively, you can also click **Set Properties**.

2. **Set Properties:** The page displays the list of files selected in the first step, including the individual files in the selected archive. To set the properties of a file,
 - a. Select the check box next to the file name.
 - b. Select the log source from the **Associate Log Source** drop down menu.
 - c. Based on the log source that you selected, the **Associate Entity** drop down menu is populated with the corresponding entities that you can associate with the log source. Select the entity.


 **Note:**

To generate any configured real time alerts for logs, make sure to specify the entity associated with the logs while uploading.



- d. To preview the result of processing the log entries with the selected log source, click **Preview Log Source**  icon. You can change the log source and preview again to ensure that the log file is processed as required.
- e. Optionally, you can specify the advanced properties when the required parameters are not available in the log entry. Click **Advanced Properties**  icon. From the drop down menu, select the values of the parameters **Timezone**, **Char Encoding**, **Date Format**, and **Year**. Click **Save**.
 - **Timezone**: The timezone information to use for processing the log entries. By default, the timezone information in the log entry is used for processing. When the information is not available in the log entry, the value that you select from the menu is considered. In case, the value is not available from this menu or directly from the log entry, then the timezone of the entity is considered. When no information is available on the timezone, the default value considered is UTC.
 - **Char Encoding**: The character encoding of the log files that are being uploaded.
 - **Date Format**: The format for the date information that's available in the log files. Use this parameter to remove any ambiguity (like 12/10 can be 10th December or 12th October) in identifying the format of the date in the given log entry. For example, when the date is 12/10, where it can be interpreted as 12th October or 10th December, you can use DAY_MONTH or MONTH_DAY to remove ambiguity. In case the date is 12/10/08, you can use DAY_MONTH_YEAR, MONTH_DAY_YEAR, or YEAR_MONTH_DAY.
 - **Year**: The year information to use for processing the log entries when the log entries do not have the year information in the timestamp.



You can remove selected files from the list, if required. If the file that's removed is a part of the archive, then the remaining files of the archive will still be uploaded.

- f. Click Next  icon. Alternatively, you can click **Review**.
3. **Review:** Review the properties of the files that you've selected for upload. To confirm the properties and initiate the upload, click **Upload**.

Oracle Log Analytics indexes and processes the files. After the upload is complete, you can view the log data in the Log Explorer. You can verify the upload in the Uploads page.


Verify an Upload


Upon completing an upload of the on-demand log data, you can view the summary of uploads and verify the file status.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface.
In the OMC Navigation bar, click **Administration Home**.
2. In the Oracle Log Analytics Configuration page, click the available count of uploads link in the **Uploads** section.
This displays the latest 500 on-demand uploads.
To refresh the summary of uploads, click the refresh  icon in the top-right corner of the display.
3. Look for the log files corresponding to the upload name that you used for the upload. Verify that the upload is complete.

Delete Uploaded Log Files

Upon completing an upload of the on-demand log data, you can view the summary of uploads and verify the file status. If, in case you notice that the file upload failed, you can delete the single failed file and upload it again.

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface.
In the OMC Navigation bar, click **Administration Home**.
2. In the Oracle Log Analytics Configuration page, click the available count of uploads link in the **Uploads** section.
This displays the latest 500 on-demand uploads and the summary of the files successful, failed, or in-progress counts.
If required, click the refresh icon in the top-right corner of the display to refresh the summary of uploads.
3. To delete an on-demand upload, click the check box next to the upload name. To delete multiple on-demand uploads, click the corresponding check boxes. Click **Delete**.
4. To delete a file in an on-demand upload, click the upload name.
This displays the list of files that were uploaded in the specified upload. You can view the status of the upload of each file adjacent to the file name.

To delete a file, click the **Open Menu** () icon adjacent to the file name entry and select **Delete**.

3

Ingest Logs from OCI Object Storage Buckets

You can ingest the log data from Oracle Cloud Infrastructure (OCI) Object Storage using REST API. The data obtained can be OCI audit logs, OCI flow logs, or other OCI logs stored in buckets. After the buckets are registered with Oracle Log Analytics, they are periodically polled for new logs.

To access OCI, provide the OCI authentication credentials that are different from the credentials you use for accessing Oracle Management Cloud. Therefore, register the OCI authentication credentials in the credential store of Oracle Management Cloud to establish connection.

Prerequisites:

- **Enable Log Collection:** Access the Oracle Management Cloud console and assign license editions. Ensure that the Log Collection toggle button is **ENABLED**.
See Access Oracle Management Cloud and Enable License Editions in *Getting Started with Oracle Management Cloud*.
- **Object Name Prefix:** The object names must have at least one prefix (directory) associated with them to process them successfully, for example, `access-logs/object1` and `sample/2020-10-01T01:10Z.log.gz`.
- **Identify Buckets:** Ensure that you've identified and noted the buckets in OCI Object Storage where the log data is collected. You must register these buckets with Oracle Log Analytics so that buckets are polled for log data.
- **For other OCI Logs** (not OCI audit logs and OCI flow logs): To facilitate Oracle Log Analytics in polling for new logs and collect them periodically, ensure that the logs are stored in the OCI Object Storage buckets in the *name incremented order*. When the logs are stored in the buckets in the name incremented order by using the API supplied by OCI, the log name will carry the time stamp. This enables identification of new logs based on their time stamp, for example, `us-phoenix-1/ad1/2019-10-04T12:50Z.log.gz`.
- **OCI Account Information:** Note the following information from your OCI account before you perform steps for registering the OCI buckets:
 - OCI Region
 - OCI Namespace

To register the OCI Object Storage buckets for log collection:

1. Create Credentials in Oracle Management Cloud credential store. See [Create Credential for OCI Authentication](#).
2. After ingesting the logs, to start viewing your OCI audit and flow log data in Oracle Log Analytics log explorer, you can use out-of-the-box log sources OCI Audit Logs and OCI VCN Flow Logs. However, to monitor other OCI logs, create a new log source and parser. See [Create a Parser](#) and [Configure New Log Sources](#).
3. Configure Oracle Log Analytics to collect logs from the list of OCI Object Storage buckets:

- a. Create a configuration file in the json format and store it on the local machine, for example, `bucket_config.json`.

```
{
  "logType": "<log_type>",
  "bucketsInfo": [
    {
      "credential": "<OCI_credential>",
      "namespace": "<OCI_namespace>",
      "region": "<OCI_region>",
      "pollSince": "<poll_time_range>",
      "logSourceName": "<log_source_name>",
      "buckets": [
        {"name": "<bucket1>"},
        {"name": "<bucket2>"}
      ]
    }
  ]
}
```

In the above format,

- **logType:** Specify `AUDIT` for OCI audit logs, `FLOW` for OCI flow logs, or `OCI_LOGS_GENERIC` for other OCI logs to ingest.
- **credential:** The name given in the credential store to the OCI credentials that you created in step 1
- **namespace:** OCI namespace collected from your OCI account
- **region:** OCI region collected from your OCI account
- **pollSince:** The time range from when the polling for logs must be performed. Specify one of the following:
 - `BEGINNING` to collect the logs from the time they began to store in the buckets
 - Absolute time from when the logs must be collected in the standard Oracle Log Analytics format. For example, `2019-12-17T00:00:00.000Z` where `Z` is UTC time zone.
 - `CURRENT_TIME` to collect the logs from the time the buckets are registered in Oracle Log Analytics which is the default setting
- **logSourceName:** In case of OCI audit logs and OCI flow logs, this is optional. In case of other OCI logs, specify the name of the log source that must be used in Oracle Log Analytics. To effectively use the scope of this parameter for different buckets, see the example json below.
- **buckets:** The OCI Object Storage bucket names from which you want to collect the logs

In the following example json file created in the above format, the log source name `OCI Audit Logs` which is global is applicable for all the buckets, but the

log source name `LinuxSyslogSource` is local and is applicable for a single bucket:

```
{
  "logType": "AUDIT",
  "bucketsInfo": [
    {
      "credential": "John_OCI_credential",
      "namespace": "ad3n3pqrs6oc",
      "region": "us-phoenix-1",
      "pollSince": "CURRENT_TIME",
      "logSourceName": "OCI Audit Logs",
      "buckets": [
        {"name": "bucket1_name",
         "logSourceName": "LinuxSyslogSource"},
        {"name": "bucket2_name"}
      ]
    }
  ]
}
```

- b. To register the buckets with Oracle Log Analytics, run the cURL command in the following format:

```
curl -X POST -k -u '<username>:<password>' -H 'X-USER-IDENTITY-DOMAIN-NAME:<identity_domain_name>' "https://<OMC_URL>/serviceapi/logan.uploads/registerOSSConfig" -H 'Content-Type:application/json' -d "@<bucket_config_json_file>"
```

In the above format:

- **username:** Your user name to access the Oracle Management Cloud account. Depending on the type of your cloud account, the username will be in one of the following formats:
 - **<username>** for Oracle Identity Cloud Service (IDCS) based account
 - **<tenant_name>.<username>** for Traditional Cloud Account

For information on the types of cloud accounts, see About Oracle Cloud Accounts in *Getting Started with Oracle Cloud*.
- **password:** The password to access the Oracle Management Cloud account
- **OMC_URL:** Obtain OMC URL from **Agents** page.
 - i. On the Oracle Management Cloud home page, click the OMC Navigation Menu on the top-left corner and navigate to **Administration > Agents**.
 - ii. On the Agents page, click the **Download** tab. The Agent Software Download page is displayed.
 - iii. Select **Cloud Agent** from the Agent Type drop-down list. The **OMC_URL** is displayed. Note the URL.
- **identity_domain_name:** Depending on the type of your cloud account, the identity domain name will be one of the following:
 - **IDCS Identity Domain:** For IDCS based cloud account, typically of the format `idcs-j29b928a146e4bdd7fef12a6e6a9excm`. Collect this from your cloud account details page.

- **Tenant Name:** For Traditional Cloud Account, typically of the format `acme`.
Follow the same steps as those to obtain `OMC_URL`.
TENANT_NAME is displayed above `OMC_URL`.

For information on the types of cloud accounts, see About Oracle Cloud Accounts in *Getting Started with Oracle Cloud*.

- **bucket_config_json_file:** The OCI properties file that you created in step a.

An example cURL command to register the buckets with Oracle Log Analytics in case of using a Traditional Cloud Account:

```
curl -X POST -k -u 'acme.JohnDoe:john_password' -H 'X-USER-IDENTITY-DOMAIN-NAME:acme' "https://acme.example.com:4443/serviceapi/logan.uploads/registerOSSConfig" -H 'Content-Type:application/json' -d "@bucket_config.json"
```

An example cURL command to register the buckets with Oracle Log Analytics in case of using an IDCS Cloud Account:

```
curl -X POST -k -u 'JohnDoe:john_password' -H 'X-USER-IDENTITY-DOMAIN-NAME:idcs-j29b928a146e4bdd7fef12a6e6a9excm' "https://omc-fb68f2df9f4a27bda5c45778f62f41.example.com/serviceapi/logan.uploads/registerOSSConfig" -H 'Content-Type:application/json' -d "@bucket_config.json"
```

After registering the buckets information, wait for the log collection to begin. Adjust the time range in your log explorer to view the data based on their time stamp. The oldest logs are collected first.

In case of errors with select few buckets, the registering action is cancelled on all the buckets listed in the configuration file.

View the Bucket Configuration

After configuring the collection of logs from OCI Object Storage buckets, you can view the configuration at any point later.

Run the cURL command in the following format:

```
curl -X GET -k -u '<username>:<password>' "https://<OMC_URL>/serviceapi/logan.uploads/getOSSConfig?logType=<log_type>"
```

In the above format:

- **logType:** Specify `AUDIT` for OCI audit logs, `FLOW` for OCI flow logs, or `GENERAL` for any other OCI logs.
- **username:** Your user name to access the Oracle Management Cloud account. Depending on the type of your cloud account, the username will be in one of the following formats:
 - `<username>` for Oracle Identity Cloud Service (IDCS) based account
 - `<tenant_name>.<username>` for Traditional Cloud Account.

Follow the same steps as those to obtain OMC_URL. **TENANT_NAME** is displayed above OMC_URL.

- **password**: The password to access the Oracle Management Cloud account
- **OMC_URL**: Obtain OMC URL from **Agents** page.
 1. On the Oracle Management Cloud home page, click the OMC Navigation Menu on the top-left corner and navigate to **Administration > Agents**.
 2. On the Agents page, click the **Download** tab. The Agent Software Download page is displayed.
 3. Select **Cloud Agent** from the Agent Type drop-down list. The **OMC_URL** is displayed. Note the URL.

An example output of the command:

```
[ {
  "bucketId": "20d11212-59de-34d5-84aa-76d04b5b7166",
  "ociCredential" : "John_OCI_credential",
  "ociNamespace" : "ad3n3pqrs6oc",
  "ociRegion" : "us-phoenix-1",
  "ociBucket" : "odu-auditlog-pull",
  "logType" : "AUDIT",
  "createdOn" : "2019-11-08T14:50:12.058Z"
}]
```

Unregister the Buckets for Log Collection

At any point after the OCI Object Storage buckets are registered for log collection, you can unregister them. After unregistering, the log collection from the specified buckets is stopped. However, previously collected log data from those buckets will continue to be available in Oracle Log Analytics.

Run the cURL command in the following format:

```
curl -X DELETE -k -u '<username>:<password>' "https://<OMC_URL>/serviceapi/logan.uploads/unregisterOSSConfig" -d '{"bucketIds":["<bucket_IDs>"]}' -H 'Content-Type:application/json'
```

In the above format:

- **username**: Your user name to access the Oracle Management Cloud account. Depending on the type of your cloud account, the username will be in one of the following formats:
 - **<username>** for Oracle Identity Cloud Service (IDCS) based account
 - **<tenant_name>.<username>** for Traditional Cloud Account. Follow the same steps as those to obtain OMC_URL. **TENANT_NAME** is displayed above OMC_URL.
- **password**: The password to access the Oracle Management Cloud account
- **OMC_URL**: Obtain OMC URL from **Agents** page.
 1. On the Oracle Management Cloud home page, click the OMC Navigation Menu on the top-left corner and navigate to **Administration > Agents**.

2. On the Agents page, click the **Download** tab. The Agent Software Download page is displayed.
 3. Select **Cloud Agent** from the Agent Type drop-down list. The **OMC_URL** is displayed. Note the URL.
- **bucketIds**: The IDs of the buckets that you want to unregister. You can obtain this by viewing the bucket configuration using REST API. See [View the Bucket Configuration](#).

Note that if a bucket has undergone the cycle of register > unregister > register again, then all the logs from that bucket is collected again.

4

Use Fluentd for Log Collection

Use the open source data collector software, Fluentd to collect log data from your source. Install the Oracle supplied output plug-in to allow the log data to be collected in Oracle Log Analytics.

Fluentd software has components which work together to collect the log data from the input sources, transform the logs, and route the log data to the desired output. Oracle provides the output plugin installing which, you can ingest the logs from any of your input sources into Oracle Log Analytics.

Prerequisites:

Install Fluentd and Input Plug-ins: Before performing the following steps, ensure that you have installed Fluentd and the relevant input plug-ins for your input sources. See <https://docs.fluentd.org/v0.12/quickstart/installation>.

Enable Log Collection: Access the Oracle Management Cloud console and assign license editions. Ensure that the Log Collection toggle button is **ENABLED**.

See Access Oracle Management Cloud and Enable License Editions in *Getting Started with Oracle Management Cloud*.

Topics:

- [Install the Output Plug-In](#)
- [Edit Fluentd Configuration File](#)
- [Configure the Format of the Incoming Log Events](#)

Install the Output Plug-In

Use the *gem* file provided by Oracle for the installation of the output plug-in.

Prerequisites: To ensure that the logs from your input source can be processed by the output plug-in provided by Oracle, verify that the input log events conform to the prescribed format, for example, by configuring the *record_transformer* filter plug-in to alter the format accordingly. See [Configure the Format of the Incoming Log Events](#).

Note: If you must monitor multiple log files, then you can use the *Multi Process Workers* feature to process them simultaneously. It's recommended that to process *N* files in a set up with *C* CPU's for this plugin, use `ceil (N/C)` workers. You can have up to *N/Workers* number of files on each worker. For more information, see [Multi Process Workers](#) in *Fluentd Documentation*.

1. Download the output plug-in file `fluent-plugin-oracle-omc-loganalytics-1.0.gem` and store it in your location machine.
2. Install the Fluentd output plug-in by running the following command:

- For RubyGems:

```
gem install fluent-plugin-oracle-omc-loganalytics-1.0.gem
```

- For td-agent:

```
td-agent-gem install fluent-plugin-oracle-omc-  
loganalytics-1.0.gem
```

3. Configure Fluentd to route the log data to Oracle Log Analytics. Edit the Fluentd configuration file and save it as `fluentd.conf`. See [Edit Fluentd Configuration File](#).

If you're using td-agent, edit the configuration file provided by td-agent.

4. To start collecting logs on Oracle Log Analytics, run Fluentd or td-agent:

- Fluentd:

```
fluentd -c <path to fluentd.conf>
```

- Start td-agent:

```
TZ=utc /etc/init.d/td-agent start
```

To troubleshoot errors, if you encounter any during log collection or during the set up, see [docs.fluentd.org](#). If you use td-agent, then you can use the log file `/var/log/td-agent/td-agent.log` to debug issues.

To stop td-agent at any point, run the following command:

```
TZ=utc /etc/init.d/td-agent stop
```

Edit Fluentd Configuration File

Edit the configuration file provided by Fluentd or td-agent and provide the information pertaining to Oracle Log Analytics and other customizations.

The output plug-in buffers the incoming events before sending them to Oracle Log Analytics. The plug-in will separate the log events into chunks by the value of the fields `Tag` and the `sourceName`. The chunks are then transferred to Oracle Log Analytics.

It is recommended that a secondary plug-in is configured which would be used by Fluentd to dump the backup data when the output plug-in continues to fail in writing the buffer chunks and exceeds the timeout threshold for retries. Also, for unrecoverable errors, Fluentd will abort the chunk immediately and move it into secondary or the backup directory. For more information, see [Fluentd Documentation](#).

The Fluentd configuration file will be of the following format:

```
<match pattern>  
  @type oracle_omc_loganalytics  
  http_proxy           <your_proxy>  
  omc_oauth_client_id <your_omc_oauth_client_id>  
  omc_oauth_client_secret <your_omc_oauth_client_secret>  
  omc_oauth_username  <your_omc_oauth_user>
```

```

omc_oauth_password          <your_omc_oauth_password>
omc_oauth_scope             <your_omc_oauth_scope>
omc_oauth_token_url         <your_omc_oauth_token_url>
omc_oauth_upload_url        <your_omc_oauth_upload_url>
<buffer tag, sourceName>
  @type file
  path                      <your_path_buffer_chunk_files>
  overflow_action block
</buffer>
<secondary>
  @type file
  path                      <your_path_backup_failed_chunks>
</secondary>
</match>

```

In the above format,

- **http_proxy**: The proxy URL, if you're using one
- **omc_oauth_client_id**: The ID of the client application
- **omc_oauth_client_secret**: Secret of the client application
- **omc_oauth_username**: The user name to access Oracle Management Cloud
- **omc_oauth_password**: The user name to access Oracle Management Cloud
- **omc_oauth_scope**: Scope of the personal access token. To compose this URL:
 1. On the Oracle Management Cloud home page, click the OMC Navigation Menu on the top-left corner and navigate to **Administration > Agents**.
 2. On the Agents page, click the **Download** tab. The Agent Software Download page is displayed.
 3. Select **Cloud Agent** from the Agent Type drop-down list. The **OMC_URL** is displayed. Note the URL.
 4. Insert OMC_URL in the following scope URL format:

```
https://<OMC_URL>/serviceapi/%20offline_access
```

The resulting URL is the scope.

- **omc_oauth_token_url**: The URL to obtain the personal access token. To compose this URL:

1. Your IDCS console URL is found when you are logging in to Oracle Management Cloud. It follows the format:

```
https://<IDCS_DOMAIN_NAME>/ui/v1/adminconsole/?root=users
```

2. Obtain **IDCS_DOMAIN_NAME** from your IDCS console URL.
3. Insert IDCS_DOMAIN_NAME in the following token URL format:

```
https://<IDCS_DOMAIN_NAME>/oauth2/v1/token
```

The resulting URL is the token URL.

- **omc_oauth_upload_url**: The URL to upload the logs to Oracle Log Analytics. To compose this URL, obtain OMC_URL:
 1. Obtain OMC_URL by following the same steps as for *omc_oauth_scope* parameter.
 2. Insert OMC_URL in the following upload URL format:

```
https://<OMC_URL>/serviceapi/loganalytics/logEventsReceiver/  
upload
```

The resulting URL is the upload URL.

- **tag**: The tag that you have defined for the log event in the *record_transformer* plug-in, which will be used for distinguishing the log content that must be consumed by the buffer for chunking. See [Configure the Format of the Incoming Log Events](#).
- **sourceName**: The log source name that you have defined for the log event in the *record_transformer* plug-in, which will be used for distinguishing the log content that must be consumed by the buffer for chunking. See [Configure the Format of the Incoming Log Events](#).
- **path** in buffer section: The location where the buffer chunks are stored.
- **path** in secondary section: The location where the backup of the failed chunks are stored. The exact location of the backup directory is determined by the parameter *root_dir*. See **output: Backup for broken chunks** in [Fluentd Documentation](#).

For more information on the OAuth properties, see [Authentication: Enable OAuth With REST API](#) in *REST API for Oracle Management Cloud*.

Some of the *optional* properties that you can define in the configuration file:

- **omc_oauth_token_max_tries**: Defines the maximum number of retries to get the access token. Default value is 3.
- **omc_oauth_token_expiration_time**: The access token expiration time. The default value is 3600 sec.
- **omc_oauth_unauthorized_retry_time**: The wait time before a retry attempt is made to obtain access token. The default value is 3.
- **verify_ssl**: The default value is `true`. Disable it if you do not want to verify ssl connection.

The output plug-in uses the following configuration in the base framework of Fluentd to control the buffering and flushing behavior. It is recommended that these values are retained for better throughput:

```
chunk_limit_size 1m  
flush_interval 60s  
flush_thread_interval 0.5  
flush_thread_burst_interval 0.05  
flush_thread_count 10
```

To override the above buffer configuration values, see [Fluentd Documentation](#).

Here's an example configuration file with the mandatory properties where some of the parameters are defined as environment variables:

```
<match pattern>
  @type oracle_omc_loganalytics
  http_proxy           "#{ENV['HTTP_PROXY']}"
  omc_oauth_client_id  "#{ENV['OMC_OAUTH_CLIENT_ID']}"
  omc_oauth_client_secret "#{ENV['OMC_OAUTH_CLIENT_SECRET']}"
  omc_oauth_username   "#{ENV['OMC_OAUTH_USER']}"
  omc_oauth_password   "#{ENV['OMC_OAUTH_PASS']}"
  omc_oauth_scope      https://<OMC_URL>/serviceapi/
%20offline_access
  omc_oauth_token_url  https://<IDCS_DOMAIN_NAME>/oauth2/v1/token
  omc_oauth_upload_url https://<OMC_URL>/serviceapi/loganalytics/
logEventsReceiver/upload
  <buffer tag, sourceName>
    @type file
    path                <your_path_buffer_chunk_files>
    overflow_action block
  </buffer>
  <secondary>
    @type file
    path                <your_path_backup_failed_chunks>
  </secondary>
</match>
```

Configure the Format of the Incoming Log Events

The incoming log events must be in a specific format so that the Fluentd plug-in provided by oracle can process the log data, chunk them, and transfer them to Oracle Log Analytics.

Ensure that the following mandatory parameters are available in the Fluentd event processed by the output plug-in, for example, by configuring the *record_transformer* filter plug-in :

- **message:** The actual content of the log obtained from the input source
- **entityType:** The entity type with which this log data is associated
- **entityName:** The entity name with which this log data is associated
- **sourceName:** The log source name. See the list of available out-of-the-box log sources at [Out-of-the-Box Log Sources](#).
- **tag:** The tag which will be used by Oracle's Fluentd plug-in to filter the log events that must be consumed by Oracle Log Analytics.

The following optional parameters can be included in the *record_transformer* filter plug-in:

- **logEntity:** The entity with which this log data is associated, typically a file name
- **logMetadata:** The metadata specifying the key-value pairs. Each *key* must be from the out-of-the-box fields available in Oracle Log Analytics or user-defined by following the steps in [Create a Field](#). Also, to avoid the metadata pair from getting rejected during processing, ensure that the *value* is of the correct type.

The fields are typically used to associate with the parse expressions.

Note that configuring the *record_transformer* filter plug-in is only one of the ways of including the required parameters in the incoming events. There could be other ways too.

When you use the input tail plugin `@type multiline`, set the parameter `multiline_flush_interval` to a suitable value to ensure that all the log lines are uploaded to Oracle Management Cloud in time. If the parameter is not set, then the last line of an inactive log file will be processed only when stopping the td-agent.

An example input configuration that can be used for monitoring log files from the log sources *Apache HTTP Server Access Logs* and *Linux Syslog Logs*:

```
<source>
  @type tail
  <parse>
    @type multiline
    multiline_flush_interval 5s
    format_firstline /([0-9A-Fa-f.:%/]+)\s+([\w-]+)\s+([\w-]+)
\s+/
    format1 /^(?<message>.*)/
  </parse>
  path access.log
  pos_file access.log.pos
  path_key tailed_path
  tag omc.apache.access
</source>

<filter omc.apache.access>
  @type record_transformer
  enable_ruby true
  <record>
    entityType omc_host_linux
    entityName host.example.com
    sourceName "Apache HTTP Server Access Logs"
    logMetadata "${Environment: 'test', Type: 'testMetadata'}"
    logEntity "${record['tailed_path']}"
  </record>
</filter>

<source>
  @type tail
  <parse>
    @type multiline
    multiline_flush_interval 5s
    format_firstline /\w+\s*\d{2}\s*\d{2}:\d{2}:\d{2}\s\w+/
    format1 /^(?<message>.*)/
  </parse>
  path /var/log/messages
  pos_file var.log.messages.pos
  path_key tailed_path
  tag omc.var.log.messages
</source>

<filter omc.var.log.messages>
  @type record_transformer
  <record>
    entityType omc_host_linux
    entityName host.example.com
    sourceName "Linux Syslog Logs"
```



```
      logEntity "${record['tailed_path']}"
    </record>
  </filter>
```

In the above example:

- The first *in_tail* plugin reads the logs from the tail of the log file `access.log`, and tags them with `omc.apache.access`.
- The second *in_tail* plugin reads the logs from the tail of the log file `/var/log/messages`, and tags them with `omc.var.log.messages`.

An example Fluentd event that adheres to the specified format:

```
tag: omc.apache.access
time: 1572600797
record: {
  "message": "xx.xx.xx.xx - - [14/Feb/2019:18:25:14 +0100] \"GET /
administrator/ HTTP/1.1\" 200 4263 \"-\" \"Mozilla/5.0 (Windows NT 6.0;
rv:34.0) Gecko/20100101 Firefox/34.0\" \"-\"",
  "entityType": "omc_host_linux",
  "entityName": "host.example.com",
  "sourceName": "Apache HTTP Server Access Logs",
  "logMetadata": {
    "Environment": "test",
    "Type": "testMetadata"
  },
  "logEntity": "access.log"
}
```


Part II

Administer Oracle Log Analytics


Get started by reviewing the workflow for administering Oracle Log Analytics.

To monitor a custom log file, you, as the Oracle Log Analytics administrator must create a new log source and a parser.

You can create a log source or a parser in multiple ways:

- From the Oracle Log Analytics Configuration page
- From the OMC Navigation  icon on the top left corner of the Oracle Log Analytics interface

You can also import source and parser definitions from an XML file. Importing source definitions imports all source-related content such as Extended Fields, Tags, Lookups, and Labels.

To import source or parser definitions, click the gear  icon on the top right corner of the Configuration page and select the XML definition file.


Topics:

- [Create a Parser](#)
- [Configure New Log Sources](#)
- [Administer: Other Actions](#)

5

Create a Parser

By creating a parser, you define how the fields are extracted from a log entry for a given type of log file.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Parsers** section, click **Create parser**.
Alternatively, in the **Log Parsers** section, you can click the available number of log parsers link and then in the Log Parsers page, click **Create**.
3. Select from **Regex Type**, **JSON Type**, or **XML Type** from the options.
The **Create Parser** page is displayed. The **Guided** and **Manual** processes for creating the parser are available for the **Regex Type**.

WARNING:

For **Regex Type**, after you've selected the **Manual** mode to create the parser, you can't change to the **Guided** mode.

If you selected **JSON Type** or **XML Type**, then only the **Manual** process is enabled.

Note:

You can also create a parser using an out-of-the-box parser as a template. Select an out-of-the-box parser in the Log Parsers page, click **Create Like**, and modify the values in the fields as per your requirement.

Oracle Log Analytics provides many out-of-the-box parsers for log sources, such as Java Hotspot Dump logs, multiple systems, such as Linux, AIX, Siebel, PeopleSoft, and so on as well as for entity types, such as Oracle Database, Oracle WebLogic Server, and Oracle Enterprise Manager Cloud Control. You can access the complete list of supported parsers and log sources from within the Oracle Log Analytics user interface.

Topics:

- [Guided Creation of the Regex Type Parser](#)
- [Manual Creation of the Regex Type Parser](#)
- [Create JSON Type Parser](#)
- [Create XML Type Parser](#)
- [Preprocess Log Events](#)

Guided Creation of the Regex Type Parser

If you want to generate the parser expression using the **Parser Builder**, then click on the **Guided** tab.

1. In the **Parser Name** field, enter the parser name. For example, enter `OBIO Performance Log Parser`.
2. In the **Example Log Content** field, paste the contents from a log file that you want to parse. You can alternatively click **Add from file**, and select the log file that you want to parse.

The log records are extracted from the file and displayed in the **Example Log Content** field.

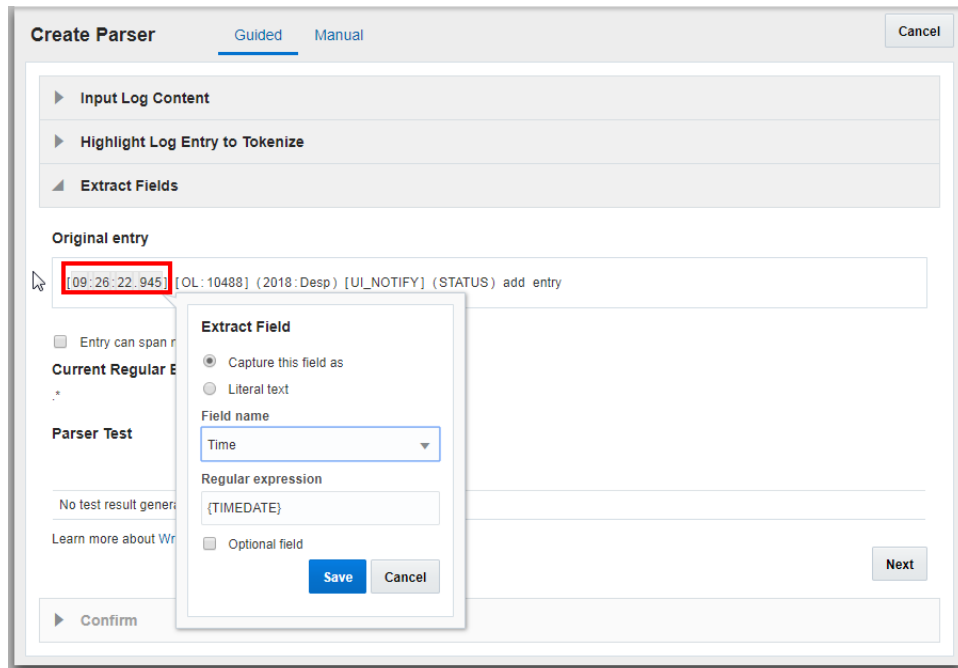
Select the **Handle entire file as a single log entry** check box, if required. If you do, then you might want to consider selecting the check box **Enable raw-text searching on this content**. This option enables you to search the log records with the *Raw Text* field. When enabled, you can view the original log content with the raw text.

Click **Next**.

3. From the log content, select the lines that represent a single log entry. To select multiple lines, hold down **Ctrl** or **Shift** key, and select.

Click **Next**.

4. In the log entry, click on each field. The Extract Field dialog box opens.



- To capture the type of field, select **Capture this field as** radio button, click the down arrow under **Field Name**, and select the field name that it corresponds with. Based on the field type, the field value in the log record will be replaced with the regular expression for that field. For example, select the time data in

the log entry, and select **Time** field name. Then the `{TIMEDATE}` regular expression is displayed.

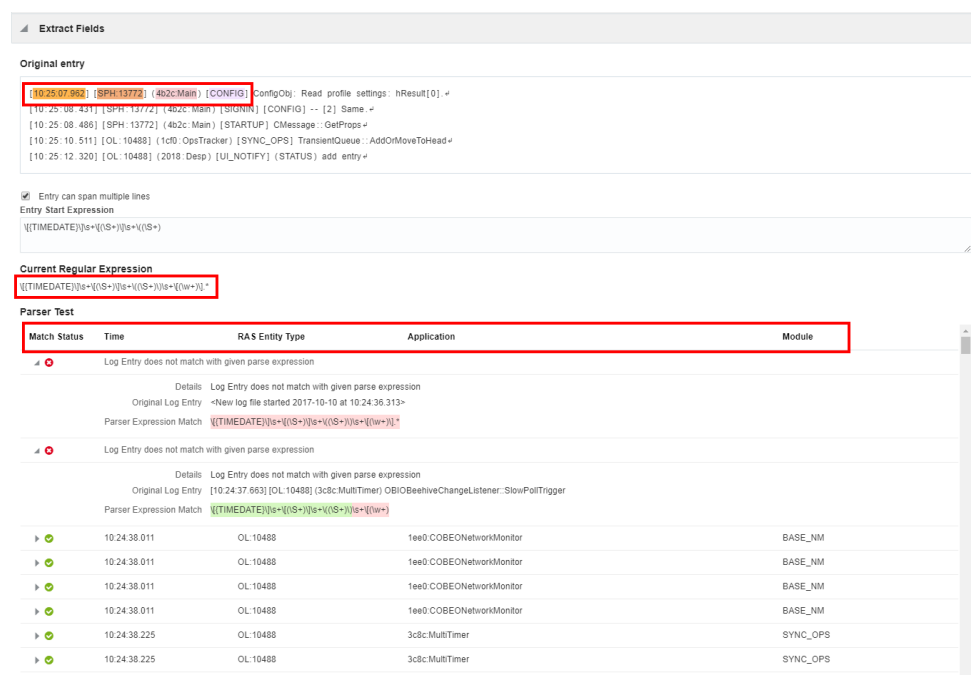
- To capture the selected field by its literal text, select the **Literal text** radio button.

Click **Save**.

5. The log entry can be a single line or multiple lines. If the log entry spans multiple lines, then by default, **Use Autogenerated Parse Expression** input method for the parse expression is selected. Alternatively, you can manually enter the log record's start expression.

Optionally, you can enter the end expression too. Use **End Expression** to indicate the end of the log record. When a log record is written to the file in multiple chunks and you want the agent to pick up the complete log record that includes all the chunks, use the end regex. The agent waits till the end regex pattern is matched to collect the log record. The same format rules apply for the end expression as that of **Entry Start Expression**.

6. After the fields are extracted, the regular expression is displayed and tested. The results of the test are displayed in a table.



The test result displays the **Step Count** which is a good measure of the efficiency of the extract expression in extracting the required fields from the example log content. Ensure to fine tune your regex expression to optimize the matches and to reduce the step count.

7. Click **Next**.
 The fields that you identified for parsing are listed along with the corresponding field names, type of data, descriptions of the fields, and the regular expressions.
8. Confirm the parser data by clicking **Create Parser**.

The parser builder will validate your input with the existing parsers. If an existing parser can be used for the example log content provided earlier, then you'll be redirected to the specific Parser Creation page.

Manual Creation of the Regex Type Parser

If you've identified the parser expression for your logs, then click on the **Manual** tab.

WARNING:

For **Regex Type**, after you've selected the **Manual** mode to create the parser, you can't change to the **Guided** mode.

1. In the **Parser** field, enter the parser name. For example, enter Database Audit Log Entries.
Provide suitable description to the parser for easy identification.
2. In the **Example Log Content** field, paste the contents from a log file that you want to parse, such as the following:

```
Tue May 6 23:51:23 2014
LENGTH : '157'
ACTION : [7] 'CONNECT'
DATABASE USER: [3] 'sys'
PRIVILEGE : [6] 'SYSDBA'
CLIENT USER: [8] 'user1'
CLIENT TERMINAL: [0] ''
STATUS: [1] '0'
DBID: [9] '592398530'
```

```
Tue May 6 23:51:23 2014 +08:00
LENGTH : '157'
ACTION : [7] 'CONNECT'
DATABASE USER: [3] 'sys'
PRIVILEGE : [6] 'SYSDBA'
CLIENT USER: [8] 'user1'
CLIENT TERMINAL: [0] ''
STATUS: [1] '0'
DBID: [9] '592398530'
```

Select the **Handle entire file as a single log entry** check box, if required. If you do, then you might want to consider selecting the check box **Enable raw-text searching on this content**. This option enables you to search the log records with the *Raw Text* field. When enabled, you can view the original log content with the raw text.

3. In the **Parse Expression** field, enter the expression with delimiters.

The parse expression is unique to each log type and depends on the format of the actual log entries. In this example, enter:

```
\w+\s+(\w{3})\s+(\d{1,2})\s+(\d{2})\s+(\d{2})\s+(\d{4})
(?:\s+([+-]\d{2})\s+)?.*
```

 **Note:**

- Oracle Log Analytics also lets you parse the local time and date available in the log files by using the `TIMEDATE` expression format.

So for those logs that use the `TIMEDATE` expression format, the preceding parse expression should be written as:

```
{TIMEDATE}\s+(\d{1,2})\s+(\d{2})\:(\d{2})\:(\d{2})\s+(\d{4})
(?:\s+([+-]\d{2})\:\?\d{2})?.*
```

- If some log events don't have a year assigned in the log, then Oracle Log Analytics assigns the year to those events.
- If the time and date are not specified in the parser for a log file that's parsed as a single log record, then the *last modified time* of the log file is considered by Oracle Log Analytics to obtain the corresponding data. Note that the date and time data can be obtained only for the log files that're sourced through the agent and not for the log files that're uploaded on-demand.

 **Note:**

- Don't include any spaces before or after the content.
- If you've included hidden characters in your parse expression, then the **Create Parser** interface issues an error message:

```
Parser expression has some hidden control characters.
To disable this default response, uncheck the Show hidden control
characters check box when the error message appears.
```

To learn more about creating parse expressions, see [Sample Parse Expressions](#).

4. Select the appropriate **Log Record Span**.

The log entry can be a single line or multiple lines. If you chose multiple lines, then enter the log record's start expression.

In the example, the start expression can be:

```
\w+\s+(\w{3})\s+(\d{1,2})\s+(\d{2})\:(\d{2})\:(\d{2})\s+(\d{4})
```

Optionally, you can enter the end expression too. Use **End Expression** to indicate the end of the log record. When a log record is written to the file in multiple chunks and you want the agent to pick up the complete log record that includes all the chunks, use the end regex. The agent waits till the end regex pattern is matched to collect the log record. The same format rules apply for the end expression as that of **Entry Start Expression**.

If you've selected **Multiple Lines** as the **Log Record Span**, then you can select **Handle entire file as a single log entry**. This option lets you parse an entire log file as a single

log record. This is particularly useful when parsing log sources such as Java Hotspot Dump logs, RPM list of packages logs, and so on.

5. In the **Fields** tab, select the relevant type for each component of the log entry.

For each component, select the name. The first component in the example can be entered as follows:

- Field Name: Month (Short Name)
- Field Data Type: STRING
- Field Description: Month component of the log entry time as short name, such as Jan
- Field Expression: `(\w{3})`

When you hover on a field name, an information icon appears. Hovering on the icon displays the description of the field in a floating window.

6. In the **Functions** tab, click **Add** to optionally add a function to pre-process log events. See [Preprocess Log Events](#).
7. Click the **Parser Test** tab to view how your newly created parser extracts values from the log content.

You can view the list of events that failed the parser test and the details of the failure.

Click **Save** to save the new parser that you just created.

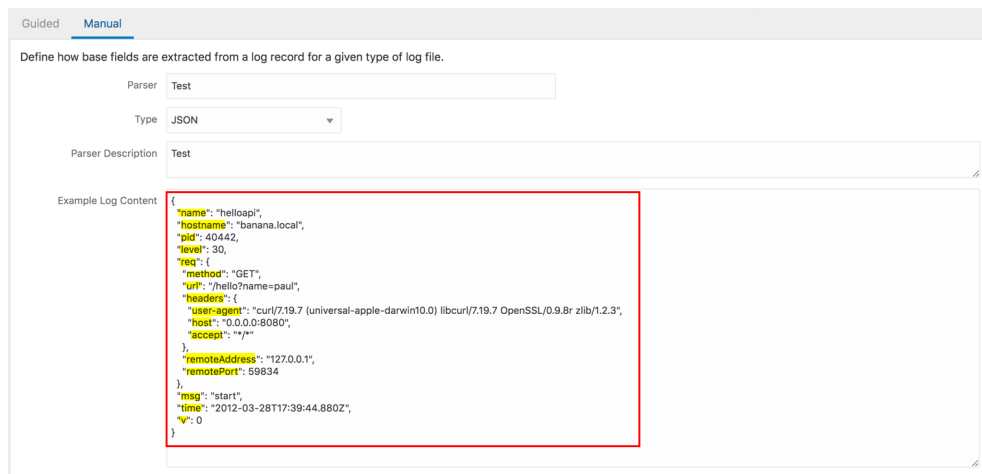
To abort creating the parser of regex type and to switch to creating a JSON type parser, under **Type**, select **JSON**.

Create JSON Type Parser

1. In the **Parser** field, enter the parser name. For example, enter Database Audit Log Entries.

Provide suitable description to the parser for easy identification.

2. In the **Example Log Content** field, paste the contents from a log file that you want to parse, such as the following:



Guided **Manual**

Define how base fields are extracted from a log record for a given type of log file.

Parser: Test

Type: JSON

Parser Description: Test

Example Log Content:

```
{
  "name": "helloapi",
  "hostname": "banana.local",
  "ip": 40442,
  "level": 30,
  "req": {
    "method": "GET",
    "uri": "/hello?name=paul",
    "headers": {
      "user-agent": "curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8r zlib/1.2.3",
      "host": "0.0.0.0:8080",
      "accept": ""
    }
  },
  "remoteAddress": "127.0.0.1",
  "remotePort": 59834
},
{
  "msg": "start",
  "time": "2012-03-28T17:39:44.880Z",
  "w": 0
}
```


Based on the example log content, the fields are picked and displayed in the Fields tab, as in the following example:

Number	JSON Path	Name	Data Type	Description
1	\$.name	Application	STRING	Indicates the name or ID of an application in a log entry. Example: Windows event log has an application GUID.
2	\$.hostname	Host Name (Client)	STRING	The hostname of a client host that may be written in a log entry. Example: The client hostname that connects to a web server to access a resource.
3	\$.pid	Please select ...		
4	\$.level	Please select ...		
5	\$.req.method	Please select ...		
6	\$.req.url	Please select ...		
7	\$.req.headers.user-agent	User Agent Client Name	STRING	The client name of an agent that accessed a resource as indicated in a log file. Example: "Mozilla" in a web server access log.
8	\$.req.headers.host	Please select ...		

- In the **Fields** tab, for the specific JSON path, select the field name. The default root path selected is \$. If you want to change the JSON root path, expand the **Advanced Setting** section, and select the **Log Entry JSON Path** from the menu.
- After the fields are selected, go to **Parser Test** tab to view the match status, and the fields picked from the example log content.
- Click **Save** to save the new parser that you just created.

To abort creating a JSON type parser and to switch to creating a parser of regex type, under **Type**, select **Regex**.

Create XML Type Parser

- In the **Parser** field, enter the parser name. For example, enter Database Audit Log - XML.
Provide suitable description to the parser for easy identification.
- In the **Example Log Content** field, paste the contents from a log file that you want to parse, such as the following:

Create Parser

Guided Manual

Define how base fields are extracted from a log record for a given type of log file.

Parser Database Audit log - XML

Type XML

Parser Description

Example Log Content

```
<AuditRecord><Audit_Type>1</Audit_Type><Session_Id>201073</Session_Id><StatementId>13</StatementId><EntryId>6</EntryId>
<Extended_Timestamp>2016-09-09T01:26:07.832814Z</Extended_Timestamp><DB_User>SYSTEM</DB_User><OS_User>
<OS_Process>26839</OS_Process><Terminal>2514</Terminal><Instance_Number>0</Instance_Number><Object_Schema>HR</Object_Schema>
<Object_Name>EMPLOYEES</Object_Name><Action>17</Action>
<TransactionId>0700600A907000</TransactionId><Returncode>0</Returncode><Sql>5703187</Sql><AuthPrivileges>
<Grantee>SUMON</Grantee>
<Priv_Used>244</Priv_Used><DBID>2791516383</DBID><Sql_Text>GRANT SELECT ON hr.employees TO sumon</Sql_Text></AuditRecord>
```

Log Entry XML Path /AuditRecord

Fields Parser Test

Number	XML Path	Name	Data Type	Description
1	/AuditRecord/Audit_Type	Event Type	STRING	Indicates a type of an event from a log entry. This can be textual or numeric value.
2	/AuditRecord/Session_Id	SSO Session ID	STRING	The session identifier of a single sign-on session.
3	/AuditRecord/StatementId	SQL Statement ID	STRING	NO DESCRIPTION
4	/AuditRecord/EntryId	Audit ID	STRING	Identifier of an audit record that this log entry relates to.
5	/AuditRecord/Extended_Timestamp	Time	TIMESTAMP	Time and Date of the log entry. Example: -Jan 1, 2008 13:01:01.000

Based on the example log content, Oracle Log Analytics automatically identifies the list of XML elements that represent the log records.

- From **Log Entry XML Path** menu, select the XML element suitable for the log records of interest. For the above example log content, select /AuditRecord.

Based on the selection of the XML path, the fields are picked and displayed in the Fields tab.

- In the **Fields** tab, select the field name for each entry of the XML path.
- After the fields are selected, go to **Parser Test** tab to view the match status.

Create Parser

Guided Manual

Define how base fields are extracted from a log record for a given type of log file.

Parser Database Audit log - XML

Type XML

Parser Description

Example Log Content

```
<AuditRecord><Audit_Type>1</Audit_Type><Session_Id>201073</Session_Id><StatementId>13</StatementId><EntryId>6</EntryId>
<Extended_Timestamp>2016-09-09T01:26:07.832814Z</Extended_Timestamp><DB_User>SYSTEM</DB_User><OS_User>
<OS_Process>26839</OS_Process><Terminal>2514</Terminal><Instance_Number>0</Instance_Number><Object_Schema>HR</Object_Schema>
<Object_Name>EMPLOYEES</Object_Name><Action>17</Action>
<TransactionId>0700600A907000</TransactionId><Returncode>0</Returncode><Sql>5703187</Sql><AuthPrivileges>
<Grantee>SUMON</Grantee>
<Priv_Used>244</Priv_Used><DBID>2791516383</DBID><Sql_Text>GRANT SELECT ON hr.employees TO sumon</Sql_Text></AuditRecord>
```

Log Entry XML Path /AuditRecord

Fields Parser Test

Match Status	Event Type	SSO Session ID	SQL Statement ID	Audit ID	Time	User Name (Database Client)	User Name
▶	1	201073	13	6	2016-09-09T01:26:07.832Z	SYSTEM	

- Click **Save** to save the new parser that you just created.

To abort creating a XML type parser and to switch to creating a parser of regex type, under **Type**, select **Regex**.

Preprocess Log Events

For certain log sources, such as Database Trace Files and MySQL, Oracle Log Analytics provides functions that allows you to preprocess log events and enrich the resultant log entries.

Currently, Oracle Log Analytics provides the following functions to preprocess log events:

- [Master Detail Function](#)
- [Find Replace Function](#)
- [Time Offset Function](#)

To preprocess log events, click the **Functions** tab and then click the **Add** button while creating a parser.

In the resultant Add Function dialog box, select the required function (based on the log source) and specify the relevant field values.

The screenshot shows the Oracle Management Cloud Log Analytics interface. On the left, the 'Create Parser' dialog is open, with the 'Manual' tab selected. The 'Parser' is named 'jrtest1' and the 'Parse Expression' is '(d+)|s(d+)|s(d+)|s(d+)'. The 'Log Record Delimiter' is set to 'Always one line per entry'. Below this, there is a 'Fields' tab and a 'Functions' tab. The 'Add' button in the 'Functions' tab is highlighted with a red box. On the right, the 'Add Function' dialog is open. It has a 'Name' field with the value 'copy time to rows', a 'Function' dropdown set to 'Find Replace', a 'Catch Expression' field with the value '|<(?<foo>[^\>]+)|>', a 'Find Expression' field with the value '(*)d', and a 'Replace String' field with the value 'foo'. The 'Status' is checked and labeled 'Enabled'. Below these fields is a 'Function Test' section with a text area containing example log content. A red box highlights the 'Test' button. At the bottom, there are two columns: 'Example Log Content' and 'Processed Log Content'. The 'Example Log Content' shows a list of log entries with some highlighted. The 'Processed Log Content' shows the same list with the highlighted parts replaced by the string 'foo'.

To view the result of application of the function on the example log content, under **Function Test** section, click **Test**. A comparative result is displayed to help you determine the correctness of the field values.

Master Detail Function

This function lets you enrich log entries with the fields from the header of log files. This function is particularly helpful for logs that contain a block of body as a header and then entries in the body.

This function enriches each log body entry with the fields of the header log entry. Database Trace Files are one of the examples of these types of logs.

To capture the header and its corresponding fields for enriching the time-based body log entries, at the time of parser creation, you need to select the corresponding **Header Parser** name in the Add Function dialog box.

Examples:

- In these types of logs, the header mostly appears somewhere at the beginning in the log file, followed by other entries. See the following:

```
Trace file /scratch/emga/DB1212/diag/rdbms/lxr1212/lxr1212_1/trace/
lxr1212_1_ora_5071.trc
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Real Application Clusters, Automatic Storage
Management, OLAP,
Advanced Analytics and Real Application Testing options
ORACLE_HOME = /scratch/emga/DB1212/dbh
System name: Linux
Node name: slc00drj
Release: 2.6.18-308.4.1.0.1.el5xen
Version: #1 SMP Tue Apr 17 16:41:30 EDT 2012
Machine: x86_64
VM name: Xen Version: 3.4 (PVM)
Instance name: lxr1212_1
Redo thread mounted by this instance: 1
Oracle process number: 35
Unix process pid: 5071, image: oracle@slc00drj (TNS V1-V3)

*** 2015-10-12 21:12:06.169
*** SESSION ID:(355.19953) 2015-10-12 21:12:06.169
*** CLIENT ID:() 2015-10-12 21:12:06.169
*** SERVICE NAME:(SYS$USERS) 2015-10-12 21:12:06.169
*** MODULE NAME:(sqlplus@slc00drj (TNS V1-V3)) 2015-10-12
21:12:06.169
*** CLIENT DRIVER:() 2015-10-12 21:12:06.169
*** ACTION NAME:() 2015-10-12 21:12:06.169

2015-10-12 21:12:06.169: [ GPNP]clsgpnp_dbmsGetItem_profile: [at
clsgpnp_dbms.c:345] Result: (0) CLSGPNP_OK. (:GPNP00401:)got ASM-
Profile.Mode='legacy'
*** CLIENT DRIVER:(SQL*PLUS) 2015-10-12 21:12:06.290

SERVER COMPONENT id=UTLRP_BGN: timestamp=2015-10-12 21:12:06
*** 2015-10-12 21:12:10.078
SERVER COMPONENT id=UTLRP_END: timestamp=2015-10-12 21:12:10
*** 2015-10-12 21:12:39.209
KJHA:2phase clscrs_flag:840 instSid:
KJHA:2phase ctx 2 clscrs_flag:840 instSid:lxr1212_1
KJHA:2phase clscrs_flag:840 dbname:
KJHA:2phase ctx 2 clscrs_flag:840 dbname:lxr1212
KJHA:2phase WARNING!!! Instance:lxr1212_1 of kspins type:1 does not
support 2 phase CRS
*** 2015-10-12 21:12:39.222
Stopping background process SMCO
*** 2015-10-12 21:12:40.220
ksimdel: READY status 5
*** 2015-10-12 21:12:47.628

...
```

```
KJHA:2phase WARNING!!! Instance:lxrl212_1 of kspins type:1 does not  
support 2 phase CRS
```

For the preceding example, using the Master Detail function, Oracle Log Analytics enriches the time-based body log entries with the fields from the header content.

- Observe the following log example:

```
Server: prodsrv123  
Application: OrderAppA  
  
2017-08-01 23:02:43 INFO DataLifecycle Starting backup process  
2017-08-01 23:02:43 ERROR OrderModule Order failed due to transaction  
timeout  
2017-08-01 23:02:43 INFO DataLifecycle Backup process completed.  
Status=success  
2017-08-01 23:02:43 WARN OrderModule Order completed with warnings:  
inventory on backorder
```

In the preceding example, we have four log entries that must be captured in Oracle Log Analytics. The server name and application name appear only at the beginning of the log file. To include the server name and the application name in each log entry:

1. Define a parser for the header that'll parse the server and application fields:
Server:\s*(\S+).*?Application:\s*(\S+)

Edit Parser
Save Cancel

Define how base fields are extracted from a log record for a given type of log file.

Parser Example Header Parser

Parse Expression

Log Record Delimiter Always one line per entry
 Entry can span multiple lines

Entry Start Expression
 Handle entire file as a single log entry

Example Log Content

Server: prodsrv123
 Application: OrderAppA

 2017-08-01 23:02:43 INFO DataLifecycle Starting backup process
 2017-08-01 23:02:43 ERROR OrderModule Order failed due to transaction timeout

Fields

Functions

Parser Test

Number	Name	Parse Expression	Data Type	Description
1	Host Name (Server) ▾	(\S+)	STRING	The server hostname the log entry is related to. Also could be thought of as source host name for some types of log entries which capture source and destination host names.
2	Application ▾	(\S+)	STRING	Indicates the name or ID of an application in a log entry. Example: Windows event log has an application GUID.

2. Define a second parser to parse the remaining body of the log:

Edit Parser
Save Cancel

Define how base fields are extracted from a log record for a given type of log file.

Parser: Example Body Parser

Parse Expression:

Log Record Delimiter: Always one line per entry
 Entry can span multiple lines

Entry Start Expression:

Handle entire file as a single log entry

Example Log Content:


```
2017-08-01 23:02:43 INFO DataLifecycle Starting backup process
2017-08-01 23:02:43 ERROR OrderModule Order failed due to transaction timeout
2017-08-01 23:02:43 INFO DataLifecycle Backup process completed. Status=success
2017-08-01 23:02:43 WARN OrderModule Order completed with warnings: inventory on
```

Fields

Parser Test

Number	Name	Parse Expression	Data Type	Description
1	Severity	(S+)	STRING	Log severity level of the message. Examples: "DEBUG", "ERROR", "NOTIFICATION", etc.
2	Component	(S+)	STRING	Name of the component this log entry is related to.
3	Message	(.*)	STRING	Primary content message body of the log entry

3. To the body parser, add a Master-Detail function instance and select the header parser that you'd defined in step 1.
4. Add the body parser that you'd defined in the step 2 to a log source, and associate the log source with an entity to start the log collection.

You'll then be able to get four log entries with the server name and application name added to each entry.


Find Replace Function

This function lets you extract text from a log line and add it to other log lines conditionally based on given patterns. For instance, you can use this capability to add missing time stamps to MySQL general and slow query logs.

The find-replace function has the following attributes:

- **Catch Expression:** Regular expression that is matched with every log line and the matched regular expression named group text is saved in memory to be used in Replace Expression.
 - If the catch expression matches a complete line, then the replace expression will not be applied to the subsequent log line. This is to avoid having the same line twice in cases where you want to prepend a missing line.

- A line matched with the catch expression will not be processed for the find expression. So, a find and replace operation cannot be performed in the same log line.
- You can specify multiple groups with different names.
- **Find Expression:** This regular expression specifies the text to be replaced by the text matched by named groups in Catch Expression in log lines.
 - The pattern to match must be grouped.
 - The find expression is not run on those lines that matched catch expression. So, a search and replace operation cannot be performed in the same log line.
 - The find expression can have multiple groups. The text matching in each group will be replaced by the text created by the replace expression.
- **Replace Expression:** This custom notation indicates the text to replace groups found in Find Expression. The group names should be enclosed in parentheses.
 - The group name must be enclosed in brackets.
 - You can include the static text.
 - The text created by the replace expression will replace the text matching in all the groups in the find expression.

Click the **Help**  icon next to the fields **Catch Expression**, **Find Expression**, and **Replace String** to get the description of the field, sample expression, sample content, and the action performed.

Examples:

- The objective of this example is to get the time stamp from the log line containing the text `# Time:` and add it to the log lines starting with `# User@Host` that have no time stamp. Consider the following log data:

```
# Time: 160201 1:55:58
# User@Host: root[root] @ localhost [] Id: 1
# Query_time: 0.001320 Lock_time: 0.000000 Rows_sent: 1
Rows_examined: 1
select @@version_comment limit 1;
# User@Host: root[root] @ localhost [] Id: 2
# Query_time: 0.000138 Lock_time: 0.000000 Rows_sent: 1
Rows_examined: 2
SET timestamp=1454579783;
SELECT DATABASE();
```

The values of the **Catch Expression**, **Find Expression**, and **Replace Expression** attributes can be:

- The **Catch Expression** value to match the time stamp log line and save it in the memory with the name *timestr* is `^(?<timestr>^# Time:.*).`
- The **Find Expression** value to find the lines to which the time stamp log line must be prepended is `(^)# User@Host:.*.`
- The **Replace Expression** value to replace the start of the log lines that have the time stamp missing in them is `(timestr).`

After adding the find-replace function, you'll notice the following change in the log lines:

```
# Time: 160201 1:55:58
# User@Host: root[root] @ localhost [] Id: 1
# Query_time: 0.001320 Lock_time: 0.000000 Rows_sent: 1 Rows_examined: 1
select @@version_comment limit 1;
# Time: 160201 1:55:58
# User@Host: root[root] @ localhost [] Id: 2
# Query_time: 0.000138 Lock_time: 0.000000 Rows_sent: 1 Rows_examined: 2
SET timestamp=1454579783;
SELECT DATABASE();
```

In the preceding result of the example, you can notice that the find-replace function has inserted the timestamp before the User@host entry in each log line that it encountered while pre-processing the log.

- The objective of this example is to catch multiple parameters and replace at multiple places in the log data.

Consider the following log data:

```
160203 21:23:54 Child process "proc1", owner foo, SUCCESS, parent init
160203 21:23:54 Child process "proc2" -
160203 21:23:54 Child process "proc3" -
```

In the preceding log lines, the second and third lines don't contain the user data. So, find-replace function must pick the values from the first line of the log, and replace the values in the second and third lines.

The values of the **Catch Expression**, **Find Expression**, and **Replace Expression** attributes can be:

- The **Catch Expression** value to obtain the `foo` and `init` users info from the first log line and save it in the memory with the parameters `user1` and `user2` is `^.*?owner\s+(?<user1>\w+)\s*,\s*.*?parent\s+(?<user2>\w+)\s*.*`.
- The **Find Expression** value to find the lines that have the hyphenation (-) character is `.*?(-) .*`.
- The **Replace Expression** value is `, owner (user1), UNKNOWN, parent (user2)`.

After adding the find-replace function, you'll notice the following change in the log lines:

```
160203 21:23:54 Child process "proc1", owner foo, SUCCESS, parent init
160203 21:23:54 Child process "proc2", owner foo, UNKNOWN, parent init
160203 21:23:54 Child process "proc3", owner foo, UNKNOWN, parent init
```

In the preceding result of the example, you can notice that the find-replace function has inserted the `user1` and `user2` info in place of the hyphenation (-) character entry in each log line that it encountered while pre-processing the log.

Time Offset Function

Some of the log records will have time stamp missing, some will have only the time offset, and some will have neither. Oracle Log Analytics extracts the required information and assigns the time stamp to each log record.

These are some of the scenarios and the corresponding solutions for assigning the time stamp using the time fset function:

Example Log Content	Scenario	Solution
<pre>Process started May 5, 2017 12:34:53 AM 0.0 host1 debug services started 0.5 host1 debug cache populated 1.4 host1 info connected to cluster 2.7 host1 error cache failure Process started May 6, 2017 12:36:54 AM 0.1 host1 debug services started 0.4 host1 debug cache populated 2.1 host1 info connected to cluster 3.4 host1 error cache failure</pre>	<p>Log file has time stamp in the initial logs and has offsets later.</p>	<p>Pick the time stamp from the initial log records and assign it to the later log records adjusted with time offsets.</p>
<pre>1 [Startup] Timestamp=0, PName=USER_START, ProcessID=11961, ClientIP=xx.xx.xx.xx, Date=20-MAR-2018 03:35:17, ServerName=host2, Tracegroup=debug 10 [CLIENT_TIME] Timestamp=1370, Milliseconds=328 100 [Local_PU.END,6] Timestamp=16000, StartEvent=99, Duration=0</pre>	<p>Log file has initial logs with time offsets and no prior time stamp logs.</p>	<p>Pick the time stamp from the later log records and assign to to previous log records adjusted with time offsets.</p> <p>When the time offset is reset in between, that is, a smaller time offset occurs in a log record, it is corrected by considering the time stamp of the previous log record as reference.</p>

Example Log Content	Scenario	Solution
<pre>0.0 host1 debug services started 0.5 host1 debug cache populated 1.4 host1 info connected to cluster 2.7 host1 error cache failure</pre>	The log file has log records with only time offsets and no time stamp.	<p>time stamp from file's last modified time: After all the log records are traversed, the time stamp is calculated by subtracting the calculated time offset from the file's <i>last modified time</i>.</p> <p>timestamp from filename: When this option is selected in the UI, then the time stamp is picked from the filename in the format as specified by the time stamp expression.</p>

The time offsets of log entries will be relative to previously matched timestamp. We refer to this timestamp as base timestamp in this document.

Use the parser time offset function to extract the time stamp and the time stamp offset from the log records. In the **Parser Function** dialog box:

1. **Name:** Specify a name for the parser function that you're creating.
2. **Function:** Select **Time Offset**.
3. **Where to find the timestamp?:** If you want to specify where the time stamp must be picked from, then select from **Filename**, **File Last Modified Time**, and **Log Entry**. By default, **Filename** is selected.

If this is not specified, then search for the time stamp is performed in the following order:

- Traverse through the log records, and look for a match to the timestamp parser, which you will specify in the next step.
 - Pick the file *last modified time* as the time stamp.
 - If *last modified time* is not available for the file, then select system time as the time stamp.
 - Look for a match to the timestamp expression in the file name, which you will specify in the next step.
4. **Timestamp Expression:** Specify the regex to find the time stamp in the file name. By default, it uses the `{TIMEDATE}` directive.

OR

Timestamp Parser: If you've selected **Log Entry** in step 3, then select the parser from the menu to specify the time stamp format.

5. **Timestamp Offset Expression:** Specify the regex to extract the time offset in seconds and milliseconds to assign the time stamp to a log record. Only the `sec` and `msec` groups are supported. For example, `(?<sec>\d+)\.(?<msec>\d+)`.

Consider the following example log record:

```
15.225 hostA debug services started
```

The example offset expression picks 15 seconds and 225 milliseconds as the time offset from the log record.

- 6. Validate Function:** If you selected **Filename** in step 3, then under **Sample filename**, specify the file name of a sample log that can be used to test the above settings for time stamp expression or time stamp parser, and timestamp offset expression. The log content is displayed in the **Sample Log Content** field.

If you selected **Log Entry** or **File Last Modified Time** in step 3, then paste some sample log content in the **Sample Log Content** field.

- 7.** Click **Validate** to test the settings.

You can view the comparison between the original log content and the computed time based on your specifications.

- 8.** To save the above time offset function, click **Save**.

6

Configure New Log Sources


To monitor a custom log file, you, as the Oracle Log Analytics administrator must create a new log source and a parser.

Topics:

- [Create a Log Source](#)
 - [Use Extended Fields in Log Sources](#)
 - [Use Data Filters in Log Sources](#)
 - [Use Labels in Log Sources](#)
- [Create a Label](#)
- [Create a Field](#)
- [Create Lookups](#)
- [Use the Generic Parser](#)
- [Configure Field Enrichment Options](#)

Create a Log Source

Log sources define the location of your target's logs. A log source needs to be associated with one or more targets to start log monitoring.

1. From Oracle Log Analytics, click the OMC Navigation 
2. In the **Log Sources** section, click **Create source**.

Alternatively, you can click the available number of log sources link in the **Log Sources** section and then in the Log Sources page, click **Create**.

This displays the Create Log Source dialog box.

3. In the **Source** field, enter the name of the log source.
4. From the **Source Type** list, select the type for the log source, such as **File**.

Oracle Log Analytics supports five log source types:

- **File:** Use this type for parsing the majority of log messages supported by Oracle Log Analytics, such as Oracle Database, Oracle Enterprise Manager, Apache, Microsoft, Peoplesoft, Siebel, and so on.
- **Oracle Diagnostic Log (ODL):** Use this type for parsing service-oriented architecture (Oracle SOA Suite) log messages.
- **Syslog Listener:** Use this type for parsing syslog messages (system event messages).
- **Windows Event System:** Use this type for parsing Windows Event Viewer messages. Oracle Log Analytics can collect all historic Windows Event Log entries. It also supports Windows as well as custom event channels.

 **Note:**

If you select this source type, then the **File Parser** field isn't visible.

- **Database:** Use this type for parsing database instance log records, for on-premises as well as autonomous databases.

 **Note:**

- The following steps aren't applicable to database instance log sources. See [Set Up Database Instance Monitoring](#) to learn about how to configure database instance log sources.
- To create a log source for Autonomous Database, see [Set Up Autonomous Database Audit Log Collection](#).

5. Click the **Entity Type** field and select the type of entity for this log source.
 - If you selected **File** or **Oracle Diagnostic Log (ODL)** in step 4, then it's recommended that you select the entity type for your log source that most closely matches what you are going to monitor. Avoid selecting composite entity types like Database Cluster and instead select the entity type Database Instance because the logs are generated at the instance level.
 - If you selected the source type **Syslog Listener** in step 4, then select one of the variants of Host such as Host (Linux), Host (Windows), Host (AIX), or Host (Solaris) as your entity type. This is the host on which the agent is running and collecting the logs. The syslog listener is configured to receive the syslog logs from instances that might not be running on the same host. However, the agent that's installed on the syslog listener host collects those logs for which the listener is configured to collect.
 - If you selected the source type **Database** in step 4, then the entity type is limited to the eligible database types.
 - If you selected **Windows Event System** source type, then the default entity type Host (Windows) is automatically selected, and cannot be changed.

If you install a cloud agent, gateway agent, or data collection agent and enable or disable the log collection in the OMC agent management console, then the corresponding log sources of the agent also get enabled and disabled. Avoid creating the custom log sources of one of the Agent entity types.

6. Click the **File Parser** field and select the relevant parser name such as **Database Audit Log Entries Format**.

You can select multiple file parsers for the log files. This is particularly helpful when a log file has entries with different syntax and can't be parsed by a single parser. For **ODL** source type, the only parser available is **Oracle Diagnostic Logging Format**.

The File Parser field isn't available for **Windows Event System** source type. Oracle Log Analytics parsers are based on regular expressions. For the **Windows Event System** source type, Oracle Log Analytics retrieves already parsed log data. So, you don't need any parser for **Windows Event System** logs.

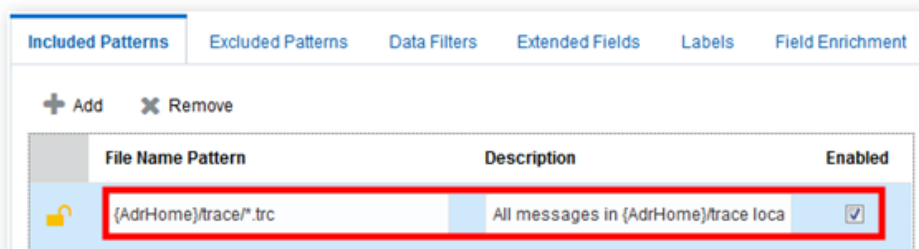
The **Author** field already has your user name.

7. To automatically associate this log source with all matching entity types, select the **Auto-Associate** check box.
8. In the **Included Patterns** tab, click **Add** to specify file name patterns for this log source.

Enter the file name pattern and description.

You can enter parameters within braces {}, such as {AdrHome}, as a part of the file name pattern. Oracle Log Analytics replaces the parameters with the actual value at runtime. You can view all the parameters for a particular target type by clicking **See all available built-in parameters**.

The log source contains only those log files that match the included patterns.

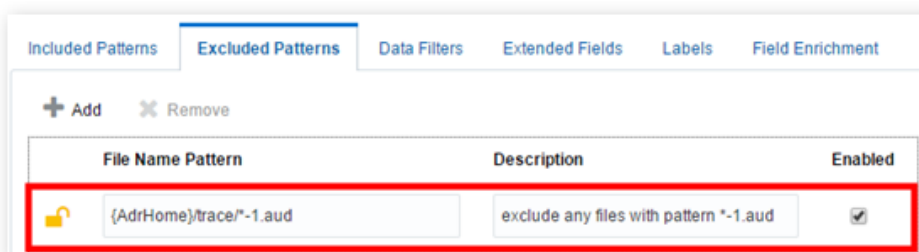


You can configure warnings in the log collection for a given pattern.

Select the **Send Warning** checkbox. In the adjacent drop-down list, select the situation in which the warning must be issued:

- **Every missing or unreadable pattern**
 - **All patterns are unreadable**
9. In the **Excluded Patterns** tab, click **Add** to define patterns of log file names that must be excluded from this log source.

For example, you can use an excluded pattern when there are files in the same location that you don't want to include in the log source definition. For example, there's a file with the name `audit.aud` in the directory `/u01/app/oracle/admin/rdbms/diag/trace/`. In the same location, there's another file with the name `audit-1.aud`. You can exclude any files with the pattern `*-1.aud`.




10. Click **Save**.

Use Extended Fields in Log Sources

The Extended Fields feature in Oracle Log Analytics lets you extract additional fields from a log entry, in addition to the fields defined by the out-of-the-box parsers.

By default, analyzing log content using a log source extracts the fields that are defined in the base parser. A base parser extracts common fields from a log entry. However, if you have a requirement to extract additional fields, then you can use the extended fields definition. For example, a base parser may be defined such that the last part of a log entry that starts with an alpha character must be displayed as the value of the Message field. If you need to parse the Message field further to extract additional fields from within the value of the Message field, then you use the Extended Fields feature to update the log source definition and define additional extended fields.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the Oracle Log Analytics Configuration page, click the count of available log sources link.
3. In the Log Sources page, select the required log source where you want to define the extended fields and click **Edit**.
4. Click the **Extended Fields** tab and then click **Add**.
5. To add a condition to the extended field, expand the **Conditions** section.
 - **Reuse Existing:** To reuse a condition that's already defined for the log source, enable the **Reuse Existing** button, and select the condition from the **Condition** menu.
 - **Create New Condition:** Enable this button if you want to define a new condition. Specify the **Condition Field**, **Operator**, and **Value**.
6. Select the **Base Field** whose value you want to extract and display as an extended field.
7. Enter an example of the value that would be extracted in the **Example Base Field Content** field.
8. Enter the extraction expression in the **Extraction Expression** field and select **Enabled** check box.

Examples:

- To extract the endpoint file name from the URI field of a Fusion Middleware Access log file, enter the following:
 - Base Field: **URI**
 - Example Content: `/service/myService1/endpoint/file1.jpg`
 - Extended Field Extraction Expression: `{Content Type:\.(jpg|html|png|ico|jsp|htm|jspx)}`
- To extract the user name from the file path of a log entity, enter the following:
 - Base Field: **Log Entity**
 - Example Content: `/u01/oracle/john/audit/134fa.xml`
 - Extended Field Extraction Expression: `/\w+/\w+/{User Name:\w+}/`

- To extract the timestamp as well as the log entry time from the following data:

```
2018-11-14T23:23:12.324Z INFO Backup transaction finished.
Start=1542111920
```

Use the following parser expression and extended field extraction expression:

- Parser: `{TIMEDATE}\s(\w+)\s(.*)`
- Extended Field Extraction Expression: `Start={Event Start Time:\d+}`

Oracle Log Analytics supports epoch seconds and milliseconds for the timestamp fields. Note that the expression `{TIMEDATE}` can be only used for log entry time.

- Click **Test** to determine the status of match of the extract expression with the example base field content. In case of success in the match, the **Step Count** is displayed which is the good measure of the effectiveness of the extract expression. If the expression is inefficient, then the parsing may timeout, thus resulting in the Extract Field Expression not getting considered for log parsing.

The following is a simple example of using the test feature for testing the effectiveness of the extract expression:

The screenshot shows the 'Create Extended Field Definition' dialog box. It has two main sections: 'Conditions (Optional)' and 'Extended Fields'. In the 'Conditions' section, 'Reuse Existing' is selected, and the condition is 'Action In (USER_Change,USER_Create,USER_Remove,USER_SU)'. In the 'Extended Fields' section, the 'Base Field' is 'Action', the 'Example Base Field Content' is 'user John.Doe logged in', and the 'Extract Expression' is 'user (User Name:\w+)'. A 'Test' button is located to the right of the 'Extract Expression' field. Below the dialog, a 'Match Status' table shows a successful match for 'John' with a 'Step Count' of 10.

Match Status	User Name
✔	John

Details Match Succeeded
Step Count 10
Example Base Field Content user John.Doe logged in
Extract Expression user (User Name:\w+)

- Click **Save**.

If you use automatic parsing that only parses time, then the extended field definition is based on the **Original Log Content** field, because that's the only field that will exist in the log results. See [Use the Generic Parser](#).

When you search for logs using the updated log source, values of the extended fields are displayed along with the fields extracted by the base parser.

Oracle Log Analytics enables you to search for the extended fields that you're looking for. You can search based on the how it was created, the type of base field, or with some example content of the field. Enter the example content in the **Search** field, or click the down arrow for the search dialog box. In the search dialog box, under **Creation Type**, select if the extended fields that you're looking for are out-of-the-box or user-defined. Under **Base Field**, you can select from the options available. You can also specify the example content or the extraction field expression that can be used for the search. Click **Apply Filters**.

Table 6-1 Sample Example Content and Extended Field Extraction Expression

Log Source	Parser Name	Base Field	Example Content	Extended Field Extraction Expression
/var/log/messages	Linux Syslog Format	Message	authenticated mount request from 10.245.251.222:735 for /scratch (/scratch)	authenticated {Action:\w+} request from {Address:[\d\.]+}:{Port:\d+} for {Directory:\S+} \s(
/var/log/yum.log	Yum Format	Message	Updated: kernel-headers-2.6.18-371.0.0.0.1.el5.x86_64	{Action:\w+}:{Package:.*)
Database Alert Log	Database Alert Log Format (Oracle DB 11.1+)	Message	Errors in file /scratch/cs113/db12101/diag/rdbms/pteintg/pteintg/trace/pteintg_smon_3088.trc (incident=4921): ORA-07445: exception encountered: core dump [sentimedop()+10] [SIGSEGV] [ADDR:0x16F9E00000B1C] [PC:0x7FC6DF02421A] [unknown code] []	Errors in file {Trace File:\S+} (incident={Incident:\d+}): {Error ID:ORA-\d+}: exception encountered: core dump [sentimedop()+10] [SIGSEGV] [ADDR:{Address:[\w\d]+}] [PC:{Program Counter:[\w\d]+}] [unknown code] []

Table 6-1 (Cont.) Sample Example Content and Extended Field Extraction Expression


Log Source	Parser Name	Base Field	Example Content	Extended Field Extraction Expression
FMW WLS Server Log	WLS Server Log Format	Message	Server state changed to STARTING	Server state changed to {Status:\w+}

Use Data Filters in Log Sources

Oracle Log Analytics lets you mask and hide sensitive information from your log records as well as hide entire log entries before the log data is uploaded to the cloud. Using the **Data Filters** tab under **Log Sources** in the **Configuration** page, you can mask IP addresses, user ID, host name, and other sensitive information with replacement strings, drop specific keywords and values from a log entry, and also hide an entire log entry.

Masking Log Data

If you want to mask information such as the user name and the host name from the log entries:

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click **Create source**.
Alternatively, you can click the available number of log sources link in the **Log Sources** section and then in the Log Sources page, click **Create**.
This displays the Create Log Source dialog box.
3. Specify the relevant values for the **Source**, **Source Type**, **Entity Type**, and **File Parser** fields.
4. In the **Included Patterns** tab, click **Add** to specify file name patterns for this log source.
5. Click the **Data Filters** tab and click **Add**.
6. Enter the mask **Name**, select **Mask** as the **Type**, enter the **Find Expression** value, and its associated **Replace Expression** value.

Name	Find Expression	Replace Expression
mask username	User=(\S+)s+	confidential
mask host	Host=(\S+)s+	mask_host

 **Note:**

The syntax of the replace string should match the syntax of the string that's being replaced. For example, a number shouldn't be replaced with a string. An IP address of the form `ddd.ddd.ddd.ddd` should be replaced with `000.000.000.000` and not with `000.000`. If the syntaxes don't match, then the parsers will break.

7. Click Save.

When you view the masked log entries for this log source, you'll find that Oracle Log Analytics has masked the values of the fields that you've specified.

- User = confidential
- Host = mask_host

 **Note:**

Apart from adding data filters when creating a log source, you can also edit an existing log source to add data filters. See [Manage Existing Log Sources](#) to learn about editing existing log sources.


 **Note:**


Data masking works on continuous log monitoring as well as on syslog listeners.

Hash Masking the Log Data

When you mask the log data using the mask as described in the previous section, the masked information is replaced by a static string provided in the Replace Expression. For example, when the user name is masked with the string `confidential`, then the user name is always replaced with the expression `confidential` in the log records for every occurrence. By using hash mask, you can hash the found value with a unique hash. For example, if the log records contain multiple user names, then each user name is hashed with a unique expression. So, if `user1` is replaced with the text hash `ebdkromlucea9ie` for every occurrence, then the hash can still be used for all analytical purposes.

To apply the hash mask data filter on your log data:

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click **Create source**. Alternatively, you can click the available number of log sources link in the **Log Sources** section and then in the Log Sources page, click **Create**. This displays the Create Log Source dialog box.

You can also edit a log source that already exists. In the Log Sources page, click  next to your log source, and click **Edit**. This displays the Edit Log Source dialog box.

3. Specify the relevant values for the **Source**, **Source Type**, **Entity Type**, and **File Parser** fields.
4. In the **Included Patterns** tab, click **Add** to specify file name patterns for this log source.
5. Click the **Data Filters** tab and click **Add**.
6. Enter the mask **Name**, select **Hash Mask** as the **Type**, enter the **Find Expression** value, and its associated **Replace Expression** value.

Name	Find Expression	Replace Expression
Mask User Name	User=(\S+)s+	Text Hash
Mask Port	Port=(\d+)s+	Numeric Hash

7. Click **Save**.

As the result of the above example hash masking, each user name is replaced by a unique text hash, and each port number is replaced by a unique numeric hash.

You can extract the hash masked log data using the hash for filtering. See [Filter Logs by Hash Mask](#).

Dropping Specific Keywords or Values from Your Log Records

Oracle Log Analytics lets you search for a specific keyword or value in log records and drop the matched keyword or value if that keyword exists in the log records.

Consider the following log record:

```
ns5xt_119131: NetScreen device_id=ns5xt_119131 [Root]system-
notification-00257(traffic): start_time="2017-02-07 05:00:03" duration=4
policy_id=2 service=smtp proto=6 src zone=Untrust dst zone=mail_servers
action=Permit sent=756 rcvd=756 src=249.17.82.75 dst=212.118.246.233
src_port=44796 dst_port=25 src-xlated ip=249.17.82.75 port=44796 dst-xlated
ip=212.118.246.233 port=25 session_id=18738
```

If you want to hide the keyword `device_id` and its value from the log record:

1. Perform Step 1 through Step 5 listed in the Masking Log Data section.
2. Enter the filter **Name**, select **Drop String** as the **Type**, and enter the **Find Expression** value such as `device_id=\S*`.
3. Click **Save**.

When you view the log entries for this log source, you'll find that Oracle Log Analytics has dropped the keywords or values that you've specified.



Note:

Ensure that your parser regular expression matches the log record pattern, otherwise Oracle Log Analytics may not parse the records properly after dropping the keyword.

 **Note:**

Apart from adding data filters when creating a log source, you can also edit an existing log source to add data filters. See [Manage Existing Log Sources](#) to learn about editing existing log sources.

Dropping an Entire Line in a Log Record Based on Specific Keywords

Oracle Log Analytics lets you search for a specific keyword or value in log records and drop an entire line in a log record if that keyword exists.

Consider the following log record:

```
ns5xt_119131: NetScreen device_id=ns5xt_119131 [Root]system-  
notification-00257(traffic): start_time="2017-02-07 05:00:03" duration=4  
policy_id=2 service=smtp proto=6 src zone=Untrust dst zone=mail_servers  
action=Permit sent=756 rcvd=756 src=249.17.82.75 dst=212.118.246.233  
src_port=44796 dst_port=25 src-xlated ip=249.17.82.75 port=44796 dst-  
xlated ip=212.118.246.233 port=25 session_id=18738
```

Let's say that you want to drop entire lines if the keyword `device_id` exists in them:

1. Perform Step 1 through Step 5 listed in the Masking Log Data section.
2. Enter the filter **Name**, select **Drop Log Entry** as the **Type**, and enter the **Find Expression** value such as `.*device_id=.*`.
3. Click **Save**.

When you view the log entries for this log source, you'll find that Oracle Log Analytics has dropped all those lines that contain the string `device_id` in them.


 **Note:**

Apart from adding data filters when creating a log source, you can also edit an existing log source to add data filters. See [Manage Existing Log Sources](#) to learn about editing existing log sources.

Use Labels in Log Sources

Oracle Log Analytics lets you add labels or tags to log entries, based on defined conditions.

You can use patterns to specify a condition. When a log entry matches that condition, the label associated with the pattern is displayed alongside the log entry.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the Oracle Log Analytics Configuration page, click the count of available log sources link.
3. In the Log Sources page, select the required log source where you want to define the extended fields and click **Edit**.

4. Click the **Labels** tab and then click **Add**.
5. Select the log field on which you want to apply the condition from the **Field** list.
6. Select the operator from the **Operator** list.
7. In the **Condition Value** field, specify the value of the condition to be matched for applying the label.
8. In the **Label** field, enter the text for the label to be applied and select the **Enabled** check box.
9. Select the output field.

Click the **Edit**  icon. The **Pick Output Field** dialog box opens.

10. Pick the **Output Field** by specifying the label to be used or by selecting from any other field. Click **Apply**.

In the following image, the log source has been configured to attach the `authentication.login` output value for the `Security Category` output field when the log entry contains the input field `Method` set to the value **CONNECT**.

You can also create a custom label to tag a specific log entry. See [Create a Label](#).

11. Click **Save**.

Oracle Log Analytics enables you to search for the labels that you're looking for. You can search based on any of the parameters defined for the labels. Enter the search string in the **Search** field, or click the down arrow for the search dialog box. You can specify the search criteria in the search dialog box. Under **Creation Type**, select if the labels that you're looking for are out-of-the-box or user-defined. Under the fields **Input Field**, **Operator**, and **Output Field**, you can select from the options available. You can also specify the condition value or the output value that can be used for the search. Click **Apply Filters**.

You can now search log data based on the labels that you've created. See [Filter Logs by Labels](#).

You can also use the labels to enrich the data set instead of creating a lookup table for a one time operation, as in the following example:



In this example, if the input field `Action` has the value `46`, then the output field `Event` is loaded with the value `delete_file`.

Create a Label

Oracle Log Analytics offers multiple out-of-the-box labels for select log sources. You can use these labels to tag the log entries in your log sources. In the following `Cisco ASA Logs` log source example, the highlighted out-of-the-box labels and more are provided by Oracle Log Analytics for ready use.

However, if you can't find the labels that you're looking for, create custom labels that can be used in log sources to tag the log entries, based on defined conditions. To create a label:

1. From Oracle Log Analytics, click the OMC Navigation (☰) icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Labels** section, click **Create Label**.
Alternatively, in the **Labels** section, you can click the available number of labels link and then in the Labels page, click **Create**.
3. In the **Label** field, enter the label name. For example, enter `Gateway Timeout`.
4. In the **Description** field, enter the details about the label. For example, enter `Java exception encountered`.
5. To assign priority to the label:
 - a. Under **Denotes Problem** field, select **Yes** check box.
 - b. In the **Problem Priority** field, click the down arrow and select a priority. For example, select `High`.

If you want to be able to filter the logs in the Log Explorer using the **Show Problem Logs Only** option, then you must assign a priority level to the problem. The problem logs are determined in the search based on the label definition of the log source which has the problem priority set. If the priority is not set, then a potential problem log may not be marked as one.

6. In the **Related Terms** field, enter the terms that are related to the log entry.
7. Click **Save**.

You can now use the new custom label in your log source to tag a log entry. See [Use Labels in Log Sources](#).

After the custom labels are associated with the log sources, you can search the log data based on the labels that you've created. See [Filter Logs by Labels](#). The labels can be used for search as in the following example use-cases:

- To obtain rapid summary of all error trends:

Label (12 selected)

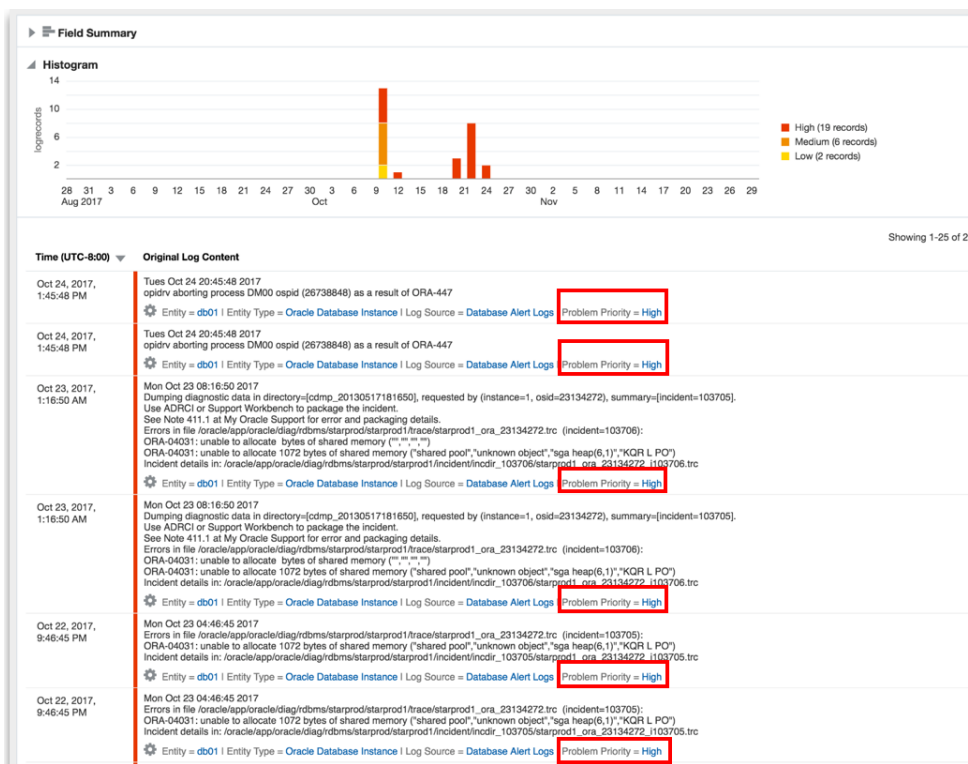
Search Selected

<input type="checkbox"/>	Label	Problem Priority	Count	Trend
<input type="checkbox"/>	HTTP Error	Low	7	
<input type="checkbox"/>	Memory Error	High	6	
<input type="checkbox"/>	Abnormal Termination	High	4	
<input type="checkbox"/>	Authentication Error	Medium	3	
<input type="checkbox"/>	Deadlock	High	2	
<input type="checkbox"/>	Storage Error	High	2	
<input type="checkbox"/>	Availability Error	High	2	
<input type="checkbox"/>	Security Warning	High	2	

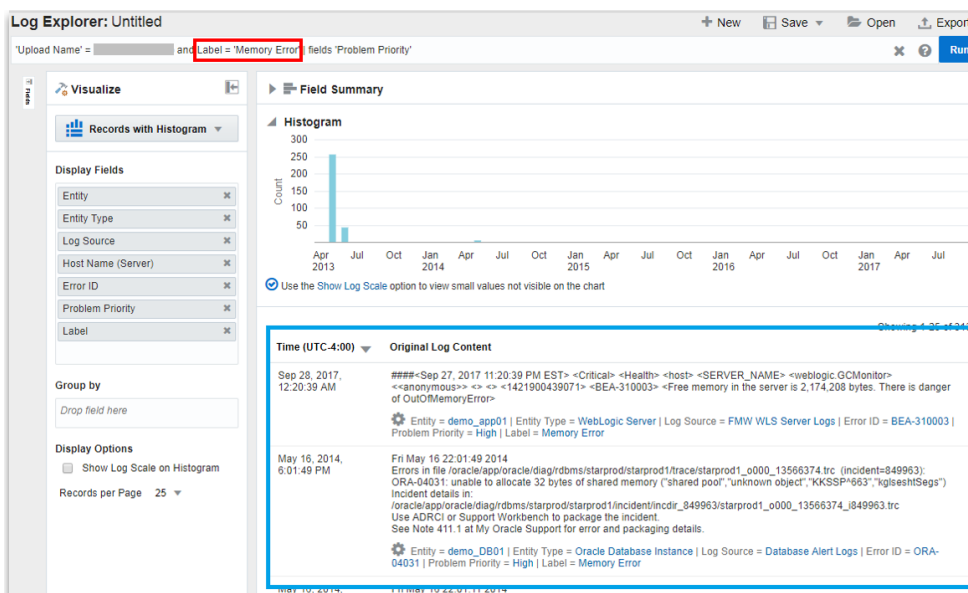
Tip: Matching log entries may have multiple values which will be included in the selected list.

Show Trend Chart | (1-12 of 12 items)

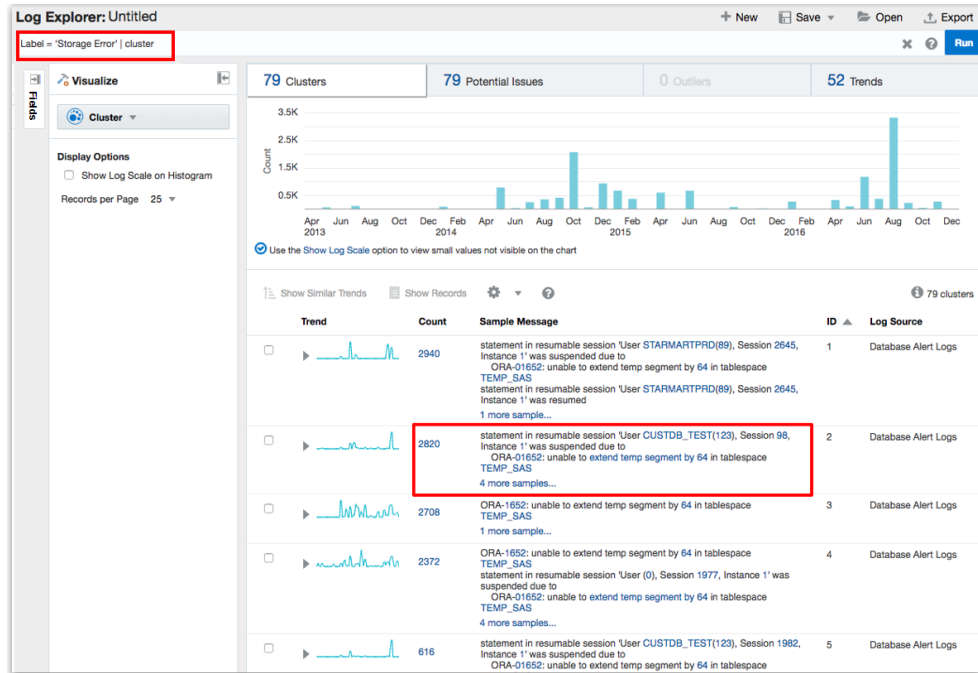
- To identify the problem events:



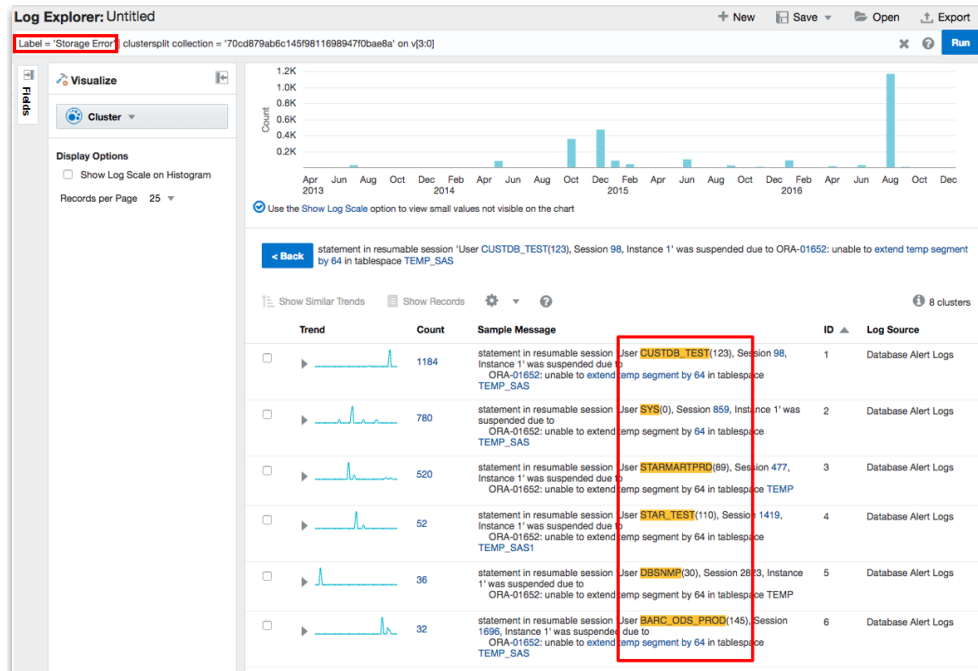
- To perform plain language analysis across log sources:



- To perform plain language analysis in combination with clusters:




View the log data within the cluster for classify results:



Create a Field

Oracle Log Analytics offers multiple out-of-the-box fields for parsers. You can use these fields to associate with the parse expressions. However, if you can't find the right

field names that you're looking for, create custom fields that can be used to associate with parse expressions.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Other Links** section, click **Fields**.
The fields page is displayed. It lists all the fields available in Oracle Log Analytics.
3. In the Fields page, click **Create**.
4. In the Create Field page, in the **Name** field, enter the name of the field you want to create. For example, enter `Correlation ID`.
5. In the **Type** select the type of field data. For example, select `String`.
6. If the field can have multiple values in the log content, then select the **Multi value** check box.
7. In the **Description** field, enter the description of field. This description can help you to identify the field in the Fields page.

You can now use the new custom field to associate with the parse expression. See [Create a Parser](#).

After the custom field is created, you can use it in the log explorer for filtering and searching. See [Filter Logs by Pinned Attributes and Fields](#).

You can also use the field for visualizing and analyzing the log data using charts and controls. See [Visualize Data Using Charts and Controls](#).

Create Lookups

Using Oracle Log Analytics, you can enrich event data by adding field-value combinations from lookups. Oracle Log Analytics uses lookups to match field-value combinations from events to an external lookup table, and if matched, Oracle Log Analytics appends the field-value combinations to the events.

For example, the Error ID field in log events doesn't provide a description of the errors. You can create a lookup that maps Error ID to descriptions, and then use the Field Enrichment options to make the descriptions available to search or visible in the log records. Lookup data can be of two types, *Lookup* or *Dictionary*. Lookup type requires that the content is defined as a set of comma separated values. These values can then be obtained by associating with a log field. Dictionary type also requires that the content is defined as a set of comma separated values, but the actual lookup is performed as an action defined by the *Operator* field in the file.

After you create a lookup, you can use it as a Field Enrichment option in your log source. See [Configure Field Enrichment Options](#).

The *Lookup* type data can be associated with the log events while ingesting logs or for analyzing logs using the query language.

In case of a *Dictionary* type lookup, you can use the data for analyzing logs using the query language only after the use of *cluster* or *link* commands. This type of lookup cannot be used for ingesting logs.


Create a CSV Lookup

After creating the CSV lookup, use *searchlookup* command to list the lookups. Use the *lookup* command to map to the fields with any query.

1. Create a lookup CSV file with the field-value combinations. For example, to create a lookup that maps Error ID to descriptions:

```
errid,description
02323,Network Not Reachable
09912,User Activity
12322,Out of Memory
```

Note that the first row is the header with `errid` and `description` titles for the values in the subsequent rows.

2. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
3. Under **Lookups**, click the **Create Lookup** link .
4. In the Lookup page, enter the name of the lookup, such as `server error code lookups` and an optional description.
5. Under **Type**, select **Lookup**.
6. Click **Import**, select the lookup CSV file that you had created earlier, and click **Save**.

Create a Dictionary Lookup

After creating the Dictionary type lookup, use *searchlookup* command to list the lookups. Use the *lookup* command to map to the fields with any query only after using the *link* or *cluster* commands in the query.

1. Create a lookup CSV file with the field-value combinations. For example:

```
Operator,Condition,Issue,Area
CONTAINS,message header or abbreviation processing
failed,Processing Error,Messaging
CONTAINS,Failed to associate the transaction context with the
response while marshalling,Marshalling Error,Response
CONTAINS,A RuntimeException was generated by the RMI
server,Exception,RMI
```

Note that the first row is the header where `Operator` and `Condition` are the mandatory parameters specified in the same order. The subsequent parameters are listed in the header row after the mandatory parameters. The subsequent rows are the values of the parameters listed in the header row in the same order.

For the list of valid operators and examples to use them, see the sections [List of Valid String Operators and Examples](#), [List of Numeric and Logical Operators and Examples](#), and [Use Comments While Defining Dictionary Lookups](#).

Note:

- If a field contains a comma, enclose the entire field in double quotes.
 - If a field contains double quotes, escape the double quote by using two double quotes.
 - A dictionary must define either all string operators or only numerical operators. The numerical operators must not be mixed with string operators in the same dictionary lookup.
2. From Oracle Log Analytics, click the OMC Navigation (☰) icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
 3. Under **Lookups**, click the **Create Lookup** link .
 4. In the Lookup page, enter the name of the lookup, such as `cluster dictionary lookups` and an optional description.
 5. Under **Type**, select **Dictionary**.
 6. Click **Import**, select the lookup CSV file that you had created earlier, and click **Save**.

For examples of using dictionary lookup in Cluster and Link, see [Use Dictionary Lookup in Cluster](#) and [Use Dictionary Lookup in Link](#).

CIDRMATCH Operator

The `CIDRMATCH` operator supports CIDR (Classless Inter-Domain Routing) match operation rule inside a dictionary lookup. For example, the following dictionary would return *Network Name* as *Database Network* if the input IP Address falls in the range between 192.0.2.0 and 192.0.2.255:

```
Operator,Condition,Network Name
CIDRMATCH,192.0.2.10/24,Database Network
```

List of Valid String Operators and Examples

Operator	Description	Example
CONTAINS	True if the value contains the string specified in the Condition field. Case-sensitive	CONTAINS,Request 'GetResponse' Timed out,Timeout Error
CONTAINS IGNORE CASE	Same as above, except the case is ignored	CONTAINS IGNORE CASE,request 'getresponse' timed out,Timeout Error
CONTAINS REGEX	True if the value matches the specified regular expression	CONTAINS REGEX,Request '\S+' Timed out,Timeout Error
CONTAINS IGNORE CASE REGEX	Same as above, but ignores the case	CONTAINS IGNORE CASE REGEX,request '\S+' timed out,Timeout Error
CONTAINS MULTILINE REGEX	Use this to match against a multi-line string	CONTAINS MULTILINE REGEX,Request 'GetResponse' Timed out,Timeout Error
CONTAINS IGNORE CASE MULTILINE REGEX	Same as above, except the case is ignored	CONTAINS IGNORE CASE MULTILINE REGEX,Request 'GetResponse' Timed out,Timeout Error

Operator	Description	Example
CONTAINS ONE OF REGEXES	Specify more than one regular expression. True if at least one matches. List the regular expressions inside [] and separate by comma. The regular expressions cannot contain a comma. If you need to use double quotes inside the regex, escape each double quote using another double quote.	CONTAINS ONE OF REGEXES,"[Request '\S+' Timed out,Server\S+Timed out]",Timeout Error
NOT CONTAINS	Does not contain the specified string	NOT CONTAINS,Request 'GetResponse' Timed out,Success
EQUAL	Content equals the specified value	EQUAL,500,HTTP Server Error
EQUAL IGNORE CASE	Same as above, except the case is ignored	EQUAL,In-Progress,Request In Progress
NOT EQUAL	True if the content is not equal to the value specified	NOT EQUAL,200,HTTP Request Failed
STARTS WITH	Compares to the beginning of the content	STARTS WITH,Request failed with,Fail
ENDS WITH	Compares to the end of the content	ENDS WITH,timed out,Timeout
IN	True if at least one of the value is equal	IN,"[500,501,502,503]",HTTP Server Error
IN IGNORE CASE	Same as above, except the case is ignored	IN IGNORE CASE,[fail,timeout,error,fatal],Request Failed
NOT IN	True if the content is not equal to any value in the list	NOT IN,"[500,501,503,400,401,404]",HTTP Request Successful
NULL	True if the content in field is null	NULL,,No Value
NOT NULL	True if the content in field is not null	NOT NULL,,Value Present

List of Numeric and Logical Operators and Examples

Operator	Description	Example
=	Numerical Equal To	=,1,Value is 1
!=	Numerical Not Equal To	!=,1,Value is Not 1
>	Above the given value	>,1,Value is above 1
<	Below the given value	<,1,Value is below 1
>=	Above or equal to the given value	>=,1,Value is equal or above 1
<=	Below or equal to the given value	<=,1,Value is equal or below 1
BETWEEN	Between the given two values, both inclusive	BETWEEN,1-10,Value is equal or above 1 and equal or below 10

Operator	Description	Example
> AND <	Above N1 and Below N2	> AND <,1-10,Above 1 and below 10
>= AND <=	Same as Between. Above or equal to N1 and Below or equal to N2	>= AND <=,1-10,Above or equal to 1 and below or equal to 10
>= AND <	Above or equal to N1 and Below N2	>= AND <,1-10,Above or equal to 1 and below 10
> AND <=	Above N1 and Below or equal to N2	> AND <=,1-10,Above 1 and below or equal to 10
> OR <	Above N1 or Below N2	> OR <,1-10 Above 1 or below 10
>= OR <=	Above or equal to N1 or Below or equal to N2	>= OR <=,100-10,Above or equal to 100 or below or equal to 10
>= OR <	Above or equal to N1 or Below N2	>= OR <,10-1,Above or equal to 10 or below 1
> OR <=	Above N1 or Below or equal to N2	> OR <=,100-10,Above 100 or below or equal to 10
>= OR !=	Above or equal to N1 or not equal to N2	>= OR !=,10-1,Above or equal to 10 or not equal to 1
<= OR !=	Below or equal to N1 or not equal to N2	<= OR !=,10-100,Below or equal to 10 or not equal to 100
>= OR =	Above or equal to N1 or equal to N2	>= OR =,10-1,Above or equal to 1 or equal to 1
<= OR =	Below or equal to N1 or equal to N2	<= OR =,10-100,Below or equal to 10 or equal to 100
> AND !=	Above N1 and not equal to N2	> AND !=,10-100,Above 10 and not equal to 100
< AND !=	Below N1 and not equal to N2	< AND !=,10-1,Below 10 and not equal to 1

Use Comments While Defining Dictionary Lookups

Use # as the first field to add comments to a dictionary lookup. Following is an example of a sample lookup with comments:


```
Operator,Condition,Label,Module
# -----
# Startup/Shutdown and Terminations
# -----
CONTAINS,Server started in RUNNING mode,Server Started,WebLogic Server
```

```
CONTAINS,A critical service failed. The server will shut itself
down,Server Shutdown,WebLogic Server
CONTAINS,state changed to FAILED,Server Failed,
CONTAINS,Removing .* from cluster view due to PeerGone,Cluster
Removed,WebLogic Server
# -----
# Connection Error / Timeouts and Slowness
# -----
CONTAINS,Unable to connect to WSM policy manager,WSM Policy Manager
Connection Error,
CONTAINS REGEX,java.sql.SQLException: \S+: user requested
cancel of current operation,SQL Timeout,Database
CONTAINS,This member is running extremely slowly and may endanger the
rest of the cluster,WebLogic Cluster Slowness,WebLogic Server
```

Use the Generic Parser

Oracle Log Analytics lets you configure a generic parser to parse logs from different log sources.

This is particularly helpful when you're not sure about how to parse your logs or how to write regular expressions to parse your logs, and you just want to pass the raw log data to perform analysis. Typically, a parser defines how the fields are extracted from a log entry for a given type of log file. However, the generic parser in Log Analytics can:

- Detect the time stamp and the time zone from log entries.
 - Create a time stamp using the current time if the log entries don't have any time stamp.
 - Detect whether the log entries are multiple lined or single lined.
1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
 2. In the **Log Sources** section, click **Create source**.
Alternatively, in the **Log Sources** section, you can click the available number of log sources link and then in the Log Sources page, click **Create**.
This displays the Create Log Source dialog box.
 3. In the **Source** field, enter the name for the log source.
 4. In the **Source Type** field, select **File**.
 5. Click **Target Type** and select the type of target for this log source.
 6. Select **Automatically parse time only**. Oracle Log Analytics automatically applies the generic parser type.
 7. To automatically associate this log source with all matching target types, select the **Auto-Associate** check box.
 8. Click **Save**.

When you access the log entries of the newly created log source, Oracle Log Analytics extracts and displays the following information from the log entries:

- Time stamp:

- When a log entry doesn't have a time stamp, then the generic parser creates and displays the time stamp based on the time when the log data was collected.
- When a log entry contains a time stamp, but the time zone isn't defined, then the generic parser uses the cloud agent's time zone.
- Time zone:
 - When a log file has log entries with multiple time zones, the generic parser can support up to 11 time zones.
 - When a log displays some entries with a time zone and some without a time zone, then the generic parser follows the time zone of the latest log entry.

Time (UTC+8:00) ▼	Original Log Content
Dec 15, 2014 09:46:00.000 PM	Monday, Dec 15, 2014 1:46 PM This is the 3rd test log followed Target = slc09see.us.oracle.com Target Type = Host Log Source = gp
Dec 07, 2014 05:51:23.000 AM	Sat Dec 6 21:51:23 2014 This is the 2nd log followed Target = slc09see.us.oracle.com Target Type = Host Log Source = gp
Dec 04, 2014 03:01:08.044 PM	2014-12-04T07:01:08.044321Z This is the 1st log followed Target = slc09see.us.oracle.com Target Type = Host Log Source = gp
Apr 18, 2014 04:41:30.000 AM	Tue Apr 17 16:41:30 EDT 2014 timezone EDT Target = slc09see.us.oracle.com Target Type = Host Log Source = gp

If the time zone or the time zone offset is not indicated in the log events, then Oracle Log Analytics compares the last modified time of the OS with the timestamp of the last log entry to determine the proper time zone.

- Multiple lines: When a log entry spans multiple lines, the generic parser can capture the multiline content correctly.

Configure Field Enrichment Options


Oracle Log Analytics lets you configure Field Enrichment options so you can further extract and display meaningful information from your extended fields data.

One of the Field Enrichment options is the Geolocation Lookup that converts IP addresses or host names present in the log records to a country or country code. This can be used in log sources like Web Access Logs that have external client IP addresses.

Using the Lookup Field Enrichment option, you can match field-value combinations from events to an external lookup table.

Geolocation Lookup


After you set up the Geolocation Lookup options, you can view log records grouped by country or country code using the maps visualization.

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the Oracle Log Analytics Configuration page, click the count of available log sources link.
3. In the Log Sources page, select the required log source where you want to define the extended fields and click **Edit**.
4. Add the Extended Fields definition for the base field that contains the country-specific IP address or host names records, such as Host IP Address.
5. Click the **Field Enrichment** tab and then click **Add**.
6. In the Field Enrichment dialog box, select **Geolocation Lookup** as the **Function**.
Click the **View details** link to see a sample representation of the Geolocation Lookup function.
7. Keep the **Enabled** check box selected.
8. In the **IP or Host Name** field, select the base field name that you've used in the Extended Fields definition.
9. Click **Add**.

To use the Maps visualization in Oracle Log Analytics to view log records grouped by country or country code, see [Maps Visualization](#).

Use a Lookup in the Log Source

Oracle Log Analytics lets you enrich event data by setting up Lookup Field Enrichment options to add field-value combinations from lookups. Oracle Log Analytics uses lookups to match field-value combinations from events to an external lookup table, and if matched, Oracle Log Analytics appends the field-value combinations to the events. See [Create Lookups](#).

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the Oracle Log Analytics Configuration page, click the count of available log sources link.
3. In the Log Sources page, select the required log source where you want to define the lookup options and click **Edit**.

 **Note:**

You can also click **Create source** and create a new log source.

4. Click the **Field Enrichment** tab and then click **Add**.
5. In the Field Enrichment dialog box, select **Lookup** as the **Function**.
Click the **View details** link to see a sample representation of the Lookup function.
6. Keep the **Enabled** check box selected.
7. In the **Reference Lookup** field, select the lookup file that you uploaded earlier. The list shows all lookups that have been previously uploaded.

8. To map the key from the lookup to a field that's populated by your parser, in **Lookup Field**, select the field for which you've created the lookup, such as **Error ID**.

 **Note:**

The list for the input field will be limited to the fields that your log source populates. In this case, the **Lookup Field** is matched against your log entries field, **Error ID**.

9. Select the **Output Field**, such as **Error Text**.

When there's a match, then the lookup value is written to the output field, which in this case is the **Error Text** field.

10. Click **Add**.

When you display log records for the log source for which you created the lookup, you can see that the Output Field displays values that are populated against the log entries because of the lookup against the CSV file that you uploaded earlier. See [Create Lookups](#).

7

Administer: Other Actions

These are some of the additional administration tasks that you can perform.

Topics:

- [Set Up an Access Policy for a User](#)
- [Manage Annotations](#)
- [Manage Existing Log Sources](#)
- [Work with Entity Associations](#)
- [View Collection Warnings](#)
- [Export the Content from Oracle Log Analytics](#)
- [Purge Log Data](#)
- [Archive Log Data](#)
- [Create Credential for OCI Authentication](#)


Set Up an Access Policy for a User

As an administrator, you can have finer control on the level of data access provided to each individual user of Oracle Log Analytics. Create an access policy that defines the permissions by using query conditions and assign that policy to a user.


Topics:

- [Create an Access Policy](#)
- [Assign an Access Policy to a User](#)

Create an Access Policy


1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Access Policies** section, click **Create Policy**.
Alternatively, in the **Access Policies** section, you can click the available number of access policies link and then in the Policies tab, click **New**.
The **New Policy** page is displayed.
3. In the **Policy Name** field, enter the policy name. For example, enter `DB WLS Fatal Errors`.
4. In the **Description** field, enter the details of the policy, for example, the entity name, the user for whom the policy is created, and the restrictions that must be applied.

5. Under the **Policy Conditions** section, in the **Query condition** field, enter the query to specify the access condition. For example, to restrict the user from accessing logs that have errors of fatal severity, specify the query `Severity != Fatal`.

For more examples of the queries that can be used to define the policy condition, click the  icon.
6. Click **Select Entities** and select up to ten entities on which the specified conditions must be applied. For example, if the user must not access logs that have fatal severity errors from database and WebLogic server entities, then select `WebLogic Server` and `Oracle Database`.
7. Click **Save**.

The new access policy is created and listed in the Policies tab. You can now assign the new access policy that you created to a user to control the level of data access allowed to that user.

Assign an Access Policy to a User

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Access Policies** section, click the available number of access policies link.

The Access Policies page is displayed. Click the **Policy Assignments** tab, and click **New**.

The **New Policy Assignment** page is displayed.
3. In the **Policy Name** field, click the down arrow and select the access policy. For example, `DB WLS Fatal Errors`.
4. Click **Add** to add a user to assign the access policy. For example, `user@example.com`. You can add more users to the policy assignment, if required.
5. Click **Save**.

The access policy is assigned to the users that you specified. The assignments are listed in the Policy Assignments page.

Manage Annotations

Annotation is a unique flag that you can attach to the log records. It's composed with a descriptive message and an identifier consisting of a string which may contain spaces and special characters.


The annotations enable you to triage the issues by adding the information about the log records. Here are some of the scenarios where annotations can be useful:

- **For collaboration and sharing of found evidence:** You can share your findings or research with your team by annotating the log records that form important evidence.
- **For record-keeping of root-cause analysis:** In case of an issue, the research team conducts a detailed root-cause analysis and identifies the origin of the issue. By tagging the log records that indicate error, the team can record the research result for audit or future use.

- **For documenting patterns for future reference:** When you notice a behavioral pattern in the analysis, you can annotate the log records by detailing the observations. This can be useful for future reference when you encounter a similar pattern in another application or set of logs.

After adding the annotations to the log records, you can easily retrieve them by using the **Annotation Identifier** field to filter the log records. See [Filter Logs by Annotations](#).

Add an Annotation to Log Records

The annotation that you add must have a unique identifier and a detailed description of the annotation. The log records that are annotated carry an annotation icon  with a count of the number of the annotations added to the log records.

1. In the Log Explorer, view your logs in a records visualization like Records with Histogram or Records.

- In case you want to annotate a single log record, right click on the log record and click **Add Annotation**.
- If you want to annotate multiple log records from your search result, then click **More** menu on the top right corner of the window, and click **Annotate All Search Results**.


The Annotations dialog box opens. Verify that the log record message is visible in the panel.


2. The Annotations dialog box has two tabs, one listing the existing annotations, and the other to create a new annotation.

- If you want to add a new annotation, click the **New** tab. Input the following information about the annotation:
Identifier: This is a unique identifier of the annotation which is available for future use to annotate other log records.

Message: This is a detailed description of the annotation. For example, this can be your observation of a pattern, an association with an issue, or the details about important log records.

Click **Save**.




- If you want to reuse an existing annotation, then click the **Existing** tab. Click the **Reuse** button .
3. Close the Annotations dialog box.

The annotation is added to your selected set of log records. The annotation icon  is displayed with each log record that's annotated. The number displayed with the icon is the count of annotations added to the specific log record.

Edit an Annotation

After adding an annotation, you might want to edit it to cover a larger set of log records or to redefine the purpose of the annotation.

1. In the Log Explorer, view your logs in a records visualization like Records with Histogram or Records.
2. Search for the log records that have the specific annotation that you want to edit. See [Filter Logs by Annotations](#).

3. In the table of records, click the annotation icon . The Annotations dialog box opens. Click the **Existing** tab to see the annotations associated with the log record.
 4. Click the **Menu** icon  next to the annotation that you want to edit. Click **Edit** .
 5. Edit the message associated with the annotation. Click **Save**. Verify that the annotation message is updated in the Existing tab of the Annotations dialog box.
- Close the Annotations dialog box.



Manage Existing Log Sources

You can use the Log Sources page to edit existing log sources and add entity associations to existing log sources.

You can enable existing log sources by associating entities to them. See [Associate Entities to Existing Log Sources](#).

Edit Log Source

Modify the existing log source to customize it for your use case, but ensure that you consider the dependencies such as data filters, labels, extended fields, and other parser dependencies when you edit.

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click the available count of log sources link.
3. Click **Open Menu** () next to the log source entry that you want to edit and select **Edit**.
The Edit Log Source page is displayed.
4. Modify the log source definition and click **Save**.

Override Out-of-the-Box parsers

In an out-of-the-box log source, the default file parsers are already specified. If you want to override the out-of-the-box parsers used or change the order in which the parsers are applied on the logs, then follow these steps:

1. Under **File Parser > Specific Parsers** > click **Custom** > click the **Select Parsers** area.
2. Type a few characters from the name of the parser to get the list of suggestions. Select the parser.



Repeat the selection process to include multiple parsers. You can also include parsers that you've created for this customization. Ensure to specify the parsers in the same order in which they must be applied on the logs.

Follow the above steps to customize the log source if your log files are slightly different. Otherwise, create a new log source.

Important: Ensure that the new parsers that you selected have the same output fields as the old parsers because of the data enrichment dependency.

Create a Log Source Based on an Existing One

If you're not sure about how to create a log source, then you can use any existing log source to create a new one.

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click the available count of log sources link.
3. Click **Open Menu** () next to the log source entry based on which you want to create a new log source and select **Create Like**.

The Create Log Source page is displayed with the log definition fields populated with the definitions of the existing log source.

4. Modify the log source definition and click **Save**.

Work with Entity Associations

Configure new entity association and manage existing ones to enable log sources for specific targets to collect log data.


Topics:

- [Configure New Entity Associations](#)
- [Manage Existing Entity Associations](#)

Configure New Entity Associations

You can configure new entity associations or enable log sources for a target for collecting log data. To configure entity associations on a large scale, you can use the source-entity association APIs.

If you've enabled Oracle Infrastructure Monitoring on specific entities, then those entities are automatically available on Oracle Log Analytics to configure entity association for the log sources. See [Manage Existing Entity Associations](#).

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Entities** section, click **New association**.

Alternatively, you can click the count of available entities link in the **Entities** section and then in the Entities page, click **New Association**.

The Associate Entities for Log Collection page is displayed.

3. In the **Select Entities** section, select the target type, such as **WebLogic Server** from the **Entity Type** list.

 **Note:**

An entity type will be shown in the **Entity Type** list only when there is at least one existing log source with the specific entity type.

4. In the **Entities** section, click **Add Entities**, select the required entities to be associated.

The **Select Entities** dialog box is displayed. The entities that're eligible for association are listed in the **Eligible** tab. You can select up to fifty entities from the list. You can select the check box to the left of the **Entity Name** heading to select all the available entities.

From the list of entities, you can also select a remote agent that you've configured using the Oracle Log Analytics REST API, and associate it with the specific log sources.

Click the **Not Eligible** tab to view the entities that're not eligible for log collection. To determine the reason for the non-eligibility of an entity, expand it and view the details.
5. Select the entities from the **Eligible** tab, and click **Select**. Then click **Continue**.
6. In the **Select Log Sources** section, select the required log sources from the list of available sources and click **Continue**.


You can select the **Select All** check box to add all the available log sources.
7. To create the target association, in the **Confirmation** section, click **Associate Entities**.

The new target is displayed in the list of available entities.

Manage Existing Entity Associations

You can use the Entities page to associate log sources to existing entities.

Associate Log Sources to Existing Entities



1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Entities** section, click the count of available entities link.
3. Click the entity name to which you want to associate log sources.

Alternatively, you can click the count of associated log sources link.

This displays the Associated Log Sources: *<entity name:port>* page.
4. Click **Add** to display the **Select Log Sources** dialog box.
5. Select the required log sources and click **Select**.

You can also click the check box to the left of the **Log Source** heading to select all the available log sources.
6. Click **Save**.
7. In the **Save changes** dialog box, click **Save**.

Associate Entities to Existing Log Sources

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click the count of available log sources link.
3. Click the **Open Menu**  icon adjacent to the log source entry to which you want to associate entities and select **Add Entity Associations**.

The Associated Entities: <Log Source Name> page is displayed with a list of associated entities.

4. Click **Add**.

The **Select Entities** dialog box is displayed. The entities that're eligible for association are listed in the **Eligible** tab. You can select up to fifty entities from the list. You can select the check box to the left of the **Entity Name** heading to select all the available entities.

From the list of entities, you can also select a remote agent that you've configured using the Oracle Log Analytics REST API, and associate it with the specific log sources.

Click the **Not Eligible** tab to view the entities that're not eligible for log collection. To determine the reason for the non-eligibility of an entity, expand it and view the details.


5. From the **Eligible** tab, select the available entities for the selected log source, and then click **Select**. Click **Save**.
6. In the **Save changes** dialog box, click **Save**.

You can remove an associated entity or republish the association between the selected log source and an entity by selecting the entity name and clicking **Remove** or **Republish Associations**.

View Collection Warnings


Oracle Log Analytics lets you view the warning messages returned by log sources. This helps you to diagnose problems with the sources and to take corrective action.

View Warnings Summary

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click the available count of log sources link.
3. In the left navigation pane, click **Collection Warnings**.

This displays the summary of Oracle Log Analytics warnings.

View Entities with Collection Warnings

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Entities** section, click the count of entities with collection warnings link.

This displays the entities page with a list of entities whose log sources have returned warning messages.

3. Click the warning icon  adjacent to an entity entry.

This displays the warning messages returned by the log sources that contain errors.

Export the Content from Oracle Log Analytics


You can export the log parsers and user-defined log sources to an XML file using the Oracle Log Analytics user interface.

Topics:

- [Export the Log Parsers](#)
- [Export the Log Sources](#)

Export the Log Parsers


Export the user-defined log parsers from the log parsers page to an xml file.

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Parsers** section, you can click the available number of log parsers link. The Log Parsers page is displayed.
3. In the **Creation Type** filter at the top of the page, select *Custom*. The user-defined parsers are listed in the Log Parsers page.
4. Click the check box next to the log parsers that you want to export. Click **Export**. A zip file is generated for download. Specify a location on your host to store the zip file.

Unzip the zip file to recover the XML file containing the log parsers that you selected for export.

Export the Log Sources

When you export a log source, the associated parsers, extended field definitions and labels are also exported to the XML file. You can only export a user-defined log source and not out-of-the-box log sources.

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Log Sources** section, click the available number of log sources link. The Log Sources page is displayed.
3. In the **Creation Type** filter at the top of the page, select *Custom*. The user-defined log sources are listed in the Log Sources page.
4. Click the check box next to the log sources that you want to export. Click **Export**.

A zip file is generated for download. Specify a location on your host to store the zip file. Unzip the zip file to recover the XML file containing the log sources that you selected for export.


Purge Log Data


Oracle Log Analytics lets you purge log events that were loaded by agent or by an on-demand upload to reduce the index size of the log data. Oracle Log Analytics billing depends on the amount of log data indexed.

Purging enables you bring down your usage to reduce overage charges. Oracle Log Analytics can purge log data automatically per a set schedule or manually based on your need.

There are multiple ways to purge log data.

- *By specifying the time and date:* All data from all buckets created prior to the selected time range gets purged.
- *By creating a purge policy:* The old log data can be purged by specifying a schedule for purging and the query to filter the data to purge.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Storage** tile, click **Manage Storage**.
3. On the Manage Storage page, you can purge log data in one of the following methods:
 - Click **Purge By Time** tab. Select the date and time prior to which the log data that must be purged was collected. The log data from all the buckets prior to the selected time period gets purged.

In the **Query** field, enter the query to select a specific set of log data. For example, to select the logs from the entities of the type `Linux (Host)`, specify the query `*'Entity Type'='Linux (Host)'`. Click the help icon  to see more examples of the queries.

Click **Estimate** to determine the size of the storage that can be reclaimed based on the selection you made in the previous fields.


Click **Purge Data**.

- Click **Purge By Policy** tab. If you want to automate the purge activity, then you can create a purge policy by specifying the purge schedule, selecting the log data to purge, and enabling the policy. Click **Create New Policy**. The Create Purge Policy dialog box opens.


Enter a name for the new purge policy.

Under **Purge Data Older than**, select the time period from when the log data must be purged.

Under **Schedule Interval**, select the periodicity, day, and time of the purge action.

In the **Query** field, enter the query to select a specific set of log data. For example, to select the logs from the log source `Apache HTTP Server Access Logs`, specify the query `'Log Source'='Apache HTTP Server Access Logs'`. Click the help icon  to see more examples of the queries.

Click **Save**.

The purge policy is enabled. To edit or disable the policy, click  next to the policy name.

To view the purge activities performed, visit the **Storage > Activity** tab. See [View Archive, Recall, and Purge Activity](#).

Archive Log Data

Create an Archive Policy to specify the duration after which the log data will be automatically moved to archive storage which is available at a lesser cost. You can also recall the archived log data for active use.

Note:

The Archive Policy feature is available only when your Oracle Management Cloud instance is running on Oracle Cloud Infrastructure.

Archive Policy: Typically, after upload, the log data is available in hot storage for active use such as display and analysis in Oracle Log Analytics. To optimize the storage cost, you can create an archive policy that moves the log data from hot storage to cold storage (archive), after the specified number of days from the log's timestamp.

In the archive policy, you can also specify the number of days after which the logs in the archive storage are deleted automatically.

Recall Archived Logs: After the log data is archived, during the archival period, you can recall the select log data for active use. The logs are selected for recall by specifying the time range in which the timestamps of the logs are present. You can also release the recalled logs back to the archive pool after active use. Note that the recalled data will count towards your storage usage until you release it.

Note:

Your archive policy and recall activity may not complete if the timelines overlap with the purge policy. Ensure to review your purge policy and archive policy to avoid losing log data that can be easily archived.

To optimize your storage cost, you may also want to consider generating metrics for your logs by identifying the key performance indicators. You can store these metrics longer than the actual logs. See [Generate Log Metrics](#).

Create Archive Policy

If you're using only the recent logs for your display and analysis tasks in Oracle Log Analytics, then create the archive policy to specify the number of days after which the logs must be moved to Oracle Cloud Infrastructure (OCI) Object Storage so that you can optimize the storage cost.

 **Note:**


The Archive Policy feature is available only when your Oracle Management Cloud instance is running on Oracle Cloud Infrastructure.

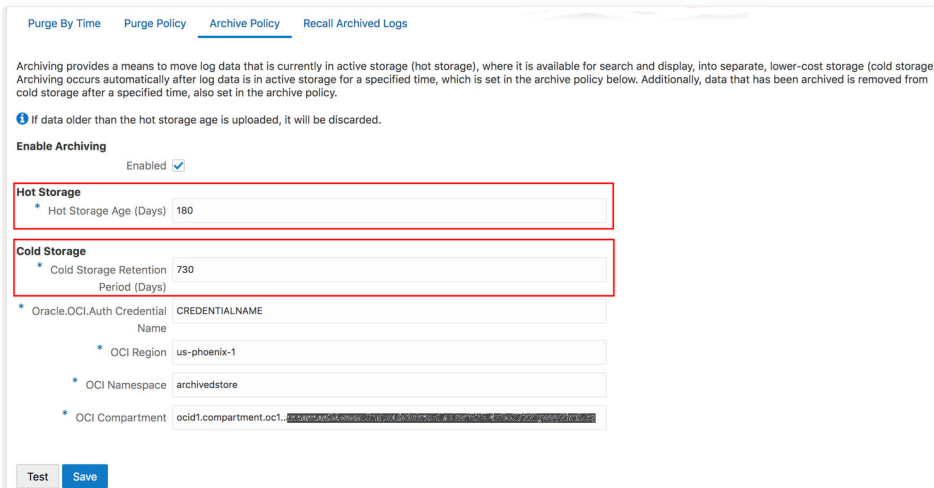
Pre-requisites: Log in to your OCI Console > **Tenancy Details** page and have the OCI namespace information ready.

Create credential to establish connection and authenticate the access to OCI Object Storage. See [Create Credential for OCI Authentication](#).

Additionally, note the details of the OCI compartment where the log data can be stored. To ensure data separation and to ensure access for select users of your OCI Object Storage, it is recommended that you create a separate compartment to store your archived log data from Oracle Log Analytics.

Ensure that the access policy for the OCI compartment is set such that you can store the log data from Oracle Log Analytics with the user credentials that you specify in the steps below. See [Oracle Cloud Infrastructure Documentation Getting Started - Adding Users](#).

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Storage** tile, click **Manage Storage**.
3. Click the **Archive Policy** tab.



Purge By Time Purge Policy **Archive Policy** Recall Archived Logs

Archiving provides a means to move log data that is currently in active storage (hot storage), where it is available for search and display, into separate, lower-cost storage (cold storage). Archiving occurs automatically after log data is in active storage for a specified time, which is set in the archive policy below. Additionally, data that has been archived is removed from cold storage after a specified time, also set in the archive policy.

Info If data older than the hot storage age is uploaded, it will be discarded.

Enable Archiving

Enabled

Hot Storage

* Hot Storage Age (Days) 180

Cold Storage

* Cold Storage Retention Period (Days) 730

* Oracle.OCI.Auth Credential Name CREDENTIALNAME

* OCI Region us-phoenix-1

* OCI Namespace archivedstore

* OCI Compartment ocid1.compartment.oc1..[redacted]

Test Save

4. Check the **Enable Archive** checkbox to enable the policy that you create.
5. **Hot Storage Age (Days):** This is the count of the days after which the log data in the hot storage must be archived. The count is calculated based on the timestamp of the logs. For example, if your logs have the timestamp July 4, 2018 23:43:12, and you've specified the Hot Storage Age as 30, then the logs will be moved to archive storage on Aug 3, 2018.

If you upload logs to Oracle Log Analytics that are older than the hot storage age, then they're not considered in the archive policy.


6. **Cold Storage Retention Period (Days):** This is the number of days that the log data must remain in the archive storage after which it must be deleted. From the example in step 5, if the log data is archived on Aug 3, 2018, and you specified Cold Storage Retention Period as 365 days, then the archived logs are deleted from archive storage on Aug 03, 2019.
7. *Other Cold Storage Details:* Enter the following information that you collected earlier:
 - **Oracle.OCI.Auth Credential Name:** The credential name that you provided in step 2.
 - **OCI Region:** This is auto-filled with the region where your Oracle Management Cloud instance is located.
 - **OCI Namespace:** Collect this from OCI.
 - **OCI Compartment:** Collect this from OCI.
8. Click **Test** to verify that the information that you entered is valid. If the connection is not established with Oracle Cloud Infrastructure using the credentials provided, then modify the OCI account information accordingly.
Click **Save**.

Note that if you purge the logs when they're in active use, then the logs won't be archived per the policy.

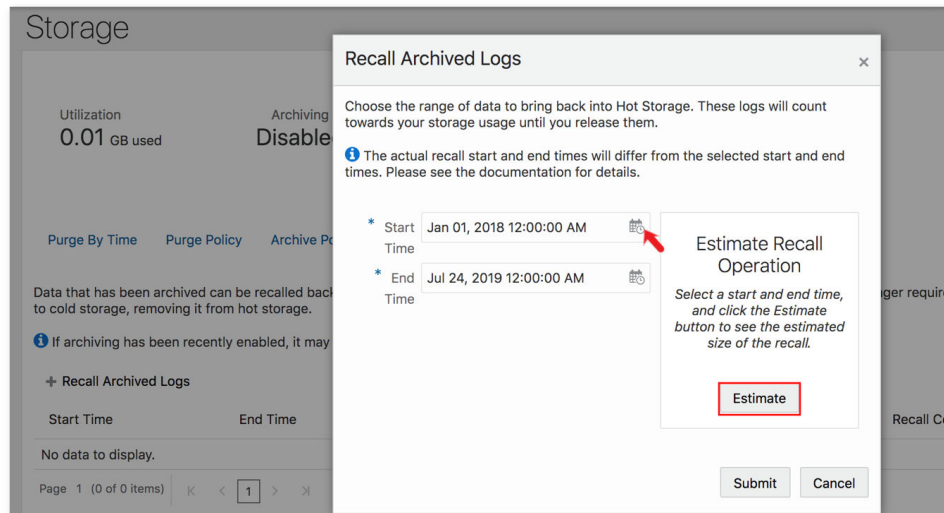
Recall Archived Logs

If you want to use those logs that are archived, for viewing and analysis, then you can recall the logs. The recalled data will count towards your storage usage until you release it.

You can recall and release your select set of logs multiple times. However, the recall feature is enabled only if you already have archived logs.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Storage** tile, click **Manage Storage**.
3. Click the **Recall Archived Logs** tab. The recall archived logs page is displayed.
4. Click **+ Recall Archived Logs** to start a new recall activity.

The **Recall Archived Logs** dialog box opens.




5. Select the time range of the logs that you want to recall, by specifying the **Start Time** and **End Time**.

Note that the start time and end time are extended to align with the log index structure. So, when you view the list of active recalls or visit the activity tab, you may get the before and after time extended beyond your chosen time range.

6. Click **Estimate** to determine the size of the logs that you've selected for recall.
7. Click **Submit** to proceed with the recall of the selected logs.

The Recall Archived Logs dialog box closes, and the recall activity is listed in the Recall Archived Logs tab. The table specifies the **status**, **start time**, **end time**, **size**, and **date** of recall activity.


Watch the status of recall. You can use the recalled logs for viewing and analysis after the recall activity is complete.

8. After active use of the recalled logs, if you want to release it back to the archive pool, click the menu icon  in the row corresponding to your recalled logs, and select **Release**.

The recalled logs will then be released back into the archive pool. This will enable you to optimize your storage size.

View Archive, Recall, and Purge Activity

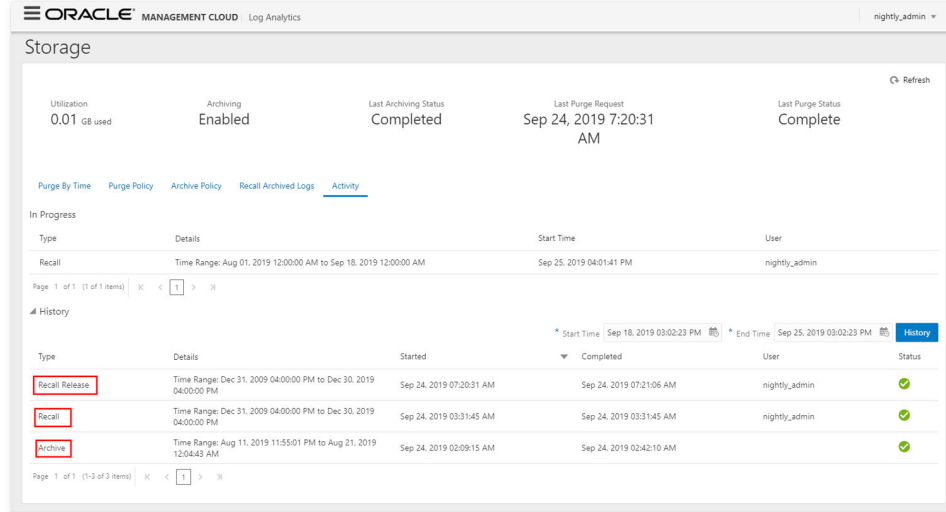
You can view the summary of your archive, recall, release, and purge activities to maintain close control of your storage use and also to track the status of your key logs that have been part of the activities.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. In the **Storage** tile, click **Manage Storage**.
3. Click the **Activity** tab.

The page displays the summary of the activities in progress.

4. To view the historic summary, specify the **Start Time**, **End Time**, and click **History**.

The page displays the summary of the past activities in the specified time range.



Create Credential for OCI Authentication

To be able to establish connection from Oracle Log Analytics to Oracle Cloud Infrastructure (OCI) Object Storage service, you must configure the Oracle Management Cloud credential store with your OCI authentication details.

Topics:

- [Prerequisites for Creating Credentials](#)
- [Create Credentials using UI](#)
- [Create Credentials using REST API](#)

Prerequisites for Creating Credentials

Keep the following authentication information ready to create OCI specific credentials in the Oracle Management Cloud credential store to access OCI from Oracle Management Cloud:

- The following details of the RSA key pair:
 - **Fingerprint for RSA key:** The fingerprint for the RSA key pair that you're using to access OCI. It looks something like this:
12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef.
 - **Private key:** The private key in the RSA key pair. The RSA key pair is generated in **PEM format** (minimum 2048 bits). You would upload the public key from the key pair in the OCI console to obtain authentication.
- **Tenancy OCID:** The OCID of the tenant.
- **User OCID:** The OCID of the user.

For more information about the keys and OCIDs, generating RSA key pair, and generating the fingerprint for the key pair, see [Oracle Cloud Infrastructure Documentation Getting Started - Required Keys and OCIDs](#).

Create Credentials using UI

1. Go to the administration page at the location **Administration > Security > Credential Store**. Click **New Credential**.

The **Create Credential** dialog box opens.

The screenshot shows a 'Create Credential' dialog box with the following fields and values:

- Credential Name:** Please Enter Credential Name
- Credential type:** Oracle.OCI.Auth
- Description:** Please Enter Description
- fingerprint:** Fingerprint for the key pair being used.
- pass_phrase:** Passphrase used for the key, if it is encrypted.
- private_key:** The private key
- tenancy:** OCID of tenant calling the API.
- user:** OCID of the user calling the API.

Buttons: Create, Cancel

2. From the **Credential type** menu, select *Oracle.OCI.Auth*.

Provide a name to identify the credentials in the field **Credential Name**, and provide a **Description**.

3. Provide the following information about your OCI account:

- **fingerprint:** The fingerprint for the RSA key pair
- **pass_phrase:** Leave this field empty
- **private_key:** The unencrypted private key in the RSA key pair. This should **not** be encrypted by using any passphrase. The private key spans over multiple lines. Ensure to replace all the newline characters with the space character and use the resulting key.
- **tenancy:** The OCID of the tenant
- **user:** The OCID of the user

Click **Create**.

Create Credentials using REST API

1. Provide the credential information to access the OCI account in the json format and store the OCI properties file on the local machine, for example, OCI_creds.json:

```
{
  "name": "<OCI_CREDENTIAL>",
  "credtype": "Oracle.OCI.Auth",
  "columnValues": {
    "user": "<User OCID>",
    "private_key": "<Unencrypted Private Key Text By Replacing Newline
Characters With Space Character>",
    "fingerprint": "<Fingerprint of the public key>",
    "tenancy": "<Tenancy OCID>"
  }
}
```

In the above format, provide the following information:

- **name:** Provide a name to identify the credentials.
 - **credtype:** Specify `Oracle.OCI.Auth` for OCI authentication.
 - **fingerprint:** The fingerprint for the RSA key pair
 - **private_key:** The unencrypted private key in the RSA key pair. This should **not** be encrypted by using any passphrase. The private key spans over multiple lines. Ensure to replace all the newline characters with the space character and use the resulting key.
 - **tenancy:** The OCID of the tenant
 - **user:** The OCID of the user
2. To register the credentials in the Oracle Management Cloud credential store, run the cURL command in the following format:

```
curl -X POST -k -u '<username>:<password>' -H 'X-USER-IDENTITY-
DOMAIN-NAME:<identity_domain_name>' "https://<OMC_URL>/serviceapi/
credentialStore/api/v1/credentials" -H 'Content-Type:application/
json' -d "@<json_file>"
```

In the above format:

- **username:** Your user name to access the Oracle Management Cloud account. Depending on the type of your cloud account, the username will be in one of the following formats:
 - **<username>** for Oracle Identity Cloud Service (IDCS) based account.
 - **<tenant_name>.<username>** for Traditional Cloud Account. Follow the same steps as those to obtain OMC_URL. **TENANT_NAME** is displayed above OMC_URL.

For information on the types of cloud accounts, see About Oracle Cloud Accounts in *Getting Started with Oracle Cloud*.

- **password**: The password to access the Oracle Management Cloud account
- **OMC_URL**: Obtain OMC URL from **Agents** page.
 - a. On the Oracle Management Cloud home page, click the OMC Navigation Menu on the top-left corner and navigate to **Administration > Agents**.
 - b. On the Agents page, click the **Download** tab. The Agent Software Download page is displayed.
 - c. Select **Cloud Agent** from the Agent Type drop-down list. The **OMC_URL** is displayed. Note the URL.
- **identity_domain_name**: Depending on the type of your cloud account, the identity domain name will be one of the following:
 - **IDCS Identity Domain**: For IDCS based cloud account, typically of the format `idcs-j29b928a146e4bdd7fef12a6e6a9excm`. Collect this from your cloud account details page.
 - **Tenant Name**: For Traditional Cloud Account, typically of the format `acme`. Follow the same steps as those to obtain OMC_URL. **TENANT_NAME** is displayed above OMC_URL.

For information on the types of cloud accounts, see About Oracle Cloud Accounts in *Getting Started with Oracle Cloud*.

- **json_file**: The OCI properties file that you created in step 1.

An example cURL command to register the OCI credentials for a traditional cloud account:

```
curl -X POST -k -u 'acme.JohnDoe:john_password' -H 'X-USER-IDENTITY-DOMAIN-NAME:acme' "https://acme.example.com:4443/serviceapi/credentialStore/api/v1/credentials" -H 'Content-Type:application/json' -d "@OCI_creds.json"
```

An example cURL command to register the OCI credentials for an IDCS based cloud account:

```
curl -X POST -k -u 'JohnDoe:john_password' -H 'X-USER-IDENTITY-DOMAIN-NAME:idcs-j29b928a146e4bdd7fef12a6e6a9excm' "https://omc-fb68f2df9f4a27bda5c45778f62f41.example.com/serviceapi/credentialStore/api/v1/credentials" -H 'Content-Type:application/json' -d "@OCI_creds.json"
```

Part III

Use Oracle Log Analytics

After setting up Oracle Log Analytics, you can search your logs for key data, visualize, analyze the log data, or optimize your set up for better efficiency.

Topics:

- [Visualize Data Using Charts and Controls](#)
- [Filter and Search Through the Log Data](#)
- [Save and Share Log Searches](#)
- [Create An Alert Rule](#)
- [Transform Logs into Operational Insight](#)

8

Visualize Data Using Charts and Controls

Use the Visualize panel of Oracle Log Analytics to present search data in a form that helps you better understand and analyze.

Topics:

- [Select the Visualization Type](#)
- [Log Scales Visualization](#)
- [View the Field Summary](#)
- [View an Entity Card](#)
- [Bar Charts Visualization](#)
- [Clusters Visualization](#)
- [Line Charts Visualization](#)
- [Maps Visualization](#)
- [Summary Tables](#)
- [Word Cloud Visualization](#)
- [Link Visualization](#)
- [Link by Cluster](#)


Using the Visualization Panel

Consider a situation where you've performed a search operation on your log data either by using the Search field or by using the target or field attributes. Now, you want to visualize the search results in a specific format for analysis.

In this section, you'll refer to the [Example Scenario: Perform Dynamic Log Analysis](#) search results, and use the Visualize panel of Oracle Log Analytics to represent the search data in the required format.

Drag the **Data** and **Visualize** palettes to increase or decrease their size for better visualization with the charts.

To change the visualization of the search results generated by [Example Scenario: Perform Dynamic Log Analysis](#) for analyzing the number of occurrences of the error `BEA-310002` over the last 30 days:

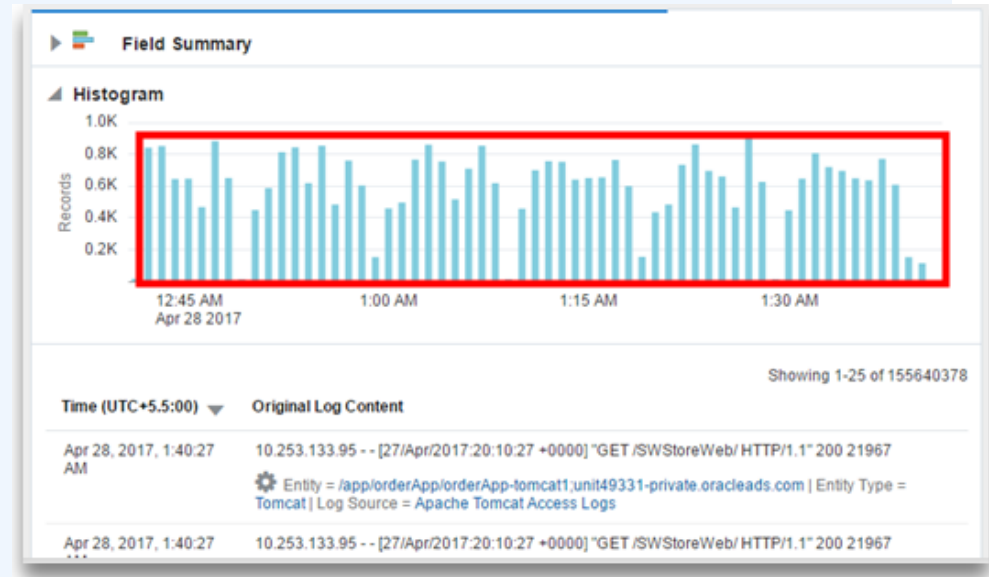
1. In Oracle Log Analytics, in the **Visualize** panel, click the visualization options.
2. Select **Records With Histogram** ()

The data is represented in the form of a table with histogram.

 **Note:**

If you run a query that needs to fetch data for a long duration, such as the last 7 days or the last 1 month, then Oracle Log Analytics may take some time to display the entire result set in the selected visualization. In this case, Oracle Log Analytics keeps updating the visualization until the query has finished running.

The following image displays the visualization when the query is still running:



The following image displays the visualization after the query has finished running:



When the data collection is in progress, the display on the chart might be incomplete. There's a drop-off in the visualization for the incoming data. This can be observed on any of the charts of Oracle Log Analytics that involve real-time display of incoming data. The following chart displays a drop-off in the line chart visualization when the data collection is in process:



Select the Visualization Type


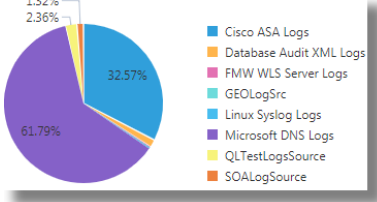

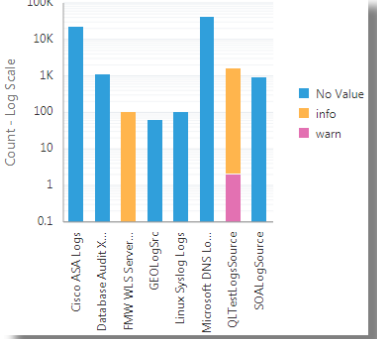
The interactive data visualizations in Oracle Log Analytics enable you to get deeper insights into your log data. Based on what you want to achieve with your data set, you can select the visualization type that best suits your application.

Here are some of the things you can do with visualizations:


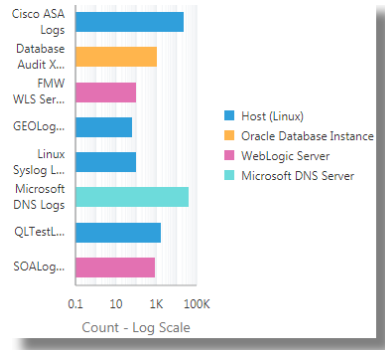

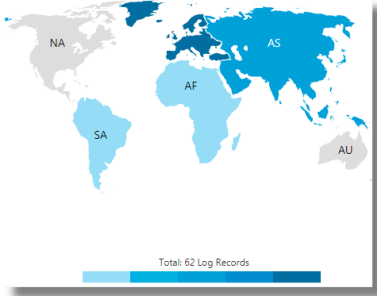
- [Compare and Contrast the Data Set Using One or Two Parameters](#)
- [Summarize the Data Set Using Key Parameters](#)
- [Group and Drill Down to the Specific Data Set](#)
- [Analyze the Data Set Using Multiple Key Parameters](#)
- [Perform Advanced Analysis of the Data Set](#)

Compare and Contrast the Data Set Using One or Two Parameters

Use these simple graphs to visualize your data set and compare the log records based on one or two key parameters:

Visualization Type	What You Input	What Output You Get	What You Can Do																																				
<p>Pie : A pie chart shows the overall composition of a data set by encoding the percentage values in angles.</p>	<p>Default Group By field: Log Source. Optionally, you can change this parameter.</p>	<p>A circular representation of the count of the log records that are grouped using the input parameter.</p>	<p>Compare the broad groups in the circle that indicate percentages of the whole data set. For example, compare the percentages of the counts of the log records from various log sources.</p>  <table border="1"> <caption>Pie Chart Data</caption> <thead> <tr> <th>Log Source</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Cisco ASA Logs</td> <td>32.57%</td> </tr> <tr> <td>Microsoft DNS Logs</td> <td>61.79%</td> </tr> <tr> <td>Database Audit XML Logs</td> <td>2.36%</td> </tr> <tr> <td>Other Sources</td> <td>1.32%</td> </tr> </tbody> </table>	Log Source	Percentage	Cisco ASA Logs	32.57%	Microsoft DNS Logs	61.79%	Database Audit XML Logs	2.36%	Other Sources	1.32%																										
Log Source	Percentage																																						
Cisco ASA Logs	32.57%																																						
Microsoft DNS Logs	61.79%																																						
Database Audit XML Logs	2.36%																																						
Other Sources	1.32%																																						
<p>Bar : The count of the log records is displayed as segmented columns against the time period.</p>	<p>Default X-axis field: Log Source. Optionally, you can change this parameter. Additionally, provide a second parameter in the Group by section to view a colored and stacked bar graph.</p>	<p>Bar graph: The input parameter represented along the x-axis as segmented columns, with the height of the column denoting the count.</p> <p>Stacked bar graph: The key input parameter is grouped by the second parameter, and is represented as a stacked bar graph along the x-axis. The overall height of the column denotes the count. The colored stack represents the grouping.</p>	<p>Bar graph: Compare the sizes of the segmented columns to compare the count of the log records based on the input parameter. For example, compare the count of log records from each log source.</p> <p>Stacked bar graph: Here, you can compare not only the count of the values of the input parameter, but also notice the grouping of it, based on the second parameter. In the following example, the count of the log records from the log sources are obtained by the overall height of the segmented columns. The log records in each column are grouped based on the severity of the errors noticed in them.</p>  <table border="1"> <caption>Stacked Bar Chart Data (Approximate)</caption> <thead> <tr> <th>Log Source</th> <th>No Value</th> <th>info</th> <th>warn</th> </tr> </thead> <tbody> <tr> <td>Cisco ASA Logs</td> <td>10000</td> <td>0</td> <td>0</td> </tr> <tr> <td>Database Audit XML Logs</td> <td>1000</td> <td>0</td> <td>0</td> </tr> <tr> <td>FMW WLS Server Logs</td> <td>100</td> <td>0</td> <td>0</td> </tr> <tr> <td>GEOLogSrc</td> <td>100</td> <td>0</td> <td>0</td> </tr> <tr> <td>Linux Syslog Logs</td> <td>100</td> <td>0</td> <td>0</td> </tr> <tr> <td>Microsoft DNS Logs</td> <td>10000</td> <td>1000</td> <td>100</td> </tr> <tr> <td>QLTestLogSource</td> <td>1000</td> <td>0</td> <td>0</td> </tr> <tr> <td>SOALogSource</td> <td>1000</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Log Source	No Value	info	warn	Cisco ASA Logs	10000	0	0	Database Audit XML Logs	1000	0	0	FMW WLS Server Logs	100	0	0	GEOLogSrc	100	0	0	Linux Syslog Logs	100	0	0	Microsoft DNS Logs	10000	1000	100	QLTestLogSource	1000	0	0	SOALogSource	1000	0	0
Log Source	No Value	info	warn																																				
Cisco ASA Logs	10000	0	0																																				
Database Audit XML Logs	1000	0	0																																				
FMW WLS Server Logs	100	0	0																																				
GEOLogSrc	100	0	0																																				
Linux Syslog Logs	100	0	0																																				
Microsoft DNS Logs	10000	1000	100																																				
QLTestLogSource	1000	0	0																																				
SOALogSource	1000	0	0																																				

See [Bar Charts Visualization](#).

Visualization Type	What You Input	What Output You Get	What You Can Do
<p>Horizontal bar</p>  : The count of the log records is displayed as segmented rows against the time period.	<p>Default Y-axis field: Log Source. Optionally, you can change this parameter. One parameter, for example, Log Source. Additionally, provide a second parameter in the Group by section to view a colored and stacked horizontal bar graph.</p>	<p>Horizontal bar graph: The input parameter represented along the y-axis as segmented columns, with the width of the row denoting the count. Stacked horizontal bar graph: The key input parameter is grouped by the second parameter, and is represented as a stacked bar graph along the y-axis. The overall width of the row denotes the count. The colored stack represents the grouping.</p>	<p>Horizontal bar graph: Compare the sizes of the segmented rows to compare the count of the log records based on the input parameter. For example, compare the count of log records from each log source. Stacked horizontal bar graph: Here, you can compare not only the count of the values of the input parameter, but also the grouping of it, based on the second parameter. In the following example, the count of the log records from the log sources are obtained by the overall width of the segmented rows. The log records in each row are grouped based on the entity type.</p> 
<p>Map : The geographical distribution of the log records is displayed on the world map based on the location the log records are collected from.</p>	<p>Default Group by field: Client Host Continent Code. Optionally, you can change this geographical parameter. For example, Client Host Country Code or Client Host Region.</p>	<p>The geographical distribution of the count of log records based on the input geographical parameter.</p>	<p>Compare the count of the log records based on their geographical distribution. In the following example, the log records are distributed based on the Client Host Continent Code field in the log data.</p> 

See [Maps Visualization](#).




Visualization Type	What You Input	What Output You Get	What You Can Do
<p>Line : The count of the log records against the specific time is plotted with the line tracing the number that represents the count.</p>	<p>Default Group By field: Log Source. Optionally, you can change this parameter.</p>	<p>A plotted line that presents the count of the input parameter along the y-axis tracked on the timeline along the x-axis.</p>	<p>Compare the count of the log records based on the input parameter represented by separate lines plotted against time. In the following example, the count of log records from various log sources are plotted against time in each line.</p> 
<p>Word Cloud : The data set is represented by a set of word tiles, whose size indicate the count of log records in each group and the colors indicate the grouping.</p>	<p>Default Group By field: Log Source. Optionally, you can change this parameter. Additionally, provide a second parameter in the Color section to further group the data set. For example, Entity Type.</p>	<p>A word cloud where the size of the word tile represents the count. Additionally, when you provide a second input parameter, you can see a colored word cloud where the words are grouped by the second parameter. The groups are represented by colors.</p>	<p>Compare the count of the log records based on the size of the word tiles that represent the input parameter. If you provided the second parameter, then you can also view the color grouping of the word tiles. In the following example, the size of the word tiles represent the count of the log records from each log source. The color of the word tiles indicate the entity type of each group.</p> 


See [Line Charts Visualization](#).


See [Word Cloud Visualization](#).

Summarize the Data Set Using Key Parameters

View these charts to get detailed information about the data set:

Visualization Type	What You Input	What Output You Get	What You Can Do
Summary table 	<p>Default Display Fields: count</p> <p>Optionally, you can select a different math function to perform on the data set. For example, Percentile, Median, or Average.</p> <p>Default Group by field: Log Source</p> <p>Optionally, you can select more input parameters for the Group by section that will enable further grouping of the data set.</p>	<p>A table that displays the following:</p> <ul style="list-style-type: none"> Each column of the table represents a display field and the fields that you want to use for grouping the data set. The number of rows in the table indicate the number of groups. 	<p>Summary table is the most versatile visualization chart that can perform statistical analysis on any type of input data. It also permits multiple input parameters in the Group by section, thus enabling more complex deductions from the analysis.</p> <ul style="list-style-type: none"> Perform statistical analysis on the entire data set. Select the fields for statistical analysis that can help you understand the data set. Group your statistical analysis to correlate the results. <p>See Summary Tables.</p>
Records 	<p>Default Display Fields: Entity, Entity Type, Log Source, Host Name (Server), Problem Priority, and Label.</p> <p>Optionally, you can select more input parameters that will display in the chart.</p>	<p>A chart of log records that contain:</p> <ul style="list-style-type: none"> The time when the log was collected Original log content and the selected display fields 	<ul style="list-style-type: none"> View the original log content to understand and correlate the values of the display fields. View the log content corresponding to a specific log collection time.
Table 	<p>Default Display Fields: Entity, Entity Type, Log Source, Host Name (Server), Problem Priority, and Label.</p> <p>Optionally, you can select more input parameters that will display in the table.</p>	<p>A table that displays the following:</p> <ul style="list-style-type: none"> Each column of the table represents a display field that you selected Each row of the table represents a log record 	<ul style="list-style-type: none"> Prioritize and select the fields that you want to view in the table to help you make decisions. Filter the log content and view only the data in each log record that is of interest to you.

Visualization Type	What You Input	What Output You Get	What You Can Do
Distinct 	<p>Default Display Fields: Log Source</p> <p>Optionally, you can select more input parameters that will display in the table.</p>	<p>A table that lists the unique values of the default field. If you included more fields, then the table displays the following:</p> <ul style="list-style-type: none"> Each column of the table represents a display field that you selected. The number of rows in the table indicate the number of groups. Each row indicates a unique group of the display fields that are available in the log data. 	<ul style="list-style-type: none"> Identify the unique values of the fields in your log data. Identify unique groups of fields in the log data.


Alternatively, use the **Tile**  visualization to summarize the data set. By default, the tile visualization summarizes the overall count of the log records. Identify the fields to group the log records in order to refine the summary. For example, you can group the log records by log source. This is a sample summary output of the grouping: `8 Distinct values of Log Source.`


Group and Drill Down to the Specific Data Set

Use these simple graph and chart visualizations to group the log records based on a parameter, and then drill down to the individual log records to investigate further.

A histogram is a graph that lets you view the underlying frequency distribution or shape of a continuous data set. It shows the dispersion of log records over a specific time period with segmented columns. You can optionally select a field for the **Group by** section to group the log records for the histogram visualization.


To learn more about the input parameters and the output for the **Records** and **Table** visualizations, see [Summarize the Data Set Using Key Parameters](#).

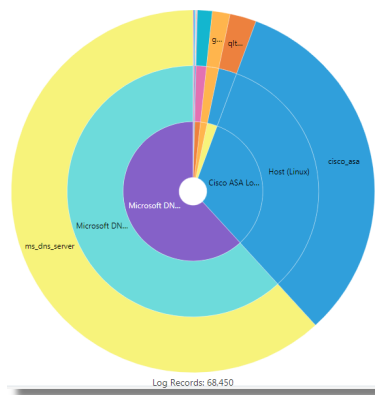
Visualization Type	What You Can Do
Records with histogram 	<ul style="list-style-type: none"> Reduce the size of the data set for understanding and analyzing by grouping the log records in the histogram, and drilling down to specific log records. You can click a select segment in the histogram to drill down to a specific set of log records and to view the original log content. The combination of the histogram graph and records chart enables you to drill down to the specific log content faster. <p>See View the Field Summary.</p>


Visualization Type	What You Can Do
Table with histogram 	<ul style="list-style-type: none"> Use an appropriate field to group the log records in the histogram visualization. From the histogram graph, identify the data set that you want to view the field details of, and view it in the table. The combination of the histogram graph and table enables you to drill down to the specific data set faster.

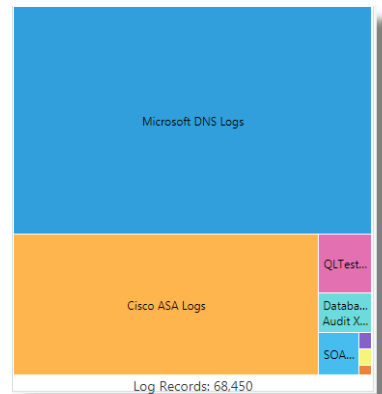
Analyze the Data Set Using Multiple Key Parameters

Use these complex graph visualizations to determine the hierarchical and fractional relationships of the fields in the whole data set:

Visualization Type	What You Input	What Output You Get	What You Can Do
Sunburst 	<p>Default Value: count</p> <p>Optionally, you can select a different field whose count can help to generate the sunburst.</p> <p>Default Group by field: Log Source</p> <p>Optionally, you can select more input parameters for the Group by section that will enable further grouping of the data set. For example, Entity Type and Entity.</p>	<p>By default, a sunburst that represents the log records grouped by the default parameter. The size of a sector in the circle indicates the count of the log records in the specific data set. If you specified more fields for grouping, you'll see a concentric sunburst, with the innermost ring representing the first computation of the grouping, and the subsequent rings representing the following computations, in that order.</p>	<p>Use the sunburst visualization to analyze hierarchical data from multiple fields. The hierarchy is represented in the form of concentric rings, with the innermost ring representing the top of the hierarchy.</p> <p>In the following example, the log records are grouped using the fields Log Source, Entity Type and Entity. Click a segment to view the Records with Histogram visualization for the specific data set. The records chart lists the original log content emphasizing the default display fields.</p>





Visualization Type	What You Input	What Output You Get	What You Can Do
Treemap 	<p>Default Value: count</p> <p>Optionally, you can select a different field whose count can help to generate the treemap.</p> <p>Default Group by field: Log Source</p> <p>Optionally, you can select more input parameters for the Group by section that will enable further grouping of the data set.</p>	<p>A treemap that represents the log records grouped by the default parameter. The size of the rectangles indicate the count of the log records in the specific data set. If you specified more fields for grouping, you'll see a nested treemap that groups the log records based on all the parameters that you specified. The nested treemap also shows the fractional relationship of the fields in each data set.</p>	<p>Use the treemap visualization to analyze the data from multiple fields that are both hierarchical and fractional, with the help of interactive nested rectangles.</p> <p>In the following example, the log records are grouped using the Log Source field. Click a rectangle to view the Records with Histogram visualization for the specific data set. The records chart lists the original log content emphasizing the default display fields.</p>



Perform Advanced Analysis of the Data Set






Use these visualizations to perform advanced analysis of the large data set to figure out the root cause an issue, to identify potential issues, to view trends, or to detect an anomaly.

Visualization Type	What You Input	What Output You Get	What You Can Do
Cluster 	<p>The cluster visualization works on the entire data set and isn't based on a specific parameter.</p>	<p>The Cluster view displays a summary banner at the top showing the following tabs:</p> <ul style="list-style-type: none"> • Total Clusters: Total number of clusters for the selected log records. • Potential Issues: Number of clusters that have potential issues based on log records containing words such as error, fatal, exception, and so on. • Outliers: Number of clusters that occurred only once during a given time period. • Trends: Number of unique trends during the time period. Many clusters may have the same trend. So, clicking this panel shows a cluster from each of the trends. 	<p>Clustering uses machine learning to identify the pattern of log records, and then to group the logs that have similar patterns. You can investigate further from each of the tabs based on your requirement. When you click any of the tabs, the histogram view of the cluster changes to display the records for the selected tab.</p> <p>Clustering helps significantly reduce the total number of log entries that you have to explore, and points out the outliers. See Clusters Visualization.</p> <p>For an example use case of cluster visualization, see Example Scenario: Detect Anomalies Using Outliers.</p>
Link 	<p>Default Link By field: Log Source.</p> <p>Optionally, you can select more input parameters for the Link By section for more relevant grouping of the log data. You can also select additional parameters for the Display Fields section.</p>	<ul style="list-style-type: none"> • The Groups tab displays a bubble chart that represents the groups formed with the fields used for linking in the commonly seen ranges. The link by field is plotted along the x-axis, and the group duration is plotted along the y-axis. The size of each bubble in the graph is determined by the number of groups contained in that bubble. <ul style="list-style-type: none"> • Trends: Project the time series data using the Link Trend feature. • The histogram tab displays the log records or groups in the histogram visualization. <p>The groups table lists parameters like Log Source, Entity Type, Entity, Count, Start Time, End Time, and Group Duration for each group. If you specified more display fields, they're included in the table too.</p>	<p>Use the link visualization to perform advanced analysis of log records by combining individual log records from across log sources into groups, based on the fields you selected for linking.</p> <p>The bubble chart shows the anomalies in the patterns based on the analysis of the groups. You can further examine the anomalies by clicking an individual bubble or select multiple bubbles. To view the details of the groups that correspond to the anomaly, select the anomaly bubble in the chart. You can investigate the anomaly to identify and rectify issues. See Link Visualization.</p> <p>For some example use cases of link visualization, see Perform Advanced Analytics with Link.</p>

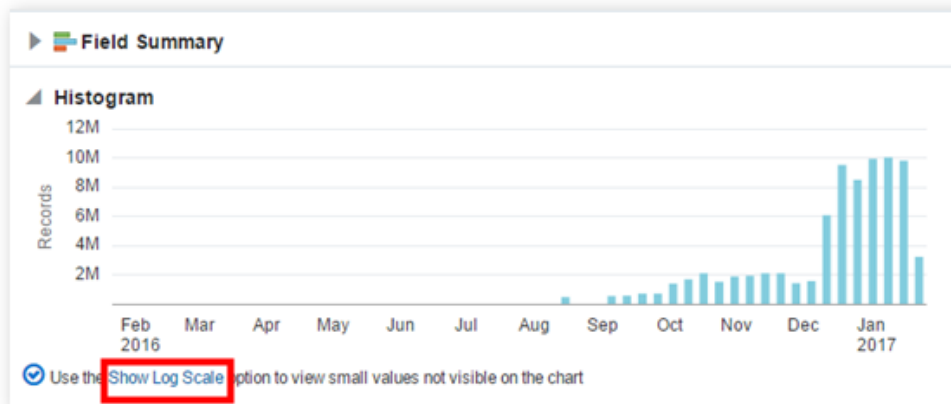
Visualization Type	What You Input	What Output You Get	What You Can Do
Link by Cluster	Select <code>cluster()</code> to group the log data using the query section and the input parameter for the Link By section for more relevant grouping of the log data.	<p>The Groups tab displays a bubble chart that represents the groups formed with the selected field and the clusters used for linking in the commonly seen ranges. The link by field is plotted along the x-axis, and the group duration is plotted along the y-axis.</p> <p>The groups table lists parameters like Entity Type, Cluster Sample, Count, Start Time, End Time, and Group Duration for each group. If you specified more display fields, they're included in the table too.</p>	<p>Use the combination of link and cluster visualizations to perform this analysis. The machine learning capability of the cluster visualization to identify clusters and potential issues, and the ability of link visualization to group the log records based on the selection of fields are combined to narrow down your analysis to small anomaly groups or potential issues.</p> <p>You can refine your query and be specific about the output required on the bubble chart. The analysis generates clusters that are grouped based on your selection of the field for analysis. You can investigate the anomalies further to arrive at conclusive decisions of the analysis.</p> <p>See Link by Cluster.</p>

Log Scales Visualization

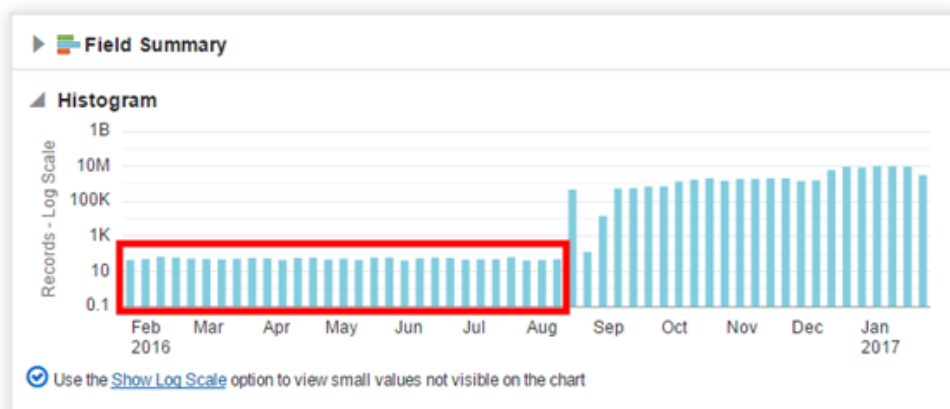
Log scales allow a large range of values to be displayed without small values being compressed down into bottom of the graph.

1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select any visualization option containing a histogram ( and ) , bar chart ( and ) , or line chart ().
3. Click **Show Log Scale** to view the smaller values that aren't otherwise visible on the chart.

This option is displayed only when the chart contains bars that aren't visible. This option is also useful when highlighting small values.



The smaller values are now displayed.



View the Field Summary



When using any variation of the Histogram visualization, Oracle Log Analytics provides a high-level summary of the key fields of the log entries that are returned as search results. The field summary, which is represented in the form of horizontal bar charts, presents a quicker way to analyze your search results, by grouping the entries based on the display fields that you've selected.

For example, in your search query, you've selected Oracle WebLogic Server as the entity type, FMW WLS Server logs as the log source, and bea-090898, bea-001110, and bea-090152 as error IDs. You've also selected Entity Type, Log Source, and Error ID as display fields. Then, the field summary displays a snapshot of the search results grouped by the display fields. If you hover your cursor over a particular error ID in the Field Summary section, then Oracle Log Analytics highlights the error ID and also highlights the relevant portions of the horizontal bar charts of all the associated entity types and log sources that are displayed in the summary. In addition, the relevant portion in the Histogram visualization is also highlighted, depicting the number of records that are associated with the selected error ID.

If you click the selected error ID, then the field summary, histogram, and the table of records change to return only those entity types, log sources, and records that contain the selected

error ID. You can add as many fields as you want in the summary by dragging and dropping attributes from the Fields panel.

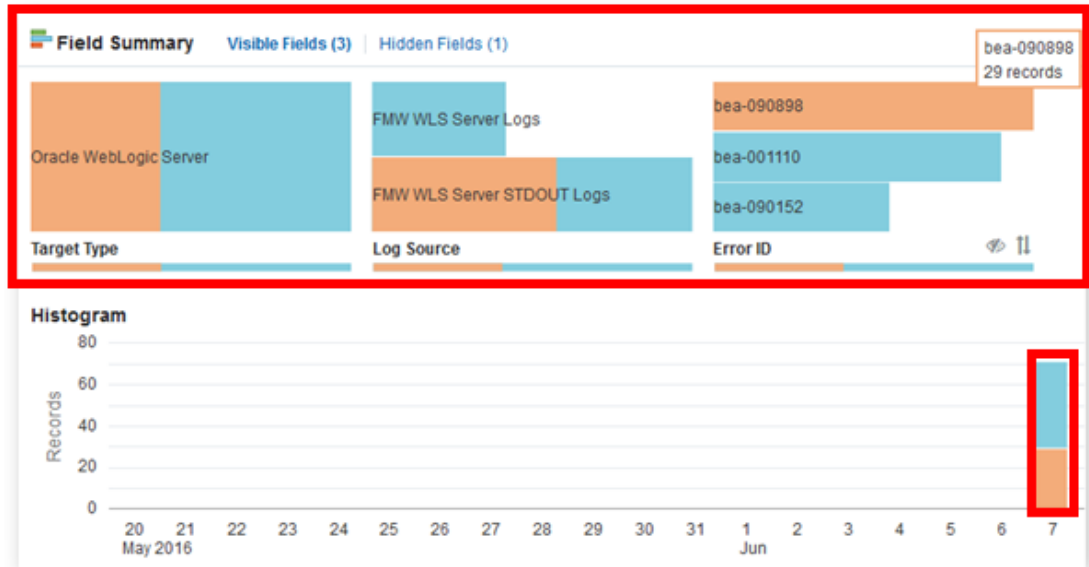
To view the field summary:

1. In Oracle Log Analytics, from the **Visualize** panel, select **Records with Histogram** (.
2. From the **Pinned** section of the **Fields** panel, click **Entity Type**. In the Entity Type dialog box, select the required Entity types, such as **Oracle WebLogic Server**, **Oracle HTTP Server**, and **Database Instance**, and click **Apply**.
3. From the **Pinned** section of the **Fields** panel, click **Log Source**. In the Log Source dialog box, select the required log sources, such as all **FMW WLS Server logs**, and click **Apply**.
4. From the **Other** section of the **Fields** panel, click **Error ID**. In the Error ID dialog box, select the required error IDs, such as **bea-090898**, **bea-001110**, and **bea-090152**, and click **Apply**.
5. From the **Pinned** section of the **Fields** panel, drag and drop **Entity Type**, **Log Source**, and **Error ID** to the **Display Fields** section of the **Visualize** panel.
6. In the **Field Summary** section, click the **Show Facet Summary** () icon.

The **Field Summary** section displays a summary of the search results grouped by **Entity Type**, **Log Source**, and **Error ID**.



Error ID Field Summary



If you click **bea-090898** in the **Error ID** field of the summary results in the field summary, histogram and the table of records change to return only those entity types, log sources, and records that contain bea-090898.



Configure the Display of the Field Summary


You can show or hide fields in the Field Summary section. You can also sort the entries of a field. In addition, you can add more fields in the summary.

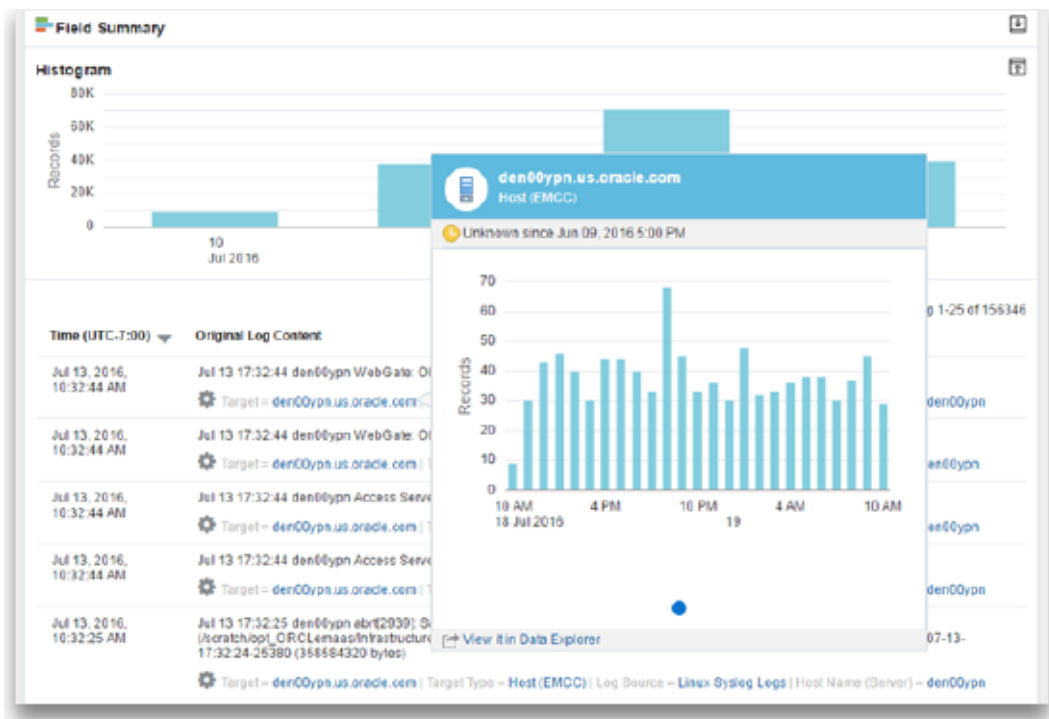
1. Click the **Hide Graph** (👁️) icon on the field that you want to hide from the summary.
The field is no longer displayed in the summary.
To display the hidden field, click the **Hidden Fields** link, and then click the **Show Graph** (👁️) icon on the field that you want to display.
2. Click the **Sort** (⬆️) icon on the field whose entries you want to sort in the ascending or descending order.
3. From the **Pinned** section of the **Fields** panel, drag and drop the **Severity** field to the **Display Fields** section of the **Visualize** panel to add the **Severity** field in the summary.

View an Entity Card

Oracle Log Analytics displays Entity Card which is the information related to specific targets in the form of a histogram. You can access entity-related information easily instead of going to other views or performing a separate search.


The Entity Card visualization displays an entity's status and associated log records (in the form of a histogram), and provides a link to the Data Explorer of Oracle IT Analytics (to view and analyze the target).

1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select **Records With Histogram** ().
3. In the log records section, hover your cursor over a target name to display the floating **View Entity Information** button.
4. Click **View Entity Information** to display the Entity Card.

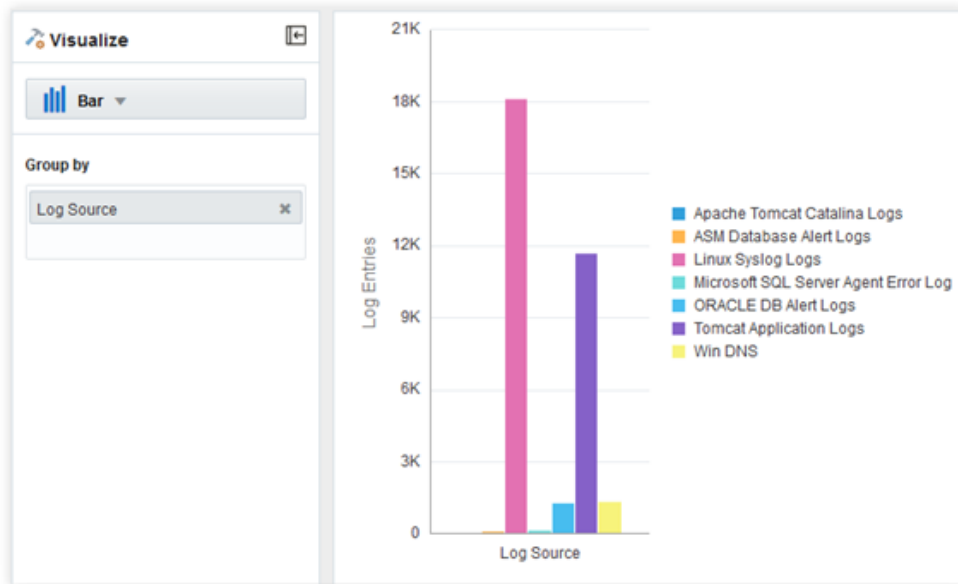


Bar Charts Visualization

You can use bar charts in Oracle Log Analytics to view log records grouped by entities, or log fields.

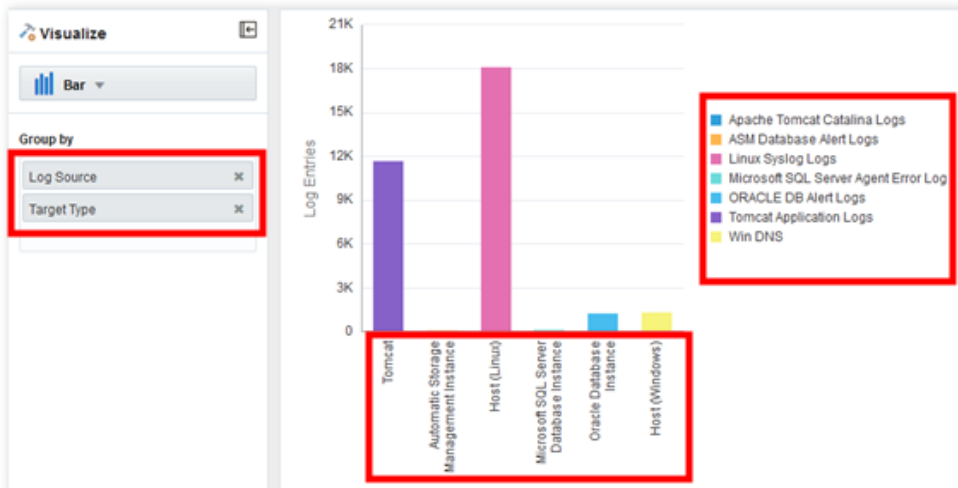
1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select **Bar** ().
3. From the **Pinned** section of the **Fields** panel, drag and drop **Log Source** to the **Group by** section of the **Visualize** panel.

This displays the log records in the form of a bar chart grouped by log sources.



- From the **Pinned** section of the **Fields** panel, drag and drop **Entity Type** to the **Group by** section of the **Visualize** panel.

Now, the bar chart changes to display the log records grouped by log sources in the y-axis against the target types displayed across the x-axis.




 **Note:**

- You can drag and drop a maximum of two fields from the **Fields** panel to the **Group by** section of the **Visualize** panel.
- For the fields with numerical values, you can optionally display their ranges in the bar chart visualization by using the **bucket** option. The bucket option groups the log records into buckets based on the range of values of a field. See [Filter Logs by Field Range](#).


Clusters Visualization

Clustering uses machine learning to identify the pattern of log records, and then to group the logs that have a similar pattern.

Clustering helps significantly reduce the total number of log entries that you have to explore and easily points out the outliers. Grouped log entries are presented as **Sample Message**. You can generate alerts for the cluster utilities like potential issues and outliers by using the *link by clusters* feature. See [Generate Alerts for Cluster Utilities](#).

1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select **Cluster** ()

You can see that similar log records are grouped in clusters along with a histogram view of all the records grouped by time interval. You can zoom in to a particular set of intervals (records grouped by time intervals in this case) in the histogram by keeping your left mouse button pressed and drawing a rectangle over the required set of intervals. After you zoom in, the cluster records change based on the selected interval.






Cluster Help

Clusters are similar log records that are grouped together. Clustering allows you to quickly identify potential issues and outliers by reducing the total number of records that need to be analyzed. Key features are:

- Potential issues are listed first (e.g., clusters containing Fatal, Exception, Error).
- Trend charts show the timeline of log records for a cluster. Similar trend shapes may indicate a further correlation between clusters.
- The sample message is an example of one of the log records. The variables in a sample message can be drilled into to show more details.
- Outliers are single log records that occurred only once during the time period.

In the following example, two of the clusters have similar trend shapes. This may indicate the log records are related and occurring at the same time. The variable values (e.g., SLC2_x86_64_28) can be drilled into for more information.

Trend	Count	Sample Message
	168	Errors in file slc1_j000_26899.trc
	8	Linux-x86_64 Error: 28: No space left on device
	8	ORACLE Instance SLC2 - Archival Error

For more cluster features and best practices, refer to the OMC Documentation [Using Log Analytics Cloud Service](#)

The Cluster view displays a summary banner at the top showing the following tabs:

- **Total Clusters:** Total number of clusters for the selected log records.
- **Potential Issues:** Number of clusters that have potential issues based on log records containing words such as error, fatal, exception, and so on.
- **Outliers:** Number of clusters that have occurred only once during a given time period.

- **Trends:** Number of unique trends during the time period. Many clusters may have the same trend. So, clicking this panel shows a cluster from each of the trends.

 **Note:**

If you hover your cursor over this panel, then you can also see the number of log records (for example, 22 clusters from 1,200 log records).

When you click any of the tabs, the histogram view of the cluster changes to display the records for the selected tab.

Each cluster pattern displays the following:

- **Trend:** This column displays a sparkline representation of the trend (called trend shape) of the generation of log messages (of a cluster) based on the time range that you selected when you clustered the records. Each trend shape is identified by a Shape ID such as 1, 2, 3, and so on. This helps you to sort the clustered records on the basis of trend shapes.

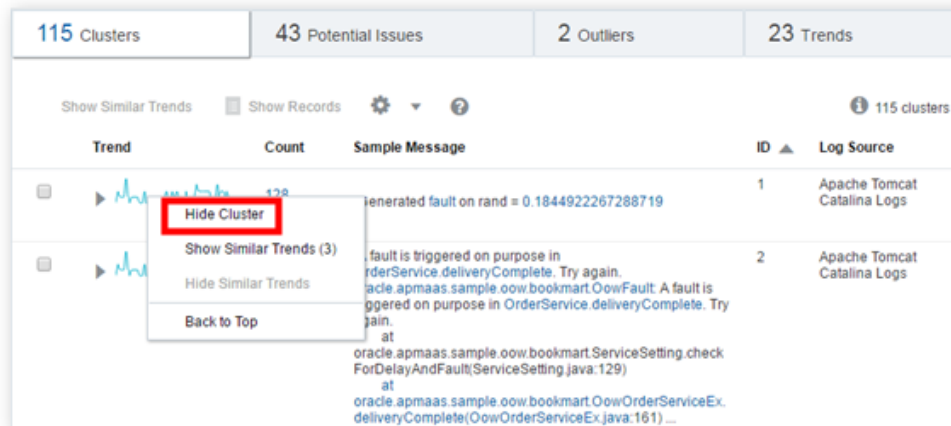
Clicking the arrow to the left of a trend entry displays the time series visualization of the cluster results. This visualization shows how the log records in a cluster were spread out based on the time range that was selected in the query. The trend shape is a sparkline representation of the time series.

- **ID:** This column lists the cluster ID. The ID is unique within the collection.
- **Count:** This column lists the number of log records having the same message signature.
- **Sample Message:** This column displays a sample log record from the message signature.
- **Log Source:** This column lists the log sources that generated the messages of the cluster.

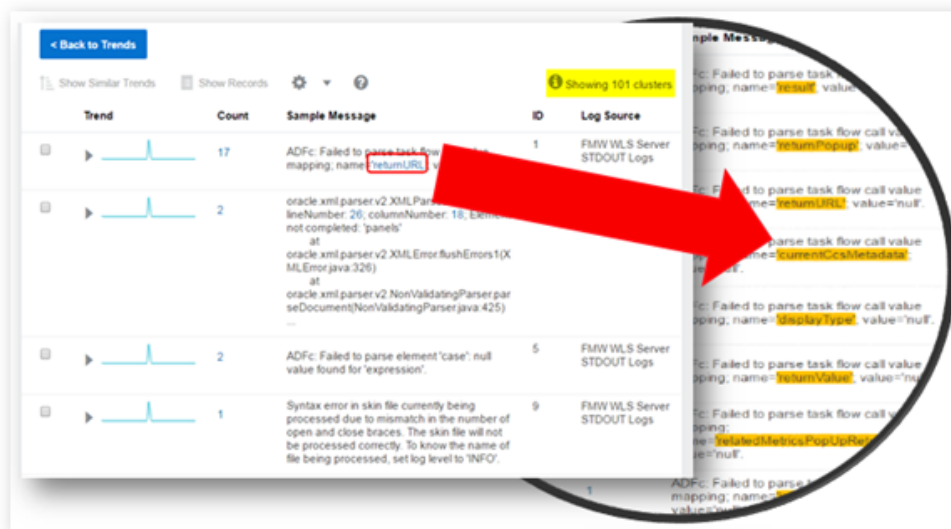


You can click **Show Similar Trends** to sort clusters in an ascending order of trend shapes. You can also select a cluster ID or multiple cluster IDs and click **Show Records** to display all the records for the selected IDs.

You can also hide a cluster message or multiple clusters from the cluster results if the output seems cluttered. Right-click the required cluster and select **Hide Cluster**.



In each record, the variable values are highlighted. You can view all the similar variables in each cluster by clicking a variable in the **Sample Message** section. Clicking the variables shows all the values (in the entire record set) for that particular variable.

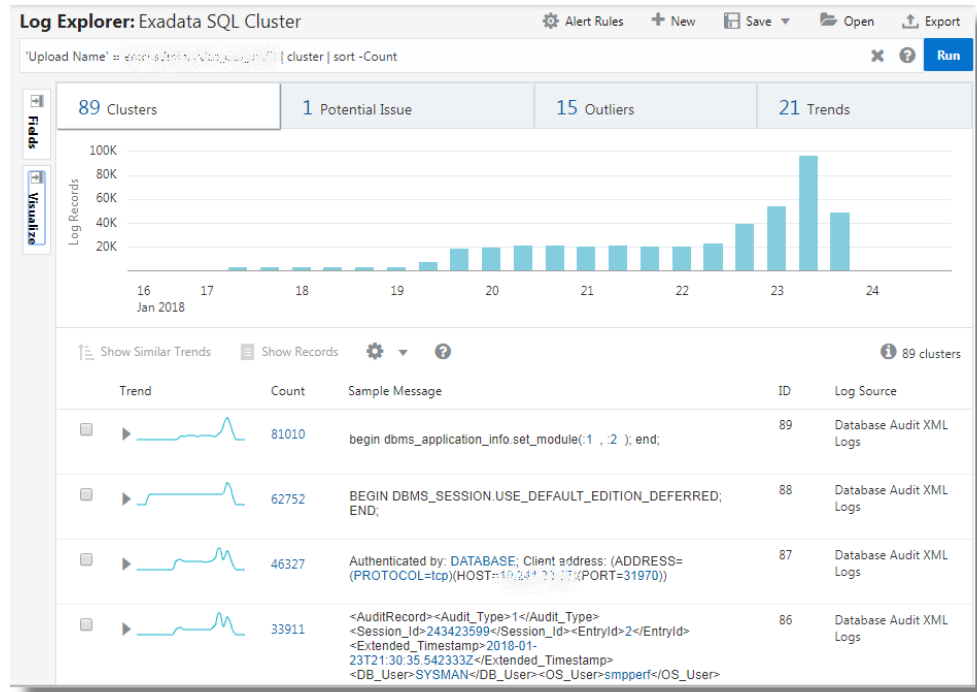


In the Sample Message section, some cluster patterns display a **<n> more samples...** link. Clicking this link displays more clusters that look similar to the selected cluster pattern.

Clicking **Back to Trends** takes you back to the previous page with context (it scrolls back to where you selected the variable to drill down further). The browser back button also takes you back to the previous page; however, the context won't be maintained, because the cluster command is executed again in that case.

Cluster the Log Data Using SQL Fields

Here's an example of clustering the SQL fields:



The large volume of log records are reduced to 89 clusters, thus offering you fewer groups of log data to analyze.

You can drill down the clusters by selecting the variables. For example, from the above set of clusters, select the cluster that has the sample message `SELECT version FROM V$INSTANCE:`

Trend	Count	Sample Message	ID	Log Source
	81010	begin dbms_application_info.set_module(1, :2); end;	89	Database Audit XML Logs
	62752	BEGIN DBMS_SESSION.USE_DEFAULT_EDITION_DEFERRED; END;	88	Database Audit XML Logs
	46327	Authenticated by: DATABASE; Client address: (ADDRESS=(PROTOCOL=tcp)(HOST=10.241.23.37)(PORT=31970))	87	Database Audit XML Logs
	33911	<AuditRecord><Audit_Type>1</Audit_Type><Session_Id>243423599</Session_Id><EntryId>2</EntryId><Extended_Timestamp>2018-01-23T21:30:35.542333Z</Extended_Timestamp><DB_User>SYSMAN</DB_User><OS_User>smpperf</OS_User><Userhost>shw0101010101010101</Userhost><OS_Process>69271</OS_Process><Terminal>unknown</Terminal><Instance_Number>3</Instance_Number><Action>101</Action><Returncode>0</Returncode><DBID>827406061</DBID></AuditRecord></Audit>	86	Database Audit XML Logs
	24567	SELECT version FROM V\$INSTANCE	85	Database Audit XML Logs
	24530	SELECT p.pack, to_number(sys_context('USERENV','CON_ID')) as con_id, sys_context('USERENV','DATABASE_ROLE') as db_role, e.cnt FROM (select nvl(max(upper(value)), 'NONE') AS pack from v\$parameter where name = 'control_management_pack_access' and con_id <= 1) p, (select count(*) as cnt from v\$version where (banner like '%Oracle Database % Enterprise Edition %' OR banner like '%Oracle Database % EE % Perf %')) e	84	Database Audit XML Logs
	15436	SELECT TO_CHAR(FROM_TZ(CAST(md_end_time AS TIMESTAMP), TO_CHAR(systimestamp, 'tzr')) AT TIME ZONE sessiontimezone, 'YYYY-MM-DD HH24:MI:SS') time, md_user_wait_time_pct, md_db_time_ps db_time_users,	83	Database Audit XML Logs

This displays the histogram visualization of the log records containing the specified sample message. You can now analyze the original log content. Click **Back to Cluster** to return to the cluster visualization.

The **Trends** panel shows the SQLs that have similar execution pattern.

The **Outliers** panel displays the SQLs that are rare and different.


Use Cluster Compare Utility

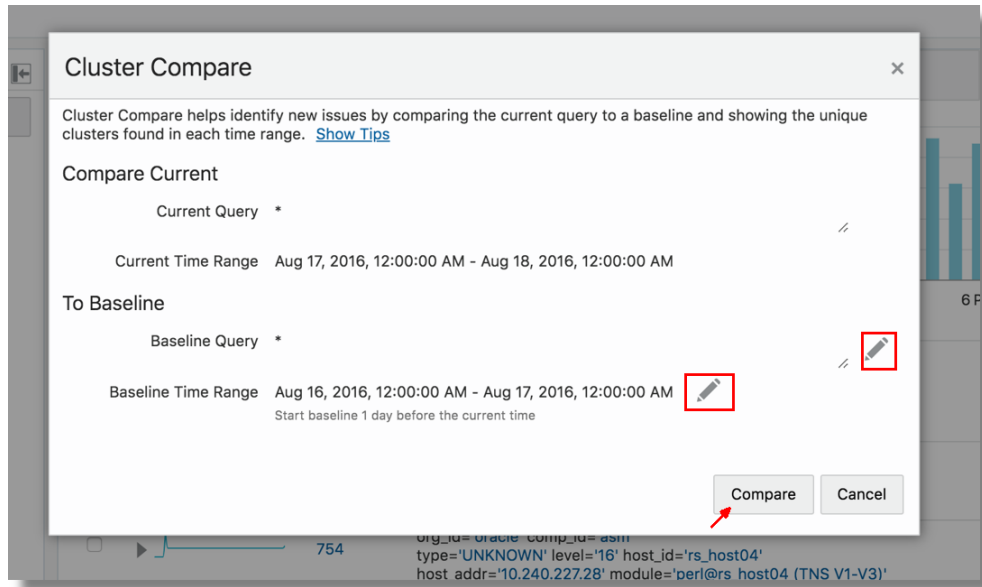
The cluster compare utility can be used to identify new issues by comparing the current set of clusters to a baseline and reducing the results by eliminating common or duplicate clusters. Some of the typical scenarios are:



- What clusters are different in this week compared to last week?
See [Cluster Compare by Time Shift](#).
- What's the difference between the cluster set of entity A and the set of entity B?
See [Cluster Compare by Current Time](#).
- Things were working well in the month X. What changed in this month?
See [Cluster Compare by Custom Time](#).

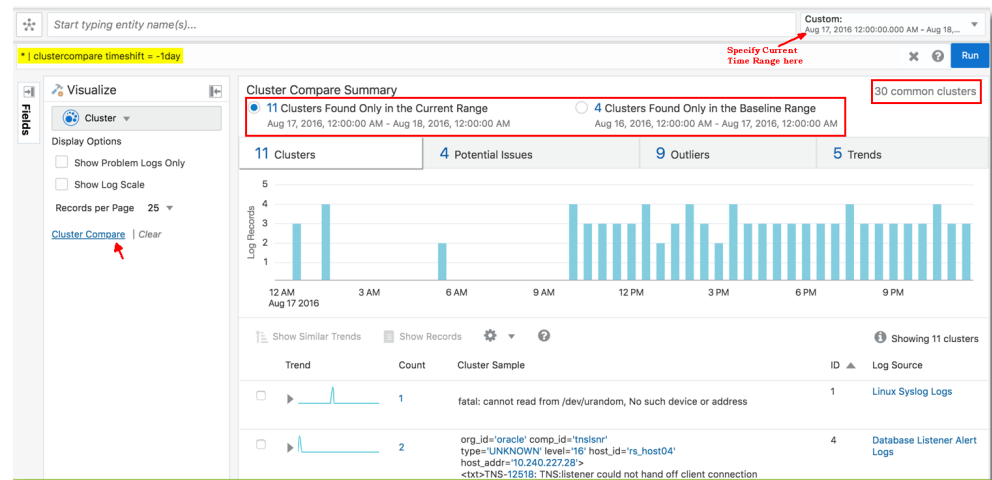
Given two sets of log data, the cluster compare utility removes the data pertaining to the common clusters, and displays histogram data and the records table that are unique to each set. For example, when you compare the log data from week x and week y, the clusters that are common to both the weeks are removed for simplification, and the data unique to each week is displayed. This enables you to identify patterns that are unique for the specific week, and analyze the behavior.

For the syntax and other details of the `clustercompare` command, see Clustercompare Command in *Using Oracle Log Analytics Search*.

1. In clusters  visualization, select your current time range. By default, the query is *. You can refine the query to filter the log data.
2. In the **Visualize** panel, click **Cluster Compare**. The cluster compare dialog box opens.



3. You can notice that the current query and the current time range are displayed for reference.
 - **Baseline Query:** By default, this is the same as your current query. Click the  and modify the baseline query, if required.
 - **Baseline Time Range:** By default, the cluster compare utility uses the **Use Time Shift** option to determine the baseline time range. Hence, the baseline time range is of the same duration as the current time range and is shifted to the period before the current time range. You can modify this by clicking the  icon and selecting **Use Custom Time** or **Use Current Time**. If you select **Use Custom Time**, then specify the custom time range using the menu.
 - Click **Compare**. You can now view the cluster comparison between the two log sets.



Click the button corresponding to each set to view the details like clusters, potential issues, outliers, trends, and records table that are unique to the set. The page also displays the number of clusters that are common between the two log sets.

In the above example, there are 11 clusters found only in the current range, 4 clusters found only in the baseline range, and 30 clusters common in both the ranges. The histogram for the current time range displays the visualization using only the log data that is unique to the current time range.

Note:

Clusters found only in the current range are returned first, followed by clusters found only in the baseline range. The combined results are limited to 500 clusters. To reduce the cluster compare results, reduce the current time range or append a command to limit the number of results. For example, append `| head 250` will limit both current and baseline clusters to 250 each. Use multi-select (click and drag hold) on the cluster histogram to reduce the current time range when using the custom time option. The time range shift value may be converted to minutes or seconds to ensure no time gaps or overlaps occur between the current and baseline time ranges.

Use Dictionary Lookup in Cluster

Use dictionary lookup after the `cluster` command to annotate clusters.

Consider the `cluster` results for **Fusion Middleware WebLogic Server Logs**. To define a dictionary to add labels based on the **Cluster Sample** field:

1. Create a CSV file with the following contents:

```
Operator,Condition,Issue,Area
CONTAINS REGEX,[Mm]alformed request .null.\.\s+Request parsing
failed,Parsing Error,Request Processing
CONTAINS,Failed to associate the transaction context with the response
while marshalling,Marshalling Error,Response
CONTAINS,A RuntimeException was generated by the RMI server,Exception,RMI
```

CONTAINS,unable to establish JMX Connectivity,Connection Error,JMX
CONTAINS REGEX,Can not locate \S+ for now. DMS will,DMS Search
Error,DMS

Import this as a Dictionary type lookup using the name **WLS Error Categories**. This lookup contains two fields, *Issue* and *Area* that can be returned on a matching condition. See [Create a Dictionary Lookup](#).

2. Use the dictionary in cluster to return a field:

Run the `cluster` command for the FMW WLS Server Logs. Add a `lookup` command after `cluster`, as shown below:

```
'Log Source' = 'FMW WLS Server Logs'
| cluster
| lookup table = 'WLS Error Categories' select Issue using 'Cluster Sample'
```

The value of **Cluster Sample** for each row is evaluated against the rules defined in the WLS Error Categories dictionary. The *Issue* field is returned from each matching row.

3. Return more than one field by selecting each field in the `lookup` command:

```
'Log Source' = 'FMW WLS Server Logs'
| cluster
| lookup table = 'WLS Error Categories' select Issue as Category, Area using 'Cluster Sample'
```

The above query selects the *Issue* field, and also renames it to *Category*. *Area* field is also selected, but not renamed.

4. Filter the cluster results using the dictionary fields:

Use the `where` command on the specific fields to filter the clusters. Consider the following query:

```
'Log Source' = 'FMW WLS Server Logs'
| cluster
| lookup table = 'WLS Error Categories' select Issue as Category, Area using 'Cluster Sample'
| where Area in (RMI, Messaging)
```

This displays only those records that matches the specified values for *Area* field.

Line Charts Visualization

Oracle Log Analytics provides a line chart visualization that lets you group by a numeric field, then graph the trend of values over time. Only numeric fields or aggregate output (sum as sum, average if the fields or output can be broken up by time) can be selected for the y-axis.

For example, to view the count of all Apache log entries, which are grouped by log source, over a period in time (say the last 7 days):

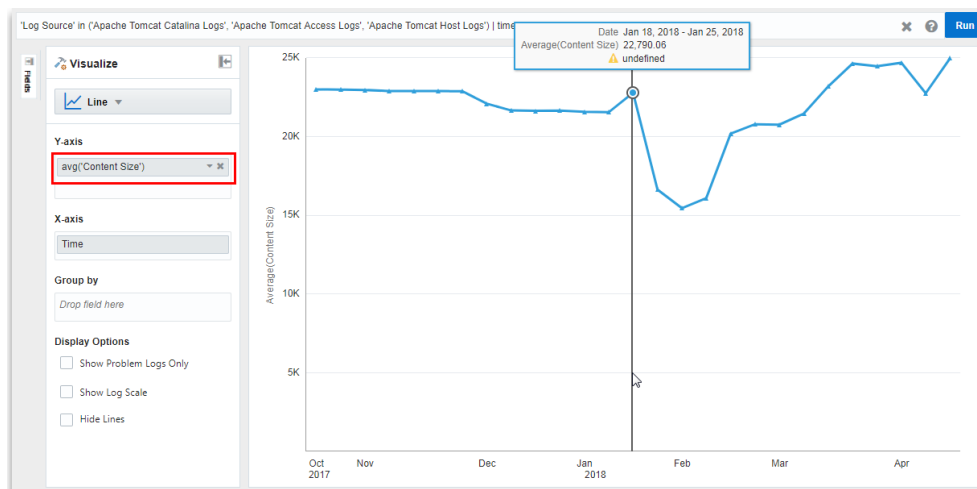
1. In Oracle Log Analytics, set the time control on the top right corner to **Last Week**.
2. In the **Pinned** section of the **Fields** panel, click **Log Source**, select all the Apache log sources, and click **Apply**.
3. In the **Visualize** panel, select **Line** (📈) from the visualization options.
 - You can select any time stamp field along x-axis for analysis.
 - You can select multiple fields to display on the y-axis by dragging them to the *Group By* region. Each field is shown with a different marker shape on the line chart.

In case of multiple data points at a specific time in a line series, you can select the **Hide Lines** check box to view the data points by hiding the lines.

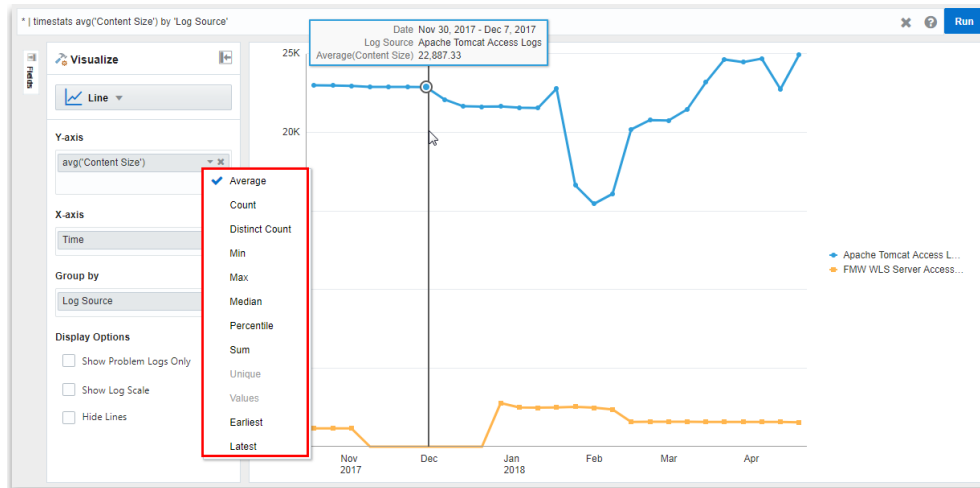
The count of Apache log entries grouped by Log Source is displayed.



Similarly, to view a graph displaying the average content size of the Apache log records over time, grouped by log source, from the **Other** section of the **Fields** panel, drag and drop the **Content Size** field to the **Y-axis** section in the **Visualize** panel. Ensure that you've selected the **Average** function from the **Y-axis** list.



You can select different statistical operations that you can perform on the selected field.



Note:

Some fields may not support all operations. For example, numeric operations, such as Average, Min, Max, Median, and so on won't work on fields that are of data type `String`.

Maps Visualization

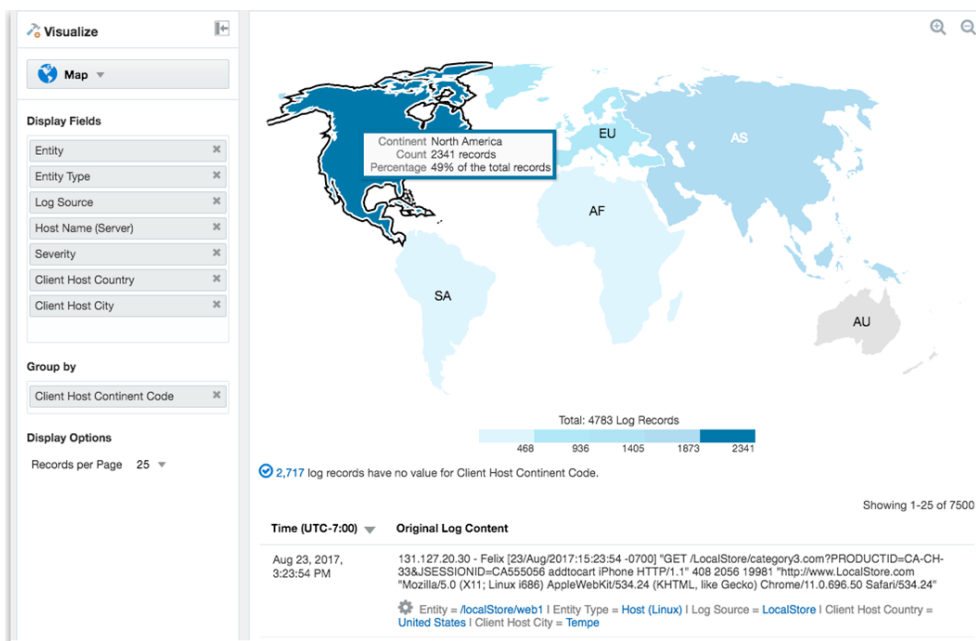
You can use the Maps visualization in Oracle Log Analytics to view log records grouped by country or country code.

Before you can use Maps to view log records based on country or country codes, set the Field Enrichment options to populate the city, country, or country code fields under the Log Source section from the Oracle Log Analytics Configuration page. See [Geolocation Lookup](#).

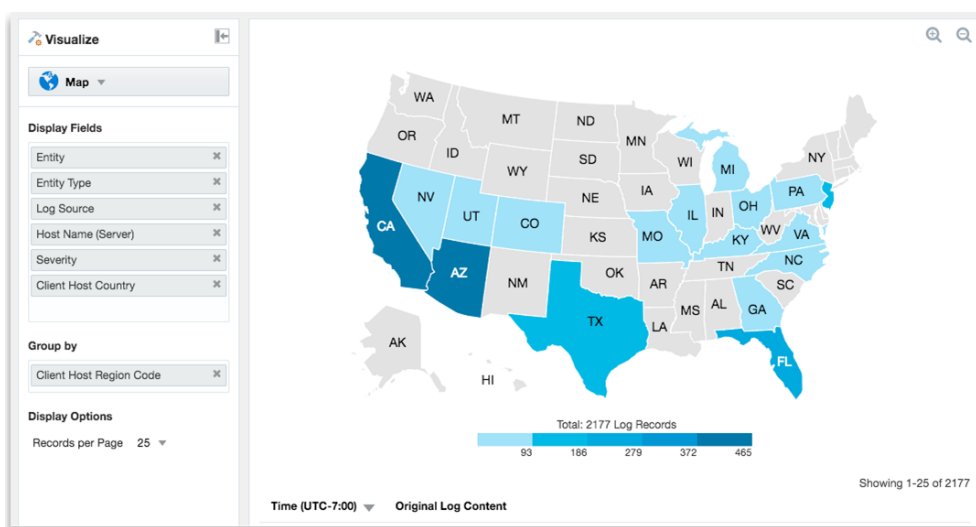
1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select **Map** (🌐).

This displays a world map where log records are grouped by **Client Coordinates**, **Client Host Continent**, **Client Host Continent Code**, **Client Host Country**, **Client Host Country Code**, **Client Host City**, **Client Host Region**, or **Client Host Region Code**.

The following example shows the map where the log records are grouped by continent:



The following example shows the map where the log records are grouped by country:




Summary Tables

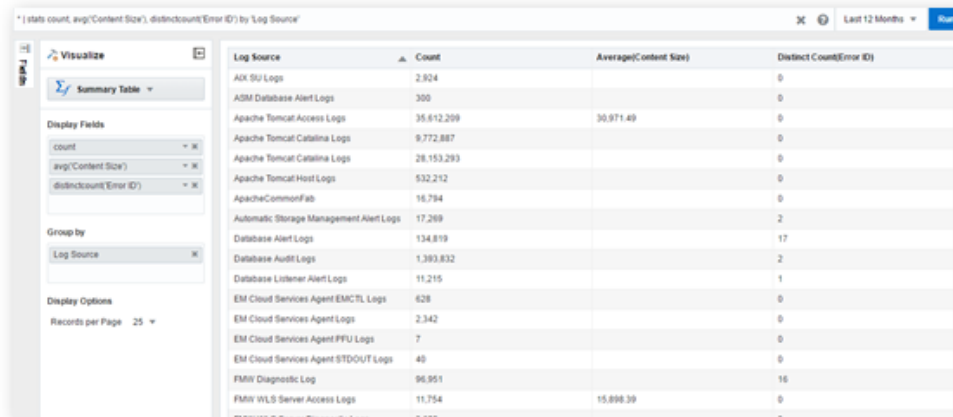
Using a Summary Table, you can view statistical information about log records in a tabular format.

You can select which measures you want to see, as well as the fields on which to base those measures. Currently, the aggregate functions available in Summary Table are average, count, distinct count, sum, min, max, median, percentile, stddev, values, earliest, and latest. For `timstats` command, the aggregate functions like `persecond`, `perminute`, `perhour`, and `perday` are additionally available.

You can also group the results by any selected fields.

1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select **Summary Table** (.
3. From the **Fields** section, drag and drop the required fields.
4. Click the down arrow to the right of the selected fields to select the function.

The summary table displays the required result.



Log Source	Count	Average(Content Size)	Distinct Count(Error ID)
AIX SU Logs	2,924		0
ASM Database Alert Logs	300		0
Apache Tomcat Access Logs	35,612,209	30,971.49	0
Apache Tomcat Catalina Logs	9,772,887		0
Apache Tomcat Catalina Logs	28,153,293		0
Apache Tomcat Host Logs	532,212		0
ApacheCommonFab	16,794		0
Automatic Storage Management Alert Logs	17,269		2
Database Alert Logs	134,819		17
Database Audit Logs	1,393,832		2
Database Listener Alert Logs	11,215		1
EM Cloud Services Agent EMCTL Logs	628		0
EM Cloud Services Agent Logs	2,342		0
EM Cloud Services Agent PFU Logs	7		0
EM Cloud Services Agent STDOU Logs	40		0
FMIN Diagnostic Log	96,951		16
FMIN WLS Server Access Logs	11,754	15,898.39	0
...


To drill-down to a specific value which is used in the group by part of your query, click on the value in the summary table.

Note:

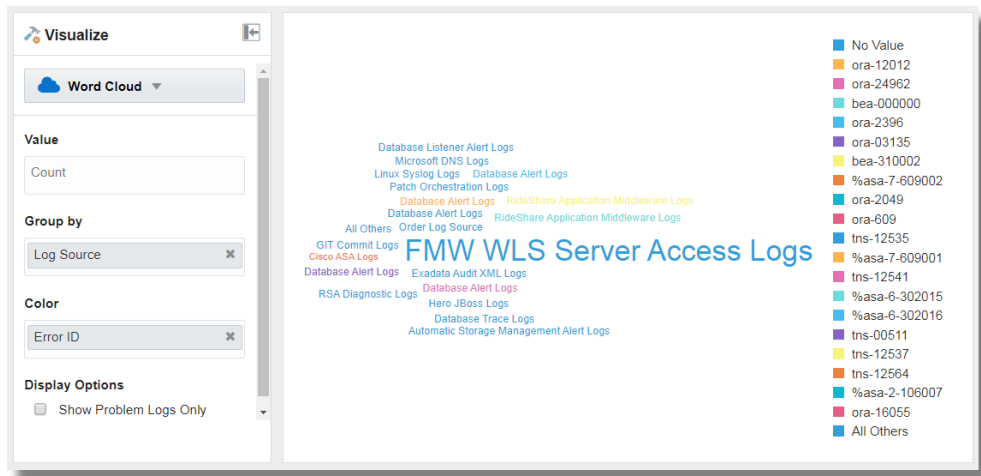
Apart from a Summary Table, all the graph and chart visualization options let you apply multiple statistical functions to your log records.

Word Cloud Visualization

You can use word cloud in Oracle Log Analytics to view log records grouped by the strings that represent the selected fields.

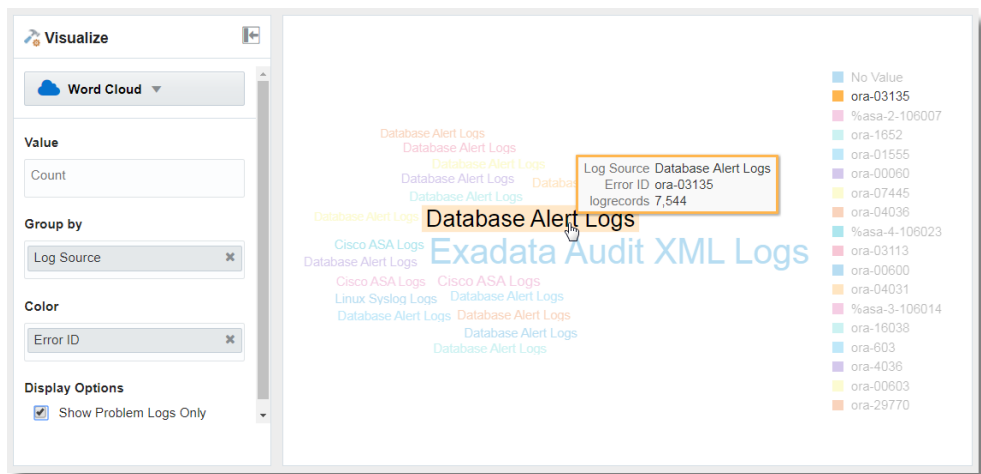
1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. From the **Visualize** panel, select **Word Cloud** (.
3. From the **Pinned** section of the **Fields** panel, drag and drop **Log Source** to the **Group by** section of the **Visualize** panel.
4. From the **Other** section of the **Fields** panel, drag and drop **Error ID** to the **Color** section of the **Visualize** panel.

This displays the log records in the form of a word cloud grouped by log sources. The log sources are represented in different colors to indicate the Error ID reported in the records.



- To view the word cloud using only the log records that reported error, in the **Visualize** panel, select **Show Problem Logs Only** check box.

Now, the word cloud changes to display only the problem logs.



Hover the cursor on a string to get more details about the group that the string represents.

Click the string to view the further analysis of the group of log records displayed as the records with histogram visualization.

Link Visualization

Link lets you perform advanced analysis of log records by combining individual log records from across log sources into groups, based on the fields you've selected for linking. You can analyze the groups by using the same fields as the ones you used for linking or additional fields for observing unusual patterns to detect anomalies.

Link command can be used for a variety of use-cases. For example, individual log records from business applications can be linked to synthesize business transactions. Groups can also be used to synthesize user sessions from web access logs. Once these linked records

have been generated, they can be analyzed for anomalous behavior. Some examples of this anomalous behavior can include:

- Business Transactions that are taking unusually long to execute or are failing.
- User sessions that are downloading large amounts of data than normal.

 **Tip:**

To use the Link feature, users need to have a good understanding of their log sources. The Link feature relies on a field or a set of fields that are used to combine individual log records. To generate meaningful associations of log records, it is important to know the relevant fields that can be used for linking the log records.

To understand the application of Link in performing advanced analytics and its advanced features, see [Perform Advanced Analytics with Link](#). These are the features highlighted in the use cases:

- *Link Trend*
- *Generating charts with virtual fields*
- *Using SQL statement as a field of analysis*
- *Generating charts for multiple fields and their values*
- *Second level aggregation*
- *Time analysis*
- *Navigation functions*

Analyze the Log Records Using Link

You can use the example of the log records from the log source `SOAOrderApp` for an order flow application, to apply the steps discussed below. Note that the following steps introduce you to the basic features of link. After familiarizing with the steps, here are some of the simple features you can use for convenience and better experience with link:

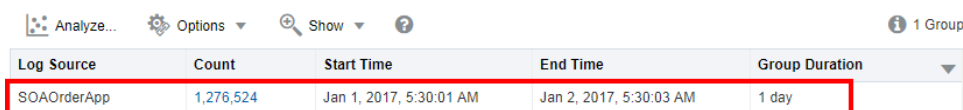
- [Use Dictionary Lookup in Link](#)
- [Semantic Clustering Using Natural Language Processing](#)
- [Generate Link Alerts](#)
- [Use the Getting Started Panel](#)
- [Features for Bubble Charts in Link Analysis](#)
 - [Change the Title of the Bubble Chart](#)
 - [Control the Color of the Bubbles in the Chart](#)
- [Features for Fields in Link Analysis](#)
 - [Add More than Two Fields](#)
 - [Rename the Fields by Editing the Query](#)
 - [Add More Fields for Analysis Using Size and Color](#)
 - [Instant Analysis of Multiple Fields Using the Link Analyzer Chart](#)

- Features for Groups in Link Analysis
 - Change the Group Alias
 - Join Multiple Groups Using the Map Command
 - Create Sub-Groups Using the Createview Command
 - Search and Highlight Link Groups

1. Select **Link**  from the **Visualize** panel.

By default, **Log Source** is used in the **Link By** field to run the `link` command. This displays the groups table. See [Groups Table](#).

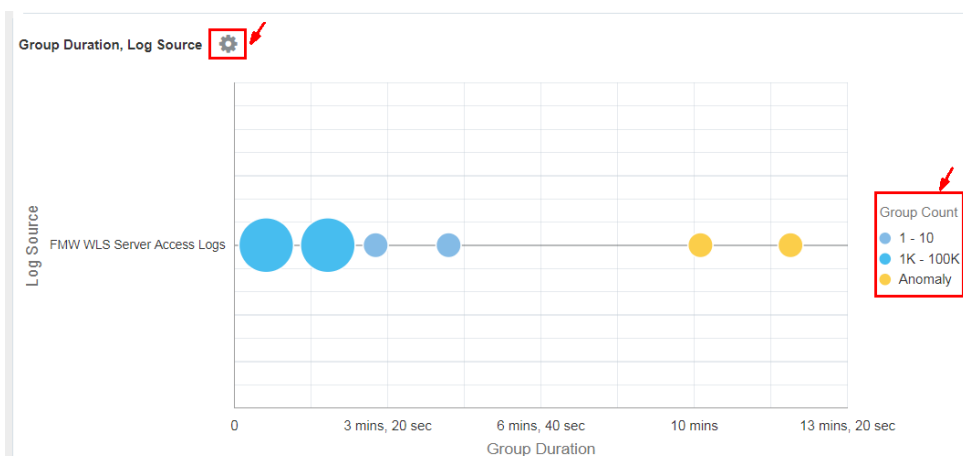
For example, the following groups table is displayed for `SOAOrderApp`:



Log Source	Count	Start Time	End Time	Group Duration
SOAOrderApp	1,276,524	Jan 1, 2017, 5:30:01 AM	Jan 2, 2017, 5:30:03 AM	1 day

2. By default, the **Group Duration** column is not included in the groups table. To include it, click **Options** > **Hide/Show Columns** > Check **Group Duration**.
3. To analyze the fields that are relevant to your analysis, drag and drop one or more fields to **Link By**, remove **Log Source** which is the default field in **Link By**, and click the check mark to run the **Link** query. You can view the updated groups table.
4. To include more columns in the table, drag and drop the fields of interest into the **Display Fields** section. This is equivalent to the `stats` command. You can add alias to any of the fields by editing the query and using `as` to display the field with a new alias. For example, `stats avg('Elapsed Time (Real)') as 'Avg Time'`.
5. To visualize the groups and to analyze the log records using a bubble chart, click **Analyze** > select the fields for analysis. For example, select `Group Duration` and `Log Source`. The same action can also perform using the `classify` command.

You can view the groups represented in the bubbles in the chart.



This analyzes the groups for the values of the fields, and creates bubbles representing the groups in the commonly seen ranges. The majority of the values are treated as the


baseline. For example, a large bubble can become the baseline, or a large number of smaller bubbles clustered together can form the baseline. Bubbles that are farthest from the baseline are typically marked as anomalies. Generally, these bubbles represent the behavior that is not typical.

This chart shows the anomalies in the patterns indicated by the yellow bubbles. The size of the bubble represents the number of groups that are contained in the bubble. The position of the bubble is determined by the values of the fields that are plotted along the x and y axes. Hover the cursor on the bubbles to view the number of groups in the bubble, their percentage as against the total number of groups, and the values of the fields plotted along the x and y axes.

You can hover the cursor on the filter legend to get more information. See [Additional Information in Analyze Chart](#).

Note:

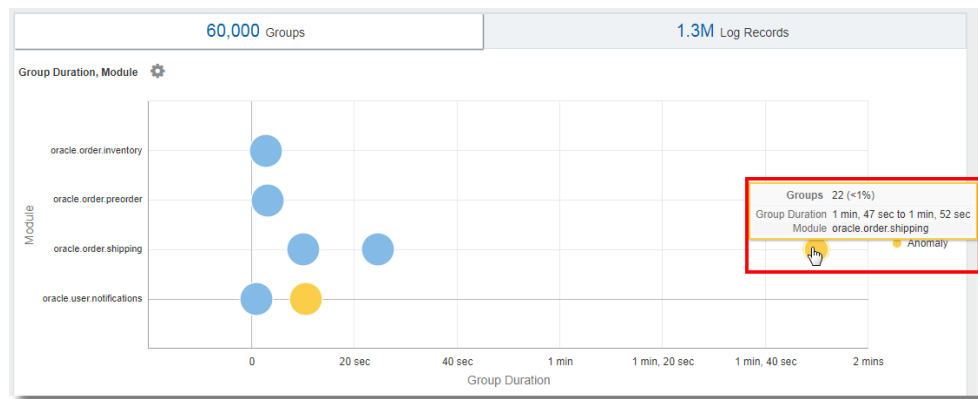
When you run the `link` command, the group duration is shown in a readable format in the bubble chart, for example, in *minutes* or *seconds*. However, if you want to run a `where` command after the `link` command to look for transactions that took more than the specified number of seconds (say, 200 seconds), then the unit that you must use is *milliseconds*.

The next step may be to further examine the anomalies by clicking individual bubble or multi-select the bubbles. To return to the original results after investigating the bubble, click the **Undo**  icon.

You can toggle the display of the groups on the bubble chart by clicking on the value of the **Group Count** legend that's available next to the chart. This can be used to reduce the number of bubbles displayed on a densely packed chart.

From the order flow application:

- a. We've selected the fields **Module** and **Context ID** to group the log records. This groups the log records based on the context ID of each record and the specific module from shipping, notifications, inventory or preorder that was used by the application in the log record.



The chart displays the bubbles that group the log records based on their values of Context ID and Module. The blue bubbles represent most of the groups that form the baseline. Notice the two anomaly bubbles that appear on the chart against the modules for shipping and notifications. The bubble on the extreme right of the chart represents the groups that're taking a longer duration to execute the module as compared to other groups. On hovering the cursor on the bubble, you can observe that the bubble consists of 22 groups that make for less than a percent of the total number. The bubble corresponds to the `oracle.order.shipping` module and has the group duration of 1 min, 47 sec to 1 min, 52 sec.

You can generate alerts to notify you when anomalies are detected in your log records. See [Generate Link Alerts](#).

- b. To view the details of the groups that correspond to the anomaly, select the anomaly bubble in the chart.
 - In the next tab, a histogram chart is displayed showing the dispersion of the log records.
 - A groups table listing each of the 22 groups and the corresponding values of the fields is also available for the analysis.

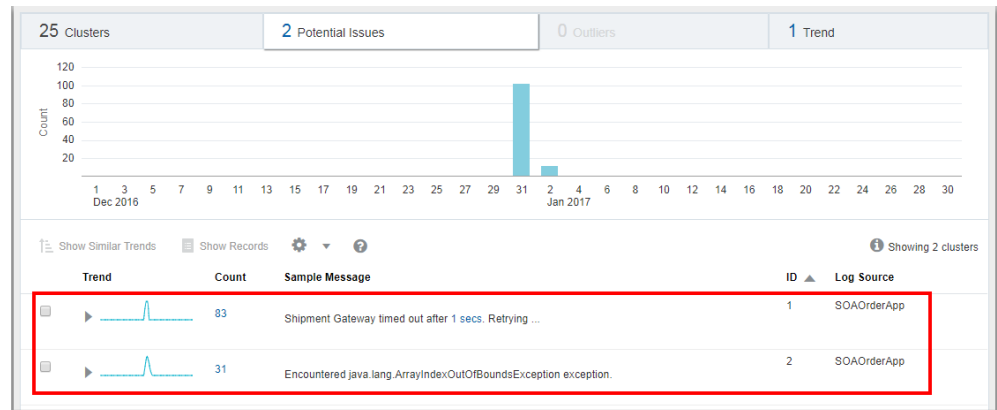
Module	Context ID	Clusters	Start Time	End Time	Group Duration
oracle.order.shipping	103533		Jan 2, 2017, 1:49:10 AM	Jan 2, 2017, 1:51:02 AM	1 min, 52 sec
oracle.order.shipping	113253	30	Jan 2, 2017, 3:17:18 AM	Jan 2, 2017, 3:19:10 AM	1 min, 52 sec
oracle.order.shipping	104414	32	Jan 2, 2017, 12:43:06 AM	Jan 2, 2017, 12:44:58 AM	1 min, 52 sec
oracle.order.shipping	100510	97	Jan 1, 2017, 12:38:32 PM	Jan 1, 2017, 12:40:24 PM	1 min, 52 sec
oracle.order.shipping	107060	21	Jan 1, 2017, 4:47:34 PM	Jan 1, 2017, 4:49:26 PM	1 min, 52 sec
oracle.order.shipping	111395	41	Jan 1, 2017, 7:26:09 PM	Jan 1, 2017, 7:28:00 PM	1 min, 51 sec
oracle.order.shipping	111590	55	Jan 1, 2017, 3:42:56 PM	Jan 1, 2017, 3:44:47 PM	1 min, 51 sec
oracle.order.shipping	107749	38	Jan 1, 2017, 10:13:00 AM	Jan 1, 2017, 10:14:51 AM	1 min, 51 sec
oracle.order.shipping	106290	25	Jan 1, 2017, 7:29:56 AM	Jan 1, 2017, 7:31:47 AM	1 min, 50 sec
oracle.order.shipping	110895	42	Jan 1, 2017, 11:29:33 AM	Jan 1, 2017, 11:31:24 AM	1 min, 50 sec
oracle.order.shipping	108782	38	Jan 2, 2017, 4:32:49 AM	Jan 2, 2017, 4:34:40 AM	1 min, 50 sec
oracle.order.shipping	106223	38	Jan 1, 2017, 8:30:38 AM	Jan 1, 2017, 8:32:29 AM	1 min, 50 sec
oracle.order.shipping	113418	31	Jan 1, 2017, 5:56:40 PM	Jan 1, 2017, 5:58:30 PM	1 min, 50 sec
oracle.order.shipping	107729	23	Jan 1, 2017, 5:05:26 PM	Jan 1, 2017, 5:07:17 PM	1 min, 50 sec
oracle.order.shipping	110275	24	Jan 2, 2017, 4:30:27 AM	Jan 2, 2017, 4:32:17 AM	1 min, 50 sec
oracle.order.shipping	114937	36	Jan 1, 2017, 12:56:17 PM	Jan 1, 2017, 12:58:07 PM	1 min, 50 sec
oracle.order.shipping	101978	30	Jan 1, 2017, 12:52:22 PM	Jan 1, 2017, 12:54:12 PM	1 min, 49 sec
oracle.order.shipping	103735	44	Jan 1, 2017, 7:37:14 AM	Jan 1, 2017, 7:39:03 AM	1 min, 48 sec
oracle.order.shipping	108379	33	Jan 2, 2017, 4:57:15 AM	Jan 2, 2017, 4:59:04 AM	1 min, 48 sec
oracle.order.shipping	104332	24	Jan 1, 2017, 12:43:25 PM	Jan 1, 2017, 12:45:14 PM	1 min, 48 sec
oracle.order.shipping	104622	36	Jan 1, 2017, 12:50:45 PM	Jan 1, 2017, 12:52:33 PM	1 min, 48 sec
oracle.order.shipping	100136	42	Jan 1, 2017, 10:46:33 PM	Jan 1, 2017, 10:48:21 PM	1 min, 47 sec

- c. **View the anomaly groups in clusters:** First select all the rows in the table by clicking on the first row, hold **Shift** key on your keyboard, and click on the last row in the table, next click the down arrow next to **Show**, and select **Clusters**.

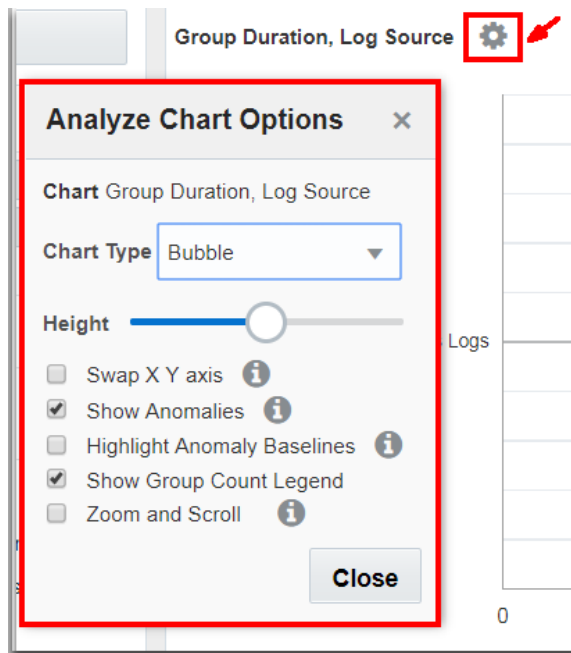
This displays the clusters. Click on the **Potential Issues** tab.

This lists the groups of log records and the sample messages indicating the anomaly. The issues point at **Shipment Gateway time out** and

java.lang.ArrayIndexOutOfBoundsException exception for the cause of delays in executing the shipping module in the specific groups.



- For more options to view the groups, click the **Chart Options** icon on the top left corner of the visualization panel. See [Analyze Chart Options](#).



- Study the groups table to understand the groups and the values of the fields in each group. See [Groups Table](#).

In line with the observation in the bubble chart of the `SOAOrderApp` log records, from the groups table, notice that the top two groups are taking 1 min, 52 sec and 1 min, 51 sec to complete the execution. This is very high compared to the group duration of the other groups.

Analyze... Options Show ? Showing 500 of 60,000 Groups

Module	Context ID	Count	Start Time	End Time	Group Duration
oracle.order.shipping	100510	97	Jan 1, 2017, 12:38:32 PM	Jan 1, 2017, 12:40:24 PM	1 min, 52 sec
oracle.order.shipping	111590	55	Jan 1, 2017, 3:42:56 PM	Jan 1, 2017, 3:44:47 PM	1 min, 51 sec
oracle.order.shipping	110367	48	Jan 1, 2017, 6:16:16 PM	Jan 1, 2017, 6:16:41 PM	25.4 sec
oracle.order.shipping	111095	45	Jan 1, 2017, 1:03:08 PM	Jan 1, 2017, 1:03:33 PM	25.4 sec
oracle.order.shipping	100305	45	Jan 1, 2017, 8:22:25 PM	Jan 1, 2017, 8:22:50 PM	25.4 sec
oracle.order.shipping	110167	46	Jan 1, 2017, 11:56:28 PM	Jan 1, 2017, 11:56:53 PM	25.3 sec
oracle.order.shipping	104112	46	Jan 1, 2017, 6:15:08 AM	Jan 1, 2017, 6:15:33 AM	25.3 sec
oracle.order.shipping	107407	46	Jan 1, 2017, 7:04:03 AM	Jan 1, 2017, 7:04:28 AM	25.3 sec
oracle.order.shipping	114670	46	Jan 1, 2017, 12:10:51 PM	Jan 1, 2017, 12:11:16 PM	25.2 sec
oracle.order.shipping	107567	45	Jan 2, 2017, 1:09:26 AM	Jan 2, 2017, 1:09:51 AM	25.2 sec
oracle.order.shipping	103652	45	Jan 2, 2017, 4:30:04 AM	Jan 2, 2017, 4:30:29 AM	25.1 sec
oracle.order.shipping	103807	46	Jan 1, 2017, 7:27:41 AM	Jan 1, 2017, 7:28:06 AM	25.1 sec
oracle.order.shipping	106477	47	Jan 1, 2017, 12:09:45 PM	Jan 1, 2017, 12:10:10 PM	25.1 sec
oracle.order.shipping	108191	46	Jan 1, 2017, 11:03:22 PM	Jan 1, 2017, 11:03:47 PM	25.1 sec
oracle.order.shipping	113307	48	Jan 1, 2017, 8:35:22 PM	Jan 1, 2017, 8:35:47 PM	25 sec
oracle.order.shipping	107514	47	Jan 1, 2017, 10:42:18 AM	Jan 1, 2017, 10:42:43 AM	25 sec
oracle.order.shipping	106841	47	Jan 1, 2017, 3:58:24 PM	Jan 1, 2017, 3:58:49 PM	25 sec
oracle.order.shipping	105296	45	Jan 2, 2017, 12:34:00 AM	Jan 2, 2017, 12:34:25 AM	25 sec
oracle.order.shipping	102598	48	Jan 1, 2017, 9:51:23 AM	Jan 1, 2017, 9:51:48 AM	24.9 sec
oracle.order.shipping	104331	45	Jan 1, 2017, 7:34:59 PM	Jan 1, 2017, 7:35:24 PM	24.9 sec
oracle.order.shipping	110678	48	Jan 1, 2017, 10:11:22 PM	Jan 1, 2017, 10:11:47 PM	24.9 sec
oracle.order.shipping	107991	46	Jan 2, 2017, 4:23:12 AM	Jan 2, 2017, 4:23:37 AM	24.9 sec
oracle.order.shipping	106689	48	Jan 1, 2017, 8:46:53 PM	Jan 1, 2017, 8:47:18 PM	24.8 sec
oracle.order.shipping	102970	47	Jan 1, 2017, 7:23:00 PM	Jan 1, 2017, 7:23:25 PM	24.8 sec
oracle.order.shipping	109347	46	Jan 1, 2017, 4:12:22 PM	Jan 1, 2017, 4:12:47 PM	24.8 sec

Page 1 of 20 (1-25 of 500 items) K < 1 2 3 4 5 ... 20 > >


8. Click the **Search and Table Options** icon:

- Click **Hide/Show Columns** and select the columns that you want to view in the table.
- Click **Display Options**:
 - **Chart Options**: Select the check box to view the Analyze and Histogram sections together
 - **Summary Options**: You can specify a format for the summary
 - **Alias Options**: Rename the groups and log records to create custom dashboards.
 - **Dashboard Options**: You can select one or more Link visualization sections like **Header**, **Summary**, **Analyze**, **Histogram**, and **Data Table** to be visible in the Dashboard widget. Make these selections before you save a Link query as a Saved Search widget.
- Click **Search Options**:
 - Select the **Show Top** check box, and identify the number of log records to view for the specified field.
 - Select the **Include Nulls** check box to view those log records that may not have all the Link By fields.
 - Under **Analyze Chart Behavior on Selection**,


- * To view the filtered group table for the groups in the selected bubble, click the **Filter Only - filter group table only** option.
- * To view the filtered group table and the re-classified bubble chart for the groups in the selected bubble, click the **Drill Down - filter group table and re-classify bubbles** option.

 **Note:**

The filtered selection is not supported in the saved searches. However, you can open the saved search and apply the same filter selection again.

9. To change the fields analyzed from the group data, click the **Analyze**  icon and select fields that have multiple values with high cardinality. By default, the first field selected for **Link By** is analyzed with the group duration to generate the analyze chart and the groups table. Click **OK**.

This displays a new chart based on the fields selected in the Analyze command.

10. To view the log records in the histogram visualization, click the **histogram** tab. The histogram chart displays the log records over time. Click the down arrow next to the **Chart options**  icon and select the type of visualization to view the data from the log records and groups on separate histograms, if necessary. See [Histogram Chart Options](#).

To generate charts for multiple fields and their values, see [Generate Charts for Multiple Fields and their Values](#).

You can save your custom query for the analysis of the log records using the Link feature to the saved searches and dashboard. See [Save and Share Log Searches](#).

For the syntax and other details of the commands used in the link visualization, see the following in *Using Oracle Log Analytics Search*:

- Addfields Command
- Classify Command
- Eventstats Command
- Link Command

Use Dictionary Lookup in Link

Similar to cluster, you can use a `lookup` command to annotate the Link results.

Consider the Link results for FMW WLS Server Access Logs. To use the dictionary lookup to provide names for different pages:

1. Create a CSV file with the following contents:

```
Operator,Condition,Name
CONTAINS,login,Login Page
CONTAINS,index,Home Page
CONTAINS ONE OF REGEXES,"[\.sh$,\.jar$]",Script Access
```

Import this as a Dictionary type lookup using the name **Page Access Types**. This lookup contains one field, *Name* that can be returned from each matching row. See [Create a Dictionary Lookup](#).

2. Use the dictionary in link:

Add a `lookup` command after `link`, as follows:

```
'Log Source' = 'FMW WLS Server Access Logs'  
| link URI, Status  
| lookup table = 'Page Access Types' select Name using URI
```

The value of **URI** field for each row is evaluated against the rules defined in the *Page Access Types* dictionary. The **Name** field is returned from each matching row.

The **Name** field contains the value from the dictionary. There can be more than one value for the **Name** field, if the URI matches against multiple fields.

3. Analyze Link data using the dictionary fields:

The **Name** field can now be used like any other field in Link. For example, the following query filters by valid values for **Name** and analyzes the results against the HTTP Status in the response:

```
'Log Source' = 'FMW WLS Server Access Logs'  
| link URI, Status  
| lookup table = 'Page Access Types' select Name using URI  
| where Name != null  
| classify Status, Name as 'Page Analysis'
```

This query produces the analytical chart showing the distribution of HTTP Status for various pages. The resulting bubble chart has the pages like "*Login Page, Home Page*", "*Home Page, Script Access*", *Home Page, Login Page*, and *Script Access* plotted along Y-axis, and the HTTP status along X-axis.

Semantic Clustering Using Natural Language Processing

Cluster Visualization allows you to cluster text messages in log records. Cluster works by grouping messages that have similar number of words in a sentence, and identifying the words that change within those sentences. Cluster does not consider the literal meaning of the words during the grouping.

The new NLP (Natural Language Processing) command supports semantic clustering. Semantic Clustering is done by extracting the relevant keywords from a message and clustering based on these keywords. Two sets of messages that have similar words are grouped together. Each such group is given a deterministic Cluster ID.

The following example shows the usage of NLP clustering and keywords on *Linux Syslog Logs*:

```
'Log Source' = 'Linux Syslogs Logs'  
| link Time, Entity, cluster()  
| nlp cluster('Cluster Sample') as 'Cluster ID',  
           keywords('Cluster Sample') as Keywords  
| classify 'Start Time', Keywords, Count, Entity as 'Cluster Keywords'
```



For more example use cases of semantic clustering, see [Examples of Semantic Clustering Using Natural Language Processing](#).

nlp Command

The `nlp` command supports two functions. `cluster()` can be used to cluster the specified field, and `keywords()` can be used to extract keywords from the specified field.

`nlp` command can be used only after the `link` command. See NLP Command in *Using Oracle Log Analytics Search*.

- **nlp cluster():**

`cluster()` takes the name of a field generated in Link, and returns a Cluster ID for each clustered value. The returned Cluster ID is a number, represented as a string. The Cluster ID can be used in queries to filter the clusters.

For example:

```
nlp cluster('Description') as 'Description ID' - This would extract relevant
keywords from the Description field. The Description ID field would contain a
unique ID for each generated cluster.
```

- **nlp keywords():**

Extracts keywords from the specified field values. The keywords are extracted based on a dictionary. The dictionary name can be supplied using the `table` option. If no dictionary is provided, the out-of-the-box default dictionary *NLP General Dictionary* is used.

For example:

```
nlp keywords('Description') as Summary - This would extract relevant
keywords from the Description field. The keywords are accessible using the
Summary field.
```

```
nlp table='My Issues' cluster('Description') as 'Description ID' -
Instead of the default dictionary, use the custom dictionary My Issues.
```

NLP Dictionary

Semantic Clustering works by splitting a message into words, extracting the relevant words and then grouping the messages that have similar words. The quality of clustering thus depends on the relevance of the keywords extracted.

- A dictionary is used to decide what words in a message should be extracted.
- The order of items in the dictionary is important. An item in the first row has higher ranking than the item in the second row.
- A dictionary is created as a .csv file, and imported using the Lookup user interface with **Dictionary Type** option.
- It is not necessary to create a dictionary, unless you want to change the ranking of words. The default out-of-the-box NLP General Dictionary is used if no dictionary is specified. It contains pre-trained *English* words.

See [Create a Dictionary Lookup](#).

Following is an example dictionary *iSCSI Errors*:

Operator	Condition	Value
CONTAINS IGNORE CASE	error	noun
CONTAINS IGNORE CASE	reported	verb
CONTAINS IGNORE CASE	iSCSI	noun
CONTAINS IGNORE CASE	connection	noun
CONTAINS IGNORE CASE	closed	verb

The first field is reserved for future use. Second field is a word. The third word specifies the type for that word. The type can be any string and can be referred to from the query using the `category` parameter.

In the above example, the word *error* has higher ranking than the words *reported* or *iSCSI*. Similarly, *connection* has higher ranking than *closed*.

Using a Dictionary

Suppose that the following text is seen in the `Message` field:

```
Kernel reported iSCSI connection 1:0 error (1020 - ISCSI_ERR_TCP_CONN_CLOSE: TCP
connection closed) state (2)
Please verify the storage and network connection for additional faults
```

The above message is parsed and split into words. Non-alphabets are removed. Following are some of the unique words generated from the split:

```
Kernel
reported
iSCSI
connection
error
ERR
TCP
CONN
CLOSE
closed
```

```
state
...
...
```

There are a total of 24 words in the message. By default, semantic clustering would attempt to extract 20 words and use these words to perform clustering. In a case like the above, the system needs to know which words are important. This is achieved by using the dictionary.

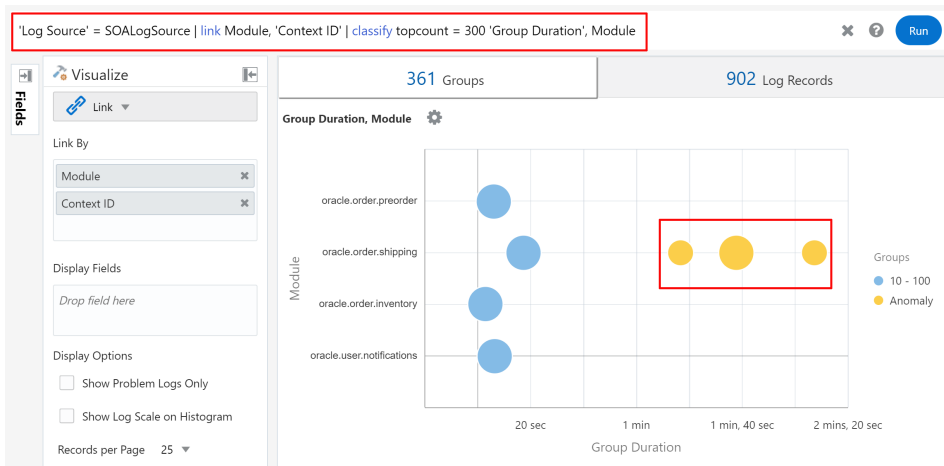
The dictionary is an ordered list. If *iSCSI Errors* is used, then NLP would not extract *ERR*, *TCP*, or *CONN* because these words are not included in the dictionary. Similarly, the words *error*, *reported*, *iSCSI*, *connection*, and *closed* are given higher priority due to their ranking in the dictionary.

Generate Link Alerts

After you have viewed your log records in the link visualization and determined the boundaries in which the anomalies typically appear, you can create alert rules to get notifications when anomalies are detected.

You can save a maximum of 50 scheduled alerts.

Consider the following order flow application example where the anomalies are detected for transactions that take more than 1 minute to complete.



1. To create an alert rule that will notify you upon detecting anomalies, you must first define the condition in the query. Edit the highlighted query, and add the `where` command to define the `Group Duration` in which the anomalies are found, that is, when it's more than or equal to 60,000 milliseconds. For example:

```
'Log Source' = SOALogSource | link Module, 'Context ID' | classify
topcount = 300 'Group Duration', Module | where 'Group Duration' >=
60000
```

2. Click the down arrow next to **Save**, and select **Save As**.
The dialog box to create the alert rule opens.
3. Specify the name for the alert under **Search Name**.

4. Check the **Create alert rule** check box.

The field **Rule Name** is automatically populated with the alert name that you specified earlier. The **Enable Rule** check box is enabled by default.

5. Under condition type, select **Fixed Threshold**.
6. Under **Results**, specify the warning and critical thresholds for the notification actions. For example, if you want a warning notification if more than one anomaly are detected, and critical notification if more than five anomalies are detected, then select the operator for `greater than or equal to`, warning threshold **1**, and critical threshold **5**.
7. Schedule the interval at which the test must be run to detect anomalies. For example, **Every Day**. This will depend on the frequency of collecting your logs, and the number of log records that you expect to be analyzed on a regular basis.

The time period of the logs analyzed for the saved search alert is the same as the run period. For example, if you select 15-minute interval, then the logs are checked for the last 15 minutes at that specific time.

You can select **Every Hour**, **Every Day**, **Every Week** or a **Custom** setting for any value between 15 minutes to 21 days as the Schedule Interval. Your saved search runs automatically based on the interval that you specify.

If you select **Every Hour**, then you can optionally specify to exclude **Weekend** or **Non-business hours** from the schedule.

If you select **Every Day**, then you can optionally specify to **Exclude Weekend** from the schedule.

8. If you want to customize your alert message, then under **Customize Message Format**, select **Use custom message**. You can customize any or all of the messages available under this section. For details, see Step 8 in [Create An Alert Rule](#).
9. In **Notifications**, specify the recipients of the alert notifications and in **Remediation Action**, select the action that must be performed automatically in response to an alert. For details, see Step 9 and Step 10 in [Create An Alert Rule](#).


Click **Save**.

The alert is now created. You can visit the Alert Rules page to view the alert that you just created, and edit it, if required. See [View and Edit Alert Rules](#).

To view the alerts generated, see [View the Entity Details for an Alert](#).

Use the Getting Started Panel

If you're new to using **Link**, then you can familiarize with the following features by using the *Getting Started Panel*:

1. On the results table header, click the **Open the Getting Started panel** () icon to open the **Getting Started Panel**.
2. On the **Getting Started** tab, click the **Show Tips** link to view some useful tips to explore options on the visualization of the Link feature.
Click **Hide Tips**.
3. Click on the **Sample Link Commands** tab. View and edit some of the sample link commands.

You can select to **Run** a link command that's listed under **Available Sample Link Commands** or **View** the link commands listed under **All Sample Link Commands**.

4. Click on the **Link Builder** tab, and run the wizard to select the **Log Source**, select up to four fields in **Link By**, select up to two fields in **Analyze Fields**, and click **Run Link** to build custom queries. You can select multiple fields at once before running the query, thus saving time from having the drag and drop operation to complete the background query for every field.

Click **Clear** to clear the selection.

For example, if you select *EBS Concurrent Request Logs - Enhanced* log source from the available sample link command and run it, you can obtain the following information:

- Requests that have already completed execution within the selected time window
- Currently running requests that show anomalous run times
- Ability to create an Alert to identify specific requests that took anomalous run time to complete, or still running but with anomalous run time



Analyze Chart Options

The following chart options are available to analyze the groups that're displayed by the Link query:

Analyze Chart Option	Utility
Chart Type	<p>Select from the bubble, scatter, tree map, and sunburst type of charts to view the groups. By default, a bubble chart is displayed.</p> <ul style="list-style-type: none"> • Bubble Chart: To analyze the data from three fields, and each field can have multiple values. The position of the bubble is determined by the values of the first and second fields that're plotted on the x and y axes, and the size of the bubble is determined by the third field. • Scatter Chart : To analyze the data from two numeric fields, to see how much one parameter is affecting the other. • Tree Map: To analyze the data from multiple fields that're both hierarchical and fractional, with the help of interactive nested rectangles. • Sunburst Chart: To analyze hierarchical data from multiple fields. The hierarchy is represented in the form of concentric rings, with the innermost ring representing the top of the hierarchy.
Height	Increase or decrease the height of the chart to suit your screen size.
Swap X Y axis	You can swap the values plotted along the x and y axes for better visualization.
Show Anomalies	View the anomalies among the groups displayed on the chart.

Analyze Chart Option	Utility
Highlight Anomaly Baselines	If you've selected to view the anomalies, then you can highlight the baselines for those anomalies.
Show Group Count Legend	Toggle the display of the Group Count legend.
Zoom and Scroll	Select Marquee zoom or Marquee select to dynamically view the data on the chart or to scroll and select multiple groups.

When displaying *Problem Priority*, Analyze charts display colors that match the severity of Problem Priority.

You can create multiple Analyze charts. Click **Analyze**  > **Create Chart** option. Configure each chart by clicking **Chart Options**  > **Chart Settings** > **Edit Chart** for that chart.

Additional Information in Analyze Chart

Hover your cursor over a filter legend in the Link Analyze Chart to view additional information about those values. For each legend displayed in the chart, the following information is additionally available:

- **Clusters:** Number of bubbles in the chart for this value
- **Groups:** Total number and percentage of groups across all the clusters
- **Average Cluster Range:** Each bubble or cluster represents a range of values. An average is computed for each bubble. This value shows the minimum and maximum averages across all the bubbles, in case of numeric values.
- **Minimum Value:** Lowest absolute value across all the bubbles for this legend range.
- **Maximum Value:** Largest absolute value across all the bubbles for this legend range.

Histogram Chart Options

Histogram shows the dispersion of log records over the time period and can be used to drill down into a specific set of log records.

You can generate charts for the log records, groups and numeric display fields. Select a row to view the range highlighted in the histogram.

The following chart options are to view the group data on the histogram:

Histogram Chart Option	Utility
Chart Type	<p>Select from the following types of visualization to view the group data:</p> <ul style="list-style-type: none"> • Bar: The log records are displayed as segmented columns against the time period. This is the default display chart. • Marker Only : The size of the log records against the specific time is represented by a marker. • Line Without Marker: The size of the log records against the specific time is plotted with the line tracing the number that represents the size. • Line With Marker: The size of the log records against the specific time is plotted with the line tracing the marker that represents the size. • Line With Area: This is similar to a line chart, but the area between the line and the axis is covered with color. The colored area represents the volume of data.
Show Combined Chart	This option combines all the individual charts into a single chart.

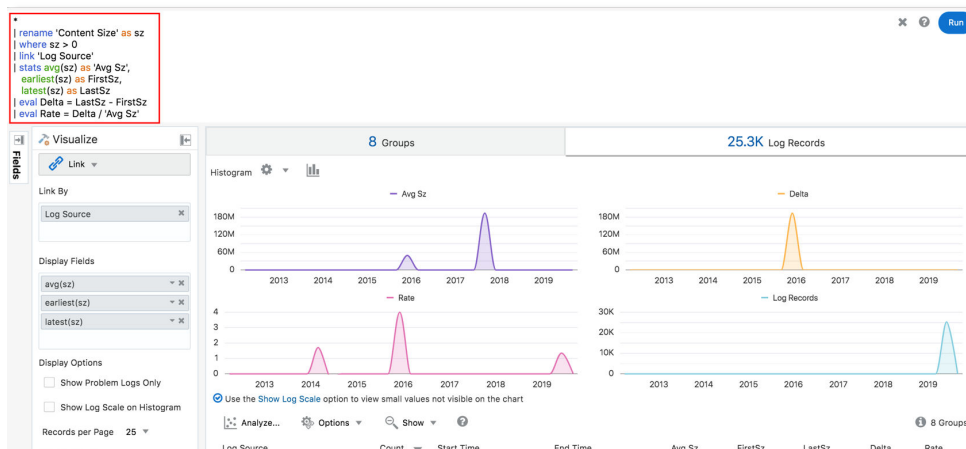
 **Note:**

- You can modify the **Height** and **Width** of the charts to optimize the visualization and view multiple charts on one line.
- When viewing multiple charts, you can deselect the **Show Correlated Tooltips** check box to show only one tooltip at a time.
- When using the log scale, the **Bar** or **Line With Marker** type of chart is recommended.

Example: For generating a chart for the numeric `eval` command, let's consider the example query:

```
*
| rename 'Content Size' as sz
| where sz > 0
| link 'Log Source'
| stats avg(sz) as 'Avg Sz', earliest(sz) as FirstSz, latest(sz) as
LastSz
| eval Delta = LastSz - FirstSz
| eval Rate = Delta / 'Avg Sz'
```

Here, the log source is the field considered for `Link By`. The chart is generated for `Delta`, `Rate`, and `Avg Sz` after the computations performed as specified in the `eval` command. The resulting *Line With Area* charts for the above fields are displayed as below:



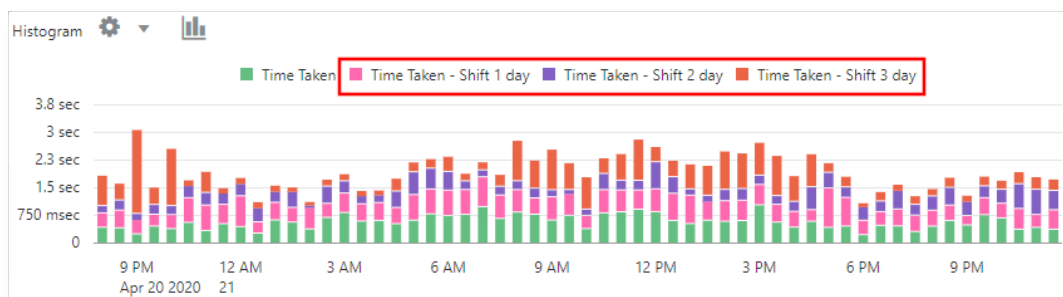
Compare Link Metrics Across Time

Use the `compare` command to compare metrics generated in link analysis to the previous time windows.

Following example query compares the *Average Duration* for each UI page across previous three days:

```
'Log Source' = 'Application Server Access Logs'
| eval 'Duration (sec)' = unit(Duration, second)
| link Time, Page
| stats avg('Duration (sec)') as 'Time Taken'
| compare fields = 'Time Taken' timeshift = -1day count = 3
```

The resulting histogram chart that indicates the comparison:



The resulting groups table that indicates the comparison:

Page	Count	Start Time	End Time	Time Taken	Time Taken - Shift 1 day	Time Taken - Shift 2 day	Time Taken - Shift 3 day
BI Security	29,188	Apr 21, 2020, 10:00:00 AM	Apr 21, 2020, 10:29:59 AM	42.8 msec	61.4 msec	60.5 msec	203.3 msec
Home Page	26,828	Apr 21, 2020, 7:00:00 AM	Apr 21, 2020, 7:29:59 AM	908.4 msec	990.7 msec	664.3 msec	525.5 msec
Home Page	24,579	Apr 21, 2020, 11:00:00 AM	Apr 21, 2020, 11:29:59 AM	1 sec	759.3 msec	341.7 msec	1.5 sec

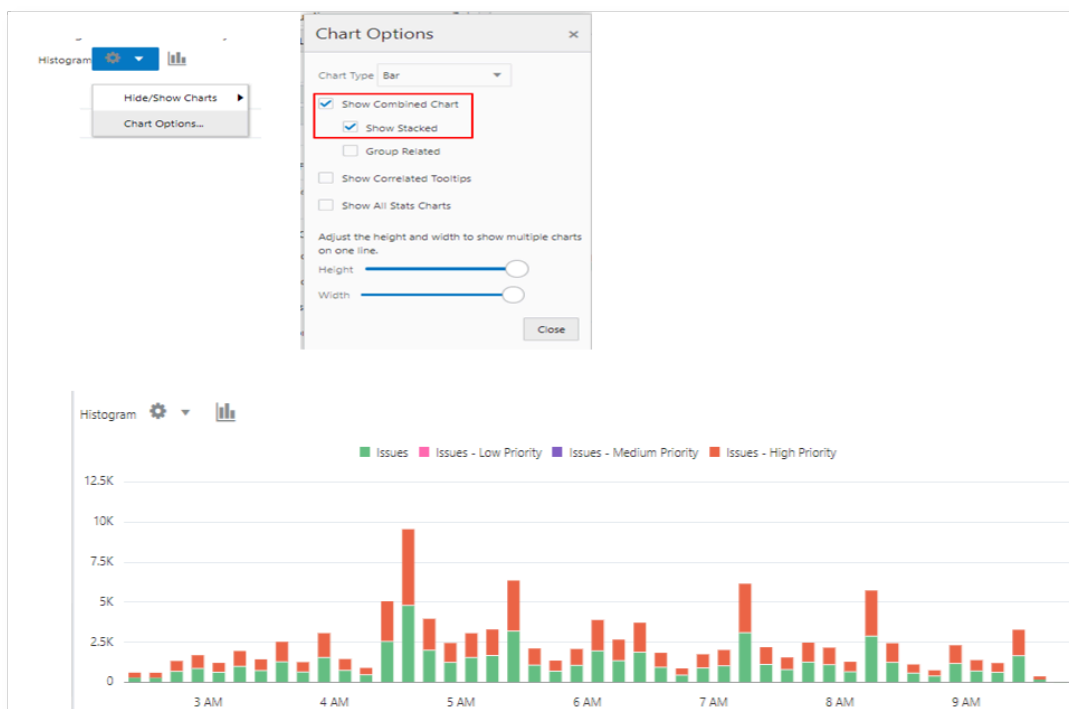
See Compare Command in *Using Log Analytics Search*.

Combine and Stack Histogram Charts

You can combine and stack charts using the **Show Combined** and **Show Stacked** options in link.

For example, the following query shows the trend of logs with various values for the *Problem Priority* field, in a stacked chart:

```
*
| link Time, Entity
| addfields
  [ 'Problem Priority' != null | stats count as Issues ],
  [ 'Problem Priority' = Low | stats count as 'Issues - Low
Priority' ],
  [ 'Problem Priority' = Medium | stats count as 'Issues - Medium
Priority' ],
  [ 'Problem Priority' = High | stats count as 'Issues - High
Priority' ]
| fields -Issues, -'Issues - Low Priority', -'Issues - Medium
Priority', -'Issues - High Priority'
```



Groups Table

The groups table displays the result of the analysis by listing the groups and the corresponding values for the following default fields:

Column	Details
Field (s)	The field that's used to analyze the group
Count	The number of log records in the group
Start Time	The start of the time period for which the logs are considered for the analysis
End Time	The end of the time period for which the logs are considered for the analysis
Group Duration	The duration of the log event for the group

When displaying *Problem Priority*, groups table displays colors that match the severity of Problem Priority.

Add URLs to Link Table

You can create links using the `url` function of the `eval` command. In the following query, the values for Search 1, Search 2, and Search 3 are assigned URLs:

```
'Log Source' = 'Database Alert Logs'
| link cluster()
| where 'Potential Issue' = '1'
| nlp keywords('Cluster Sample') as 'Database Error'
```

```
| eval 'Search 1' = url('https://www.google.com/search?q=' ||
'Database Error')
| eval 'Search 2' = url('https://www.google.com/search?q=' ||
'Database Error', Errors)
| eval 'Search 3' = url('google', 'Database Error')
```


Cluster Sample	Count	Start Time	End Time	Potential Issue	Database Error	Search 1	Search 2	Search 3
TT03: Standby redo logfile selected for thread 3 sequence 156823 for destination LOG_ARCHIVE_DEST_4 This error struct:	1,676,198	Jan 30, 2015, 2:47:58 PM	Dec 16, 2016, 8:37:07 AM	1	archive, destination, error, log, redo, selected, sequence, standby, struct, thread	https://www.google.com/search?q=archive, destination, error, log, redo, selected, sequence, standby, struct, thread	Errors	archive, destination, error, log, redo, selected, sequence, standby, struct, thread
Errors in file /bglog/.../AC/diag/rdbms/.../AC1_j002_28629.trc ORA-12012: error on auto execute of job '..._INT_B_36193956' ORA-20001: ... is locked. try again later. ORA-06512: at ... RROR', line 45 ORA-06512: at line 953 ORA-06512: at '...', line 103 ORA-06512: at line 1	1,571,440	Jan 30, 2015, 1:32:59 PM	Dec 16, 2016, 8:39:50 AM	1	auto, bug, error, errors, execute, file, job, line, try	https://www.google.com/search?q=auto, bug, error, errors, execute, file, job, line, try	Errors	auto, bug, error, errors, execute, file, job, line, try

Features for Bubble Charts in Link Analysis

Use the following features to edit the bubble chart:

Change the Title of the Bubble Chart

To improve the readability of the chart and for friendly analysis, you can change the title of the bubble chart by using the option in the Analyze dialog box.

To modify the title of the bubble chart, click Analyze  icon > In the **Analyze** dialog box, update the value of the field **Chart Title** > Click **OK**.

As a result, the title of the chart is now changed to the value that you provided.

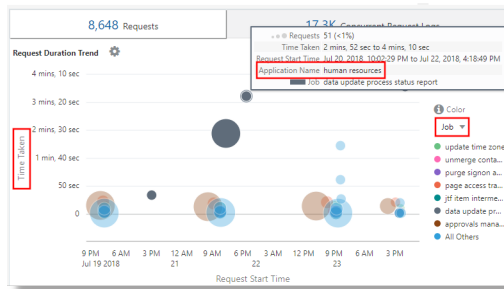
Control the Color of the Bubbles in the Chart

To plot along the X-axis, you can select a numeric, string, or time field. Only a numeric or string field can be used for the Y-axis.

- Any fields can be used to control the color of the bubbles. There are no restrictions about the types of the fields.
- Numeric fields can be used for controlling the size of the bubbles. The value of the fields control the size of the bubble. The larger the values, the larger the bubbles.

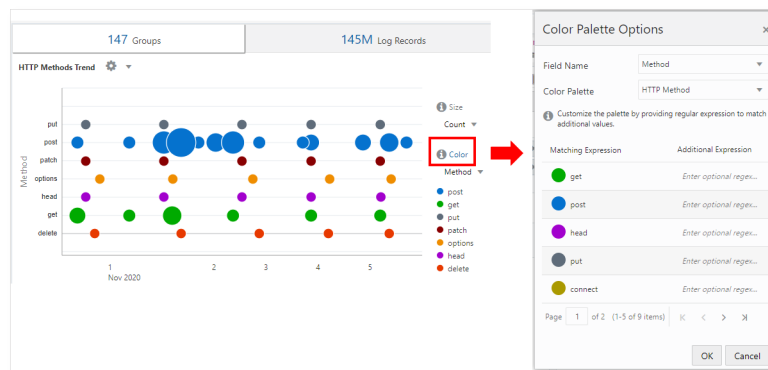
For steps to select the fields for controlling the color of the bubbles in the chart, see [Add More Fields for Analysis Using Size and Color](#).

The following chart shows the *Time Taken* for Requests, which is plotted along Y-axis, and also the *Application* and *Job* that are involved in the analysis:



By default, the Link Analyze chart automatically selects a color palette based on the values in the chart. To select a different palette or to add additional field values, click the **Color** link. In the following example, the field *Method* has HTTP Method color palette applied for different values:

```
'Log Source' = 'FMW WebLogic Server Access Logs'
| link Time, Method
| classify Time, Method, Count as 'HTTP Methods Trend'
```



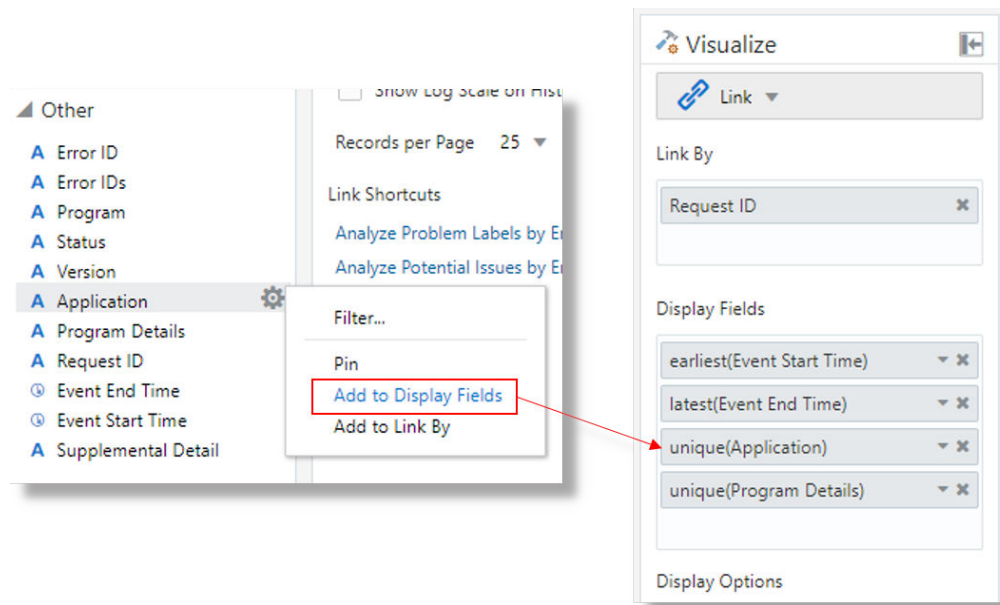
Features for Fields in Link Analysis


Use the following groups to work with the fields in the Link visualization:

Add More than Two Fields

Add more than two fields to the analysis. Each field that is added for analysis appears as a column in the **Groups Table**.

Consider the following example:



Select the field from the **Fields** panel > click the Options  icon > use the **Add to Display Fields** option to extract their values.

As a result, the Groups table has the columns for the fields Event Start Time, Event End Time, unique(Application), and unique(Program Details).

Rename the Fields by Editing the Query

By default, the fields that you add to the **Display Fields** panel will be displayed in the column names of the **Groups Table** with the name of the function that was used to create the field. Edit the query to give names to the fields.

Consider the following example for the query that is currently used to run link feature:

```
'Log Source' = 'EBS Concurrent Request Logs - Enhanced'
| link 'Request ID'
| stats earliest('Event Start Time') as 'Request Start Time',
latest('Event End Time') as 'Request End Time',
unique(Application),
unique('Program Details')
| eval 'Time Taken' = 'Request End Time' - 'Request Start Time'
| classify topcount = 300 'Request Start Time', 'Time Taken' as
'Request Analysis'
```

To change the names of the fields unique(Application) to **Application Name** and unique('Program Details') to **Job**, modify the query:

```
'Log Source' = 'EBS Concurrent Request Logs - Enhanced'
| link 'Request ID'
| stats earliest('Event Start Time') as 'Request Start Time',
latest('Event End Time') as 'Request End Time',
unique(Application) as 'Application Name',
unique('Program Details') as Job
```


```
| eval 'Time Taken' = 'Request End Time' - 'Request Start Time'
| classify topcount = 300 'Request Start Time', 'Time Taken' as 'Request Analysis'
```

After renaming the fields, you can refer to the fields using the new names. The column names in the Groups Table will have the new names of the fields.

Add More Fields for Analysis Using Size and Color

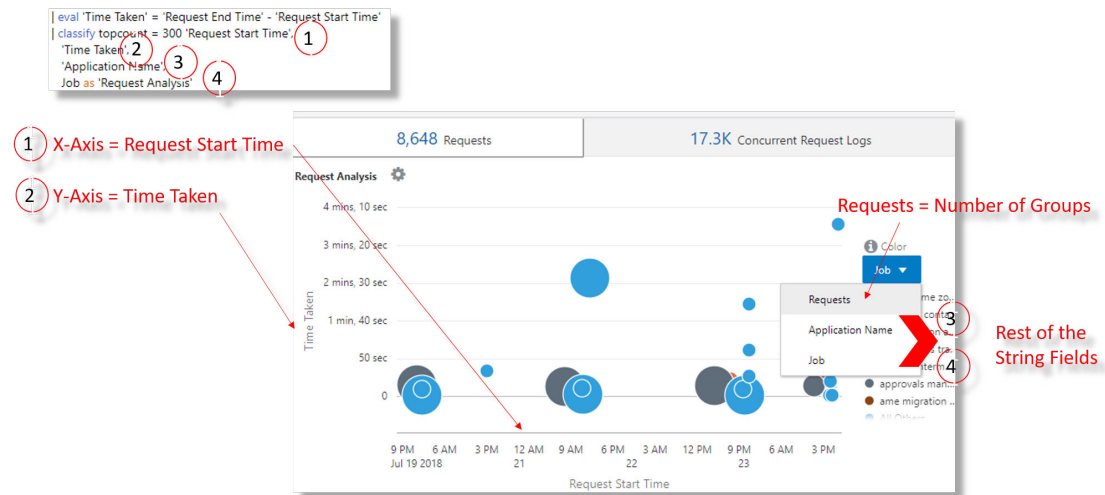
In the bubble chart, two fields are used to plot along the x-axis and y-axis. The remaining fields can be used to control the size and color of the bubbles in the chart.

Two fields are used in the chart to plot along X and Y axes. To add more fields for analysis in the bubble chart,

1. Click Analyze  > Click **Create Chart**. The **Analyze** dialog box is displayed.
2. Select the field to plot along the **X-axis**. This must be a numerical field.
3. Select the field to plot along the **Y-axis**. This must be a numerical field.
4. In the **Size / Color** panel, select the fields that must be used for defining the size and colors of the bubbles in the chart. Any fields can be used for controlling the color, but numeric fields must be used to control the size of the bubbles.
5. Click **OK**.

Additionally, *Group Count* is available as a field to control the size and color.

The `classify` command is now run with multiple fields, in the order specified in the *Analyze* selection. The following bubble chart shows multiple fields:



In the above example,

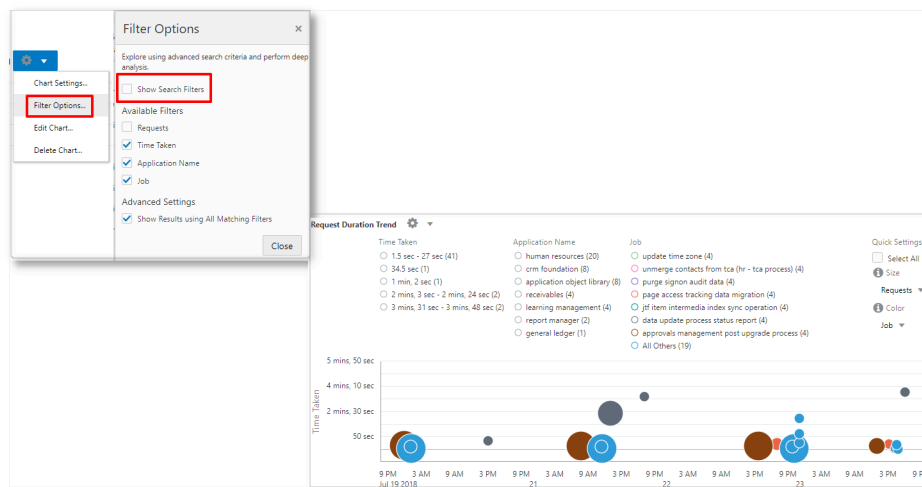
- The field `Request Start Time` is plotted along X-axis
- The field `Time Taken` is plotted along Y-axis
- The string fields `Application Name` and `Job` are used for controlling the size and color of the bubbles in the chart

Furthermore, the *Groups* alias is changed to *Requests*, and *Log Records* alias is changed to *Concurrent Request Logs*.

Instant Analysis of Multiple Fields Using the Link Analyzer Chart

Slice and dice data using multiple filters in the **Analyzer Chart**.

Use **Filter Options > Show Search Filters** to enable the filters:



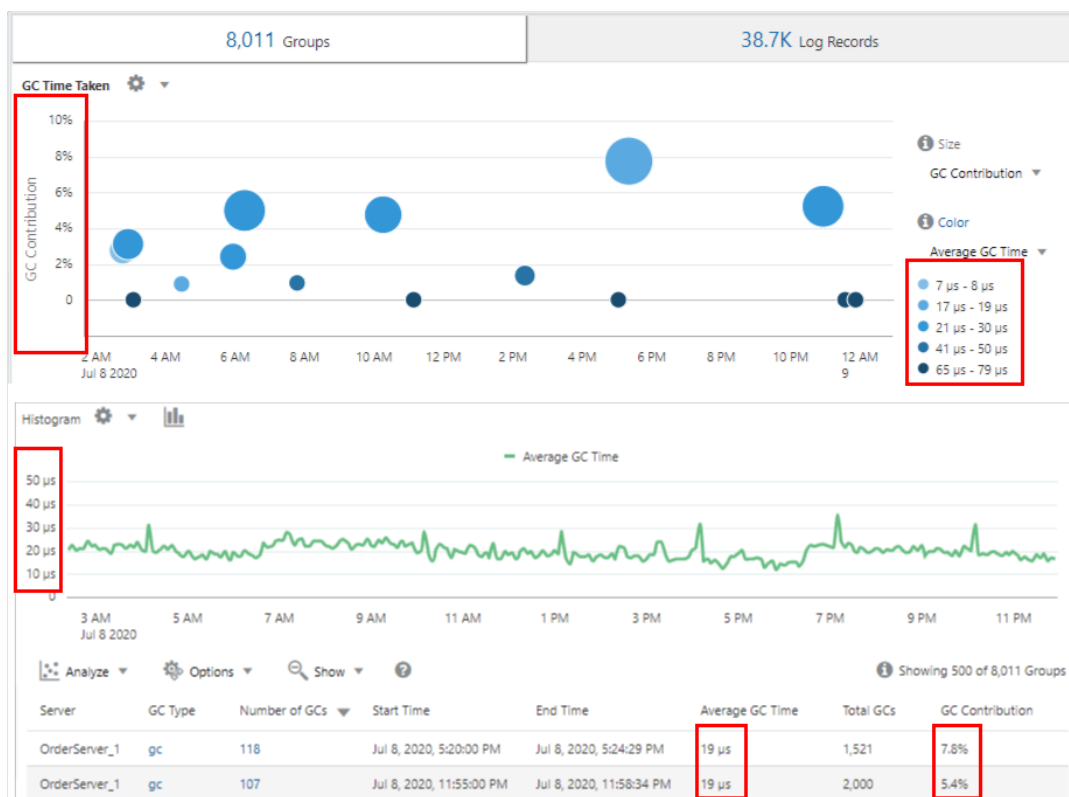
Mark a Field Type as Percentage or Microsecond

In addition to hour, minute, second and millisecond, you can now mark a field as containing value in microseconds or percentage value.

Consider the following example which illustrates use of microsecond and percentage field type:

```
| *
| eval GC = unit('GC Time', micro)
| link span = 5minute Time, Entity, 'GC Type'
| rename Count as 'Number of GCs'
| stats avg(GC) as 'Average GC Time'
| eventstats sum('Number of GCs') as 'Total GCs' by Server
| eval 'GC Contribution' = unit(100 / ('Total GCs' / 'Number of GCs'),
pct)
| classify 'Start Time', 'GC Contribution', 'Average GC Time' as 'GC
Time Taken'
```

In the following charts, the value of *GC Time* and *GC Contribution* are shown in their respective field types:



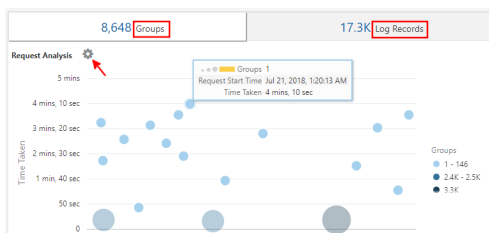
Features for Groups in Link Analysis

Use the following features to modify the groups:

Change the Group Alias

Each row in the link table corresponds to a Group. In the following example, the link command is run using the *Request ID* field. Therefore, each row of the table represents a request. You can change the alias for *Groups* and *Log Records* tabs.

The following example shows the bubble chart in the **Groups** tab. The adjacent **Log Records** tab can also be seen in the image:



Click **Search and Table Options**  icon > Click **Display Options** > Under **Alias Options**, modify the **Groups Alias** and **Log Records Alias** values.

The **Group Alias** is used when there is only one item in the Groups table.

Join Multiple Groups Using the Map Command

Use `map` command to join multiple sub-groups from the existing linked Groups. This is useful to assign a *Session ID* for related events, or to correlate events across different servers or log sources.

For example, the below query joins *Out of Memory* events with other events that are within 30 minutes, and colors these groups to highlight a context for the *Out of Memory* outage:

```
* | link Server, Label
  | createView [ * | where Label = 'Out of Memory'
                | rename Entity as 'OOM Server', 'Start Time' as
                'OOM Begin Time' ] as 'Out of Memory Events'
  | sort Entity, 'Start Time'
  | map [ * | where Label != 'Out of Memory' and Server = 'OOM Server'
        and 'Start Time' >= dateAdd('OOM Begin Time', minute,-30) and 'Start
        Time' <= 'OOM Begin Time'
        | eval Context = Yes ] using 'Out of Memory Events'
  | highlightgroups color = yellow [ * | where Context = Yes ] as '30
  Minutes before Out of Memory'
  | highlightgroups priority = high [ * | where Label = 'Out of
  Memory' ] as 'Server Out of Memory'
```

Server	Label	Count	Start Time	End Time	Context
orderapp_server1	Exception	288	Jul 8, 2020, 2:25:57 AM	Jul 8, 2020, 11:41:29 PM	
orderapp_server1	IO Error	41	Jul 8, 2020, 2:25:57 AM	Jul 8, 2020, 9:12:22 PM	
orderapp_server1	Socket Timeout	41	Jul 8, 2020, 2:25:57 AM	Jul 8, 2020, 9:12:22 PM	
orderapp_server1	Application Error	13	Jul 8, 2020, 3:10:42 AM	Jul 8, 2020, 11:09:30 PM	
orderapp_server1	Stack Trace	17	Jul 8, 2020, 11:54:32 AM	Jul 8, 2020, 12:17:41 PM	Yes
orderapp_server1	Connection Error	8	Jul 8, 2020, 12:08:46 PM	Jul 8, 2020, 12:11:37 PM	Yes
orderapp_server1	Availability Error	1	Jul 8, 2020, 12:12:09 PM	Jul 8, 2020, 12:12:09 PM	Yes
orderapp_server1	Too Many Stack Traces	1	Jul 8, 2020, 12:15:43 PM	Jul 8, 2020, 12:15:43 PM	Yes
orderapp_server1	Server Health Failed	1	Jul 8, 2020, 12:15:43 PM	Jul 8, 2020, 12:15:43 PM	Yes
orderapp_server1	Memory Error	1	Jul 8, 2020, 12:20:03 PM	Jul 8, 2020, 12:20:03 PM	Yes
orderapp_server1	Out of Memory	1	Jul 8, 2020, 12:20:03 PM	Jul 8, 2020, 12:20:03 PM	Yes

See Map Command in *Using Oracle Log Analytics Search*.

Create Sub-Groups Using the Createview Command

Use `createview` command to create sub-groups from the existing linked groups. This can be used in conjunction with the `map` command to join groups.

For example, you can group all the *Out of Memory* errors using the following command:

```
* | link Entity, Label
  | createView [ * | where Label = 'Out of Memory' ] as 'Out of
  Memory Events'
```

See Createview Command in *Using Oracle Log Analytics Search*.

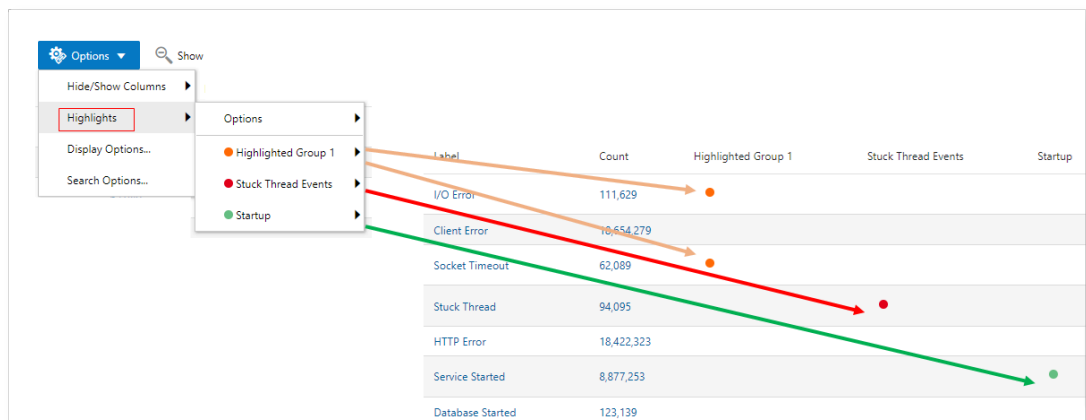
Search and Highlight Link Groups

Use `highlightgroups` command to search one or more columns in the Link results and highlight specific groups. You can optionally assign a priority to the highlighted regions. The priority would be used to color the regions. You can also explicitly specify a color.

Optionally, you can specify an alias for the highlight. This alias is displayed on mouse over on the highlighted region. The alias can also be used to turn on or off the highlight using the **Hide/Show Highlights** option under the **Options** menu.

For example:

```
* | link Label
| highlightgroups priority = medium [ * | where Label in ('I/O Error',
'Socket Timeout') ]
| highlightgroups priority = high [ * | where Label = 'Stuck Thread' ] as
'Stuck Thread Events'
| highlightgroups color = #68C182 [ * | where Label = 'Service Started' ]
as Startup
```



See `Highlightgroups` Command in *Using Oracle Log Analytics Search*.

A maximum of 500 rows are displayed in the Link table. You can navigate to any of these 500 rows using the following hot keys:

Key	Navigation
F	Go to the first record
L	Go to the last record
N	Next record from the current highlighted row
P	Previous record from the current highlighted row

You can make the navigation keys active by selecting a highlight. Go to **Options** > click **Highlights** > click **First or Last Occurrence**.

In the following example, the high GC event is highlighted in red. The other events which happened within 15 minutes are displayed for context. You can navigate to the next event to identify similarities and differences between the events.

```
'Log Source' = 'Application Server Logs'
| eval 'GC Time (sec)' = unit('GC Time', second)
| link includenulls = true span = 1minute Time, Server, Label
  | stats avg('GC Time (sec)') as 'Average GC'
  | eventstats median('Average GC') as 'Median GC' by Server
  | eval 'GC Status' = if('Average GC' > 1 and 'Average GC' >= 'Median
GC' * 2, Bad, Ok)
  | sort Server, 'Start Time'
  | createview [ * | where 'GC Status' = Bad | rename Entity as E,
'Start Time' as S ] as 'High GC Records'
  | map [ * | where Entity = E and 'Start Time' >= dateAdd(S,
minute, -15) and 'Start Time' <= S | eval Context30mins = 1 ] using
'High GC Records'
  | highlightgroups color = #A8FF33 [ * | where 'GC Status' != Bad ]
as 'GC - Ok'
  | highlightgroups color = red [ * | where 'GC Status' = Bad ]
as 'GC - Bad'
  | fields -Context30mins
```

GC Status	Server	Count	Start Time	End Time	Label	Average GC	Median GC
Ok	OrderServer_1	1	May 5, 2020 7:30:55 AM	May 5, 2020 7:30:55 AM	Exception		200 msec
Ok	OrderServer_1	1	May 5, 2020 7:30:55 AM	May 5, 2020 7:30:55 AM	Null Pointer Exception		200 msec
Ok	OrderServer_1	3	May 5, 2020 7:32:18 AM	May 5, 2020 7:32:36 AM	Exception		200 msec
Ok	OrderServer_1	3	May 5, 2020 7:32:18 AM	May 5, 2020 7:32:36 AM	Null Pointer Exception		200 msec
Bad	OrderServer_1	9	May 5, 2020 7:32:18 AM	May 5, 2020 7:32:48 AM		1.2 sec	200 msec
Ok	OrderServer_1	2	May 5, 2020 7:33:09 AM	May 5, 2020 7:33:17 AM		210 msec	200 msec
Ok	OrderServer_1	7	May 5, 2020 7:36:11 AM	May 5, 2020 7:36:35 AM			200 msec
Ok	OrderServer_1	3	May 5, 2020 7:36:11 AM	May 5, 2020 7:36:35 AM	Exception		200 msec
Ok	OrderServer_1	1	May 5, 2020 7:40:42 AM	May 5, 2020 7:40:42 AM		210 msec	200 msec

Link by Cluster

You can combine the link and cluster capabilities to classify clusters for a specific field. You can identify the entities or entity types that have the most potential issues and see any patterns or anomalies across those entities.

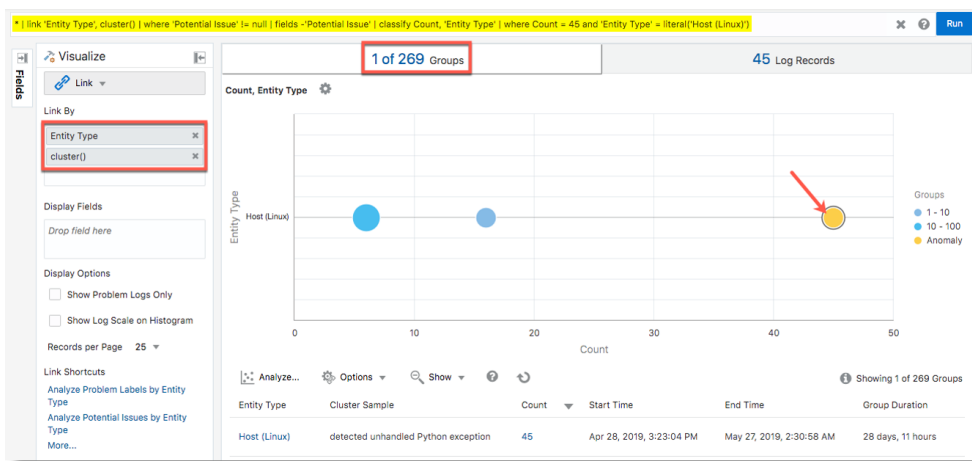
Using clusters, you can analyze a large set of log records and identify potential issues. With the **Link by Cluster** capability, you can group the log records by clusters and identify potential issues based on your selection of the field for analysis. For example,

if you want to group your clusters based on entity, entity type, or log source, then you can use link by cluster.

In the following example, the log records of `Host (Linux)` entity type are analyzed with link and cluster capabilities by including `* | link 'Entity Type', cluster()` in the query. The complete query used for analysis is:

```
* | link 'Entity Type', cluster() | where 'Potential Issue' != null | fields
-'Potential Issue' | where Count = 45 and 'Entity Type' =
literal("Host(Linux)")
```

First, the cluster command is run on the search string, in this case `*`, which produces a field called `Cluster Sample`. This field is linked with entity type to group all the clusters by entity type. The `where` clause specifies to look only for `Potential Issues`. So now, we have all the potential issues grouped by entity type. As you can see in the bubble chart, there are about 45 potential issues of the `Host (Linux)` entity type.



The groups table displays the details of the cluster sample corresponding to the anomaly group. Note the log content of the cluster sample available in the table which is possibly the cause for the potential issue: `detected unhandled Python exception`.

Follow the link [Analyze Potential Issues by Entity Type](#) for the sample command to use in your environment. Click **More** for more sample commands that you can use:

- Potential Issues by Entity
- Potential Issue Outliers by Entity
- Potential Issue Outliers by Entity Type
- Potential Issues by Entity, Severity

In the *Potential Issue Outliers* sample, the query is similar to the example discussed but has another `where` clause added (`where 'Potential Issue' != null and count = 1`) to identify all the errors that have occurred just once over the time period. However, note that the cluster samples still show the variables but the variables shown in link cannot be drilled down into.

For more information about the link visualization and for the steps to access link from the **Visualize** panel, see [Link Visualization](#).

Generate Alerts for Cluster Utilities

Using clusters, you can determine the potential issues and outliers in your log records. You can create alert rules that notify you upon detecting potential issues or outliers with the link by clusters feature.

- **Alert for Potential Issues:** The query to detect potential issues by using the link by clusters command is as follows:

```
* | link cluster() | where 'Potential Issue' != null
```

- **Alert for Outliers:** An outlier in clusters is the one cluster that differs from the other clusters, and occurs in a rare scenario. So, a typical query to detect an outlier by using the link by clusters command is as follows:

```
* | link cluster() | where Count = 1
```

It is recommended that you define a longer interval to detect the outlier while creating the alert rule, for example, `Every Day`.

To the above query, you can add more filters / specifications to identify specific log sources, and to perform additional operations on the log records.

To create the alert rule, click the down arrow next to **Save**, and select **Save As**.

The dialog box to create the alert rule opens. In the alert rule dialog box, under condition type, select **Fixed Threshold**. Also, under **Results**, to get a notification for every detection of a potential issue or outlier, select the operator for `greater than or equal to`, warning threshold **0**, and critical threshold **1**.

You can save a maximum of 50 scheduled alerts.

For the remaining steps to create an alert rule, see [Create An Alert Rule](#).

9

Filter and Search Through the Log Data

Using Oracle Log Analytics, you can search any logs and drill down to specific log entries to resolve problems quickly.

Topics:

- [Typical Workflow for Troubleshooting Problems](#)
- [Search Logs by Entities](#)
- [Use the Filter-Out Option](#)
- [Search Logs Using Keywords and Phrases](#)
- [Filter Logs by Pinned Attributes and Fields](#)

Typical Workflow for Troubleshooting Problems

Here are the common tasks for troubleshooting problems.

Task	Description	More Information
Select the entity or entities that you want to troubleshoot.	View log data pertaining to a specific entity or a set of entities.	Search Logs by Entities
Select a field and its value to filter the log data.	Add or exclude a field in the search query to refine the resultant log data set.	Use the Filter-Out Option
Search logs using keywords and phrases.	Use keywords and phrases in commands to retrieve log data.	Search Logs Using Keywords and Phrases
Filter logs using pinned attributes and field attributes.	Use the pinned and field attributes of Oracle Log Analytics to filter log data.	Filter Logs by Pinned Attributes and Fields
Analyze the log data.	Use the visualization options to organize the log data in a form that helps you to analyze and gain insights.	Visualize Data Using Charts and Controls

After you identify the logs that you want to slice and dice, use the visualization options to identify patterns, trends, potential issues, or to determine the root cause for an issue. Here are some of the examples in which such analysis has been done using visualization charts and controls.

Task	Description	More Information
Detect anomalies.	Cluster log events based on a common signature that helps you identify patterns and outliers.	See RideShare Application analysis in Clusters Visualization and Example Scenario: Detect Anomalies Using Outliers .

Task	Description	More Information
Analyze anomalies.	Group the log records based on fields using the link visualization, identify anomalous groups and view them in cluster visualization.	See order flow application analysis in Link Visualization .
Perform dynamic log analysis.	Use Records with Histogram visualization to explore logs to diagnose and troubleshoot issues at any time.	See RideShare Application analysis in Example Scenario: Perform Dynamic Log Analysis .
Analyze access logs.	Use link visualization to get meaningful insight into the usage statistics, the popularity of the URLs, the most active users, etcetera from the access logs.	See the analysis of Oracle WebLogic Server Access Logs in Perform Advanced Analytics with Link .
Analyze host log trends.	Use Bar Chart , Records with Histogram , and Clusters to analyze the logs from your host to proactively monitor the infrastructure of your applications or IT.	See <i>Analyzing Host Log Trends to Proactively Monitor Infrastructure</i> (Tutorial).

Search Logs by Entities

You can use the **Entity** field in the **Pinned** section of Oracle Log Analytics to filter logs by an entity or multiple entities.

Entities are resources, such as host machines, databases, and Oracle Fusion Middleware components, which can be managed and monitored in Oracle Management Cloud.

To search for logs for the RideShare application entities:

1. From Oracle Log Analytics, in the **Fields** panel, under **Pinned** section, click **Entity**.
2. In the Entity dialog box, select the required entities, and click **Apply**.

Note:

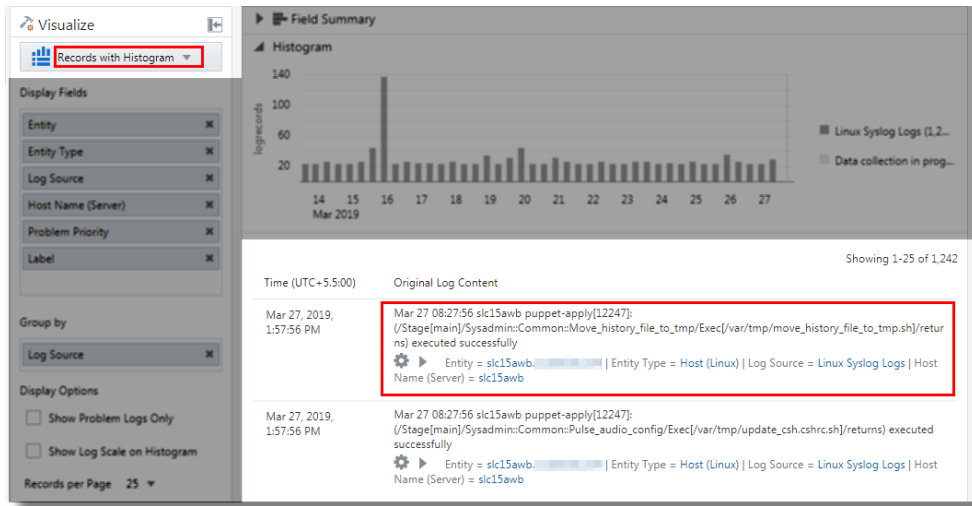
In the Entity dialog box, you can see the occurrence trend for the available entities in the form of sparklines. For the prior example, the sparklines show when the log entries corresponding to the available entities are generated based on the time range selected in the time selector on the top right corner of the dialog box.

Use the Filter-Out Option

You can use the filter-out option in the visualizations that generate a table of records to filter the log data with the fields available in the log records.

In the visualizations that provide table of records, click the field value to view the filter out options. In the following example, the records with histogram chart has a table of

records with the values available for fields like entity, entity type, log source, and host name.



When you click the field value, the following filter-out options are available:

- **Add to Search:** The field that you clicked is added to the search query, and the log data is filtered to include the corresponding field in the search. For example, if you click the entity type value `Host (Linux)` and specify to add it to search, then the previous search query is updated to include `'Entity Type'='Host (Linux)'` in the search string.
- **Exclude from Search:** This excludes the field from the search, and generates a refined result of log records that don't contain the specified field value. For example, if you click the log source value `Linux Syslog Logs` and specify to exclude it from search, then the previous search query is updated to have `'Log Source'!='Linux Syslog Logs'` in the search string. The resultant log data will have only those log records which are not collected from the specified log source.

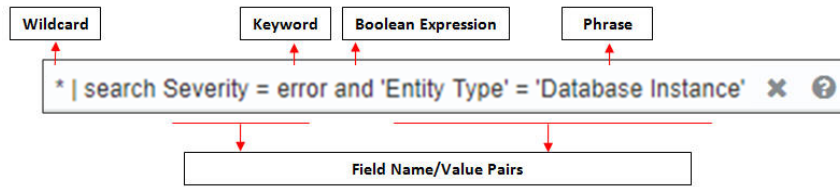
Search Logs Using Keywords and Phrases

You use commands to retrieve log data as well as to perform manipulation on that data. The first (and implicit) command in a query is the search command. A search is a series of commands delimited by a pipe (`|`) character. The first white-spaced string following the pipe character identifies the command to be used. The pipe character indicates that the results from the prior command should be used as input for the next command.

For example, to search for all database error messages, enter the following logical expression in the **Search** field of Oracle Log Analytics:

```
* | SEARCH Severity = 'error' AND 'Entity Type' = 'Database Instance'
```

In the previous example:



The following example returns the same result as the previous example:

```
Severity='error' AND 'Entity Type'='Database Instance'
```

The `SEARCH` keyword is optional, and you can directly enter your search criteria in the **Search** bar to achieve the desired results.

By enclosing the words in quotation marks and including them in the query string as a phrase (`'Database Instance'`, for example), only those logs containing the phrase `'Database Instance'` are returned. In addition, keyword searches where the substring could be interpreted as a separate directive should be specific within quotation marks. For example, to search for the string `and`, you have to enter the string within single quotation marks (`'and'`) to prevent the system from using its Boolean meaning.

List the Recent Searches

Oracle Log Analytics lets you select and run a recently used search. When you click the **Search** field or enter text in the **Search** field, Oracle Log Analytics displays a list of recently used searches. This lets you quickly access recently used search commands. You can select any of the listed commands and click **Run** to execute the selected search command.

Note:

The recently used list is available on a *per session* basis. So if you sign out of Oracle Log Analytics, and then sign in again, the list from the previous session isn't displayed. A new list of recent searches is created for your session.

See About Oracle Log Analytics Search in *Using Oracle Log Analytics Search*.

Use the Autosuggest Feature

When you enter a query in the **Search** field, the autosuggest feature of Oracle Log Analytics automatically suggests terms that you can use in your query. Oracle Log Analytics displays a list of suggestions, based on the text that you've entered in the **Search** field. For example, if you've entered the name of a field or a search action, the autosuggest feature displays the possible values only for that field or the list of available actions.

Filter Logs by Pinned Attributes and Fields

You can also filter data by using the log sources and the fields in the log messages.

- The **Pinned** attributes let you filter log data based on:
 - Log sources, such as database logs, Oracle WebLogic Server logs, and so on.
 - Log entities, which are the actual log file names.
 - Labels, which are tags added to log entries when log entries match specific defined conditions. See [Use Labels in Log Sources](#).
 - Upload names of log data uploaded on demand. See [Upload Logs to Oracle Log Analytics on Demand](#).

By default, the entities and collection details are available in the Pinned bucket of the **Fields** panel for filtering. You can pin additional fields to the Pinned bucket depending on your usage. Once pinned, the fields are moved to the Pinned bucket. You can unpin any field and remove it from the Pinned bucket and move it back to the Interesting or Other bucket.

- Based on your search and queries, Oracle Log Analytics automatically adds fields to the **Interesting** bucket for your quick reference. You can pin a field that's available under Interesting bucket. The pinned field then gets moved to the Pinned bucket.
- You can pin any field in the **Other** bucket and move it to the Pinned bucket. If you use a field from the Other bucket in your search or query, then it's moved to the Interesting bucket.

Topics:

- [Filter Logs by Source Attributes](#)
- [Filter Logs by Labels](#)
- [Filter Logs by Data Uploaded on Demand](#)
- [Filter Logs by Fields in Log Messages](#)
- [Filter Logs by Hash Mask](#)
- [Filter Logs by Annotations](#)

Filter Logs by Source Attributes

In the **Fields** panel of Oracle Log Analytics, you can use the **Log Source** field to filter logs by the source attributes such as log source and log entities.

For example, to search for logs for a particular log source, such as Database Listener Alert Logs:

1. From Oracle Log Analytics, in the **Pinned** section, click **Log Source**.
2. In the Log Source dialog box, select **Database Listener Alert Logs** and click **Apply**.

 **Note:**

- In the Log Source dialog box, you can see the occurrence trend for the available log sources in the form of sparklines. The sparklines show when the log entries corresponding to the available log sources are generated based on the time range selected in the time selector on the top right corner of the dialog box.
- You can select all the listed items by selecting the checkbox in the header pane on the top left.

Filter Logs by Labels

The labels representing the problem conditions such as `deadlock situation`, `memory issue`, `stuck thread`, `connection issue`, `abnormal termination` and so on are added to the log sources that conform to any of the problem conditions. So, you can filter the logs by specifying the label for the problem condition that you're looking for.

In the **Fields** panel of Oracle Log Analytics, you can use the **Label** field to filter log data by data labels.

1. In Oracle Log Analytics, from the **Visualize** panel, select **Records with Histogram**.
2. From the **Pinned** section, click **Label**.
3. In the Label dialog box, select the label that you want to analyze, such as `CriticalError`, and click **Apply**.

 **Note:**

- In the Label dialog box, you can see the occurrence trend for the available labels in the form of sparklines. The sparklines show when the log entries corresponding to the available labels are generated based on the time range selected in the time selector on the top right corner of the dialog box.
- You can select all the listed items by selecting the checkbox in the header pane on the top left corner of the dialog box.

4. From the **Pinned** section of the **Fields** panel, drag and drop **Label** to the **Display Fields** section of the **Visualize** panel.

Oracle Log Analytics displays all the log entries pertaining to the selected label.

Filter Logs by Data Uploaded on Demand

In the **Fields** panel of Oracle Log Analytics, you can use the **Upload Name** field to filter log data by data uploaded on demand.

For example, to search for uploaded log data for Microsoft SQL Server errors:

1. Ensure that you've uploaded your on-demand log data as specified in [Upload Logs to Oracle Log Analytics on Demand](#).
2. In Oracle Log Analytics, from the **Visualize** panel, select **Records with Histogram**.
3. From the **Pinned** section of the **Fields** panel, click **Upload Name**.
4. In the Upload Name dialog box, select the entry that you want to analyze (for example, **MicrosoftSQLServer_ErrorLog**), and click **Apply**.

 **Note:**

- In the Upload Name dialog box, you can see the occurrence trend for the available uploads in the form of sparklines. The sparklines show when the log entries corresponding to the available uploads are generated based on the time range selected in the time selector on the top right corner of the dialog box.
- You can select all the listed items by selecting the checkbox in the header pane on the top left.

Oracle Log Analytics displays all the log entries for the on-demand upload name.

Filter Logs by Fields in Log Messages

You can search logs by using fields in the **Fields** panel.

The **Fields** panel of Oracle Log Analytics lists the field attributes based on which you can filter log data.

For example, to filter only those logs where the entity type is Oracle WebLogic Server, and the values of the field attribute **Severity** are `ERROR` and `NOTIFICATION`:

1. From Oracle Log Analytics, in the **Fields** panel, click **Entity Type**.
2. In the Entity Type dialog box, select **Oracle WebLogic Server** and click **Submit**.
3. In the **Fields** panel, click **Severity**.
4. In the Severity dialog box, select **ERROR** and **NOTIFICATION**, and click **Submit**.

In the selected *<field name>* dialog box, you can see the occurrence trend for the available field value in the form of sparklines. The sparklines will show when the log entries corresponding to the available field values got generated based on the time range chosen in the time selector on the top right corner of the dialog box.

You can select all the listed items by selecting the checkbox in the header pane on the top left corner of the dialog box.

 **Note:**

Fields, such as Message, which has too many large or distinct values are not eligible to be filtered using the **Fields** panel. See [List of Non-Facetable Fields](#) for the fields that can't be filtered using the **Fields** panel.

If you try to filter such fields, Oracle Log Analytics displays a message that values for the selected field can't be displayed.

However, you can add any such field to the **Display Fields** section.

5. From the **Fields** panel, drag the **Severity** attribute and drop the attribute in the **Display Fields** section in the **Visualize** panel.

Rename a Field

You can use the `rename` command to rename one or more fields.

By renaming system-defined fields, you can control the names of the fields at the time of generating reports. See [Rename Command](#) in *Using Oracle Log Analytics Search*.

For example, to rename the **Host IP Address (Client)** field to `clientip`, in the **Search** field of Oracle Log Analytics, you need to enter the following command and press **Enter**:

```
* | rename 'Host IP Address (Client)' as clientip
```

 **Note:**

Renaming is only a runtime operation, and it doesn't affect the underlying data storage.

Filter Logs by Field Range

For the fields with numerical values, you can use the `bucket` option to group the log records into buckets based on the range of values of a field. The resultant popup window displays the counts and sparkline based on the range buckets instead of distinct values.

1. Click the **Actions** () icon next to the field.

The dialog box displays the following options:

- **Filter:** To display distinct individual values of the field
- **Bucket:** To display the ranges of the field

2. Select **Bucket**.

In the dialog box, you can see the occurrence count for the field in the form of ranges.

When the selected field is rendered in the visualizations such as the pie chart, bar chart, or treemap, the trend will be based on the value ranges and not the distinct individual values.

Filter Logs by Hash Mask

You can use `md5` function in your queries or with `where` and `eval` commands to filter the log data that has the hash masked data.

Typically, when you create a log source and define hash masks to mask specific fields, then the resultant log data will have the hash of the fields that you can use for filtering. To extract those log records that contain the hash masked information of the fields, use the `md5` function in your queries or with `where` and `eval` commands.

For example, consider the following log data:

```
Jul 1,2018 23:43:23 severe jack User logged in
Jul 2,2018 02:43:12 warning jack User logged out
Jul 2,2018 05:23:43 info jane User logged in
```

When the user name information is hash masked, then the log records will be as follows:

```
Jul 1,2018 23:43:23 severe 241fcf33eaa2ea61285f36559116cbad User logged in
Jul 2,2018 02:43:12 warning 241fcf33eaa2ea61285f36559116cbad User logged out
Jul 2,2018 05:23:43 info 8fb2f1187c72aab28236d54f0193a203 User logged in
```

The users `jack` and `jane` will have the following hash values:

```
241fcf33eaa2ea61285f36559116cbad
8fb2f1187c72aab28236d54f0193a203
```

- **Use `md5` function in your search query:** Specify the query `* | md5(jack)` to filter the hash masked records corresponding to the user `jack`.
- **Use the hash with `where` and `eval` commands:** To extract the log records corresponding to the user `jack`, you can use the hash of the user name in the search string `* | where user = "241fcf33eaa2ea61285f36559116cbad"`.
- **Use `md5` function with `where` and `eval` commands:** You can avoid using the hash for the specific user name, and instead, specify the hash mask used. For example, to extract the log records corresponding to the user `jack`, you can provide the search string `* where | user = md5("jack")`.

Filter Logs by Annotations

If you've annotated some of the log records for easy identification or for reuse, then you can filter the logs using those annotations.

By retrieving the annotated log records, you can compare them with a new set of log records when they have similar pattern or to help resolve an issue.

1. From Oracle Log Analytics, in the **Pinned** section, click **Annotation Identifier**.
2. In the Annotation Identifier dialog box, select the specific identifier.

If you want to view all the log records that have annotations associated with them, then select all the identifiers.

Click **Apply**.

The log records with the selected annotation are displayed.

10

Save and Share Log Searches

After you create and execute a search query, you can save and share your log searches as a widget for further reuse. If you've created the widget based on a fixed time range, then every time that you open the widget, it will show the results for the time range that you specified in the search. If you've created the widget for a relative time range (say the last 7 days), then every time that you open the widget, it will show the up-to-date results as per the time selector (Last 7 days).

Using saved searches, other users can also access the search query.

Topics:

- [Save a Search and Add It to a Dashboard](#)
- [Create Alerts for Saved Searches](#)
- [Create a Saved Search from an Existing One](#)
- [Create Alerts for Existing Saved Searches](#)
- [View Saved Search Anomaly Alerts and Baseline Charts](#)
- [Associate Saved Search Alerts with Entities](#)
- [Export the Search Results](#)

Save a Search and Add It to a Dashboard

After you've entered a search query and displayed the results in a chart, to save the search as a widget:

1. Click **Save**.
2. Enter the name and description of the widget.

You can now add this widget to a custom dashboard. See [Create Custom Dashboards](#).

You can view the number of saved searches in your Oracle Log Analytics instance from the Configuration page.

You can also save your search directly to a dashboard. After you've entered a search query and displayed the results in a chart, to save the search to a dashboard:

- a. Click **Save**.
- b. Click the **Add to dashboard** check box.
- c. In the **Dashboard** field, click the down arrow, and select the name of the dashboard to which you want to save the search. If you want to save the search to a new dashboard, then select **New Dashboard** and enter the name of the new dashboard.

Click **Save**.

You can now access the saved search from the specified dashboard.

From Oracle Log Analytics, click the OMC Navigation (☰) icon on the top left corner of the interface.

In the OMC Navigation bar, click **Administration Home**.

Clicking the count of saved searches link displays the Saved Searches page where you can view the list of built-in and custom saved searches. The built-in saved searches are represented with gear icons and the custom ones are represented with human icons.

Click the **Action** icon next to a saved search entry to display the following menu options:

- **Delete:** Lets you delete a custom saved search. A built-in search can't be deleted. In the case of a built-in search, the **Delete** option is grayed out (disabled).
- **View in Log Explorer:** Lets you open the saved search in the Oracle Log Analytics Explorer view.
- **Accelerate Search:** Allows you to accelerate the selected search.

When you save a query as a saved search and enable accelerated search, the query is executed in the back-end periodically and the result is stored. This helps in retrieving the query result for the query's time range in lesser execution time than usual. However, if the saved result data is not accessed for a long time, then the data is deleted for storage optimization.

- **Show Query:** Displays the query used for the search. You can additionally copy the query to the clipboard.

Create Alerts for Saved Searches

You can create alert rules based on saved searches by specifying the threshold, time range, and recipient of the email notification. When the search criteria meets the threshold value over the specified time interval, an alert is generated and an email notification is sent to the specified recipient.

For example, you want your system administrator to be notified with a warning or critical email about any of your monitored targets throwing the `ORA-0600` error message more than three to five times in the past seven days. To do this, you save your search and set an alert rule for it.

1. In Oracle Log Analytics, in the **Search** field, enter the following:

```
ORA-0600 | stats count by Target
```

2. From the **Search Dates** list, select **Last 7 Days** and click **Run**.
3. Click **Save**.
4. In the Save Search dialog box, enter the search name.

You can click **Add Search Description** and enter an optional description for the search.

5. Click **Create alert rule**.

In the **Rule Name** field, enter a rule name.

You can click **Add Rule Description** and enter an optional description for the rule.

6. For **Condition Type**, select **Fixed Threshold** or **Anomaly**.

The anomaly based alert rule will be automatically enabled after the data is collected for 30 intervals.

You can save a maximum of 50 scheduled alerts.

7. For **Operator**, select **>**, for **Warning Threshold**, enter 3, and for **Critical Threshold**, enter 5.
8. For **Schedule Interval**, specify 7 **days**.

You can select **Every Hour**, **Every Day**, **Every Week** or a **Custom** setting for any value between 15 minutes to 21 days as the Schedule Interval. Your saved search runs automatically based on the interval that you specify.

If you select **Every Hour**, then you can optionally specify to exclude **Weekend** or **Non-business hours** from the schedule.

If you select **Every Day**, then you can optionally specify to **Exclude Weekend** from the schedule.

9. If you want to customize your alert message, then under **Customize Message Format**, select **Use custom message**. You can customize any or all of the messages available under this section. For details, see Step 8 in [Create An Alert Rule](#).
10. In **Notifications**, specify the recipients of the alert notifications and in **Remediation Action**, select the action that must be performed automatically in response to an alert. For details, see Step 9 and Step 10 in [Create An Alert Rule](#).
11. Click **Save**.

Over a period of 21 days, whenever any of your monitored entities throws the `ORA-0600` error more than the specified threshold value, an email will be sent to the specified recipient listing each entity (along with the count of the error) that crossed the threshold. The email also includes a link to the Oracle Log Analytics user interface. Clicking the link takes you to the search results for the specific time range when this alert was triggered.

Create a Saved Search from an Existing One

Use the **Save As** option to customize a built-in or custom saved search.

1. In Oracle Log Analytics, click **Open**.
2. In the Open dialog box, select the saved search that you want to modify and click **Open**.
3. Update the search criteria based on your requirement, click the **Save** list, and select **Save As**.
4. In the Save Search dialog box, enter a name for the updated search. Optionally, you can create an alert for the new search.
5. Click **Save**.

The new search now appears in your list of saved searches.

Note:

The **Save** option is disabled for a built-in search, and you can perform a **Save As** operation only to save the updated, built-in search as a new one.

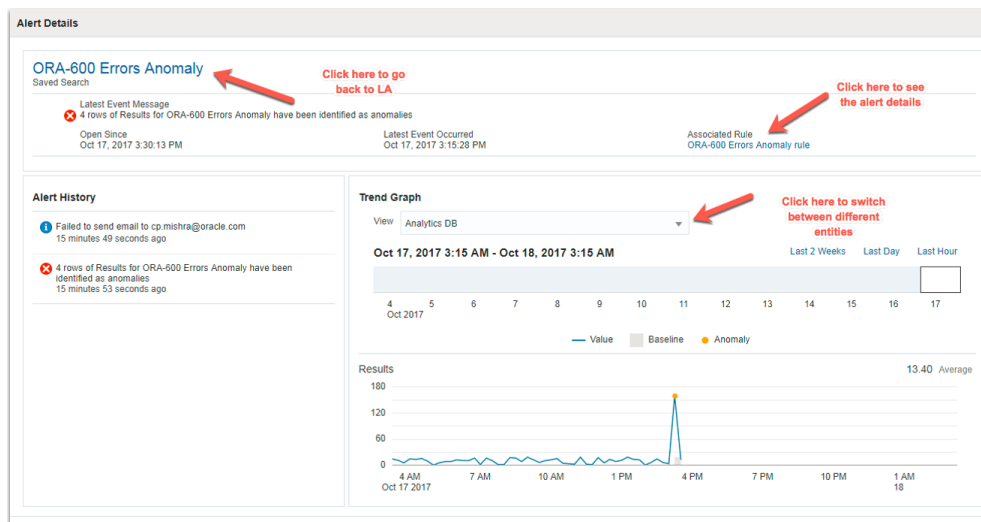
Create Alerts for Existing Saved Searches

1. In Oracle Log Analytics, click **Open**.
2. In the Open dialog box, search for the saved widget, such as ORA-0600 by Target, select the widget, and click **Open**.
3. Click **Alert Rules** (⚙️ **Alert Rules**).
4. In the Alert Rules dialog box, click **Create Alert Rule**.
The Create Alert Rule dialog box is displayed. Because you're creating the alert for an existing search, in the Create Alert Rule dialog box, the search name and the search description are populated with the values that you provided when you saved the search.
5. Enter the rule name.
6. Specify the rule details. See Steps 6 through Step 10 in [Create Alerts for Saved Searches](#).

View Saved Search Anomaly Alerts and Baseline Charts

1. From Oracle Log Analytics, click the OMC Navigation (☰) icon on the top left corner of the interface. In the OMC Navigation bar, click **Alerts**.
You can view the list of alerts with details such as **Severity**, **Message**, **Entity**, **Entity Type**, **Last Updated**, and **Duration**.
2. Click the message corresponding to the anomaly alert that you've set.
You can view the alert details.
3. Click **View more details**.

The interface displays **Alert History** and **Trend Graph** for the anomaly alert that you selected. The graph displays the anomalies detected and the baseline for the recorded data.



- To view the trend graph corresponding to the entity of your choice, click the **View** list, and select the entity.
- To view the alert rule, click the link adjacent to **Associated Rule**.
- To return to Oracle Log Analytics, click the name of the saved search on the top left corner of the interface.

Associate Saved Search Alerts with Entities

Typically, saved search alerts are associated with the **saved search** entity type. However, to trigger actions on entities in response to the alerts, the alerts must be associated with the specific entities. For example, if an alert is raised on a Linux host entity, then a restart action can be triggered in response to the alert.

To associate a saved search alert with an entity while creating a saved search alert:

1. Group the log records by target. For example:

```
Exception | stats count by target
```

2. On the result of the query, apply an **Entity Type** filter OR an **Entity** filter. For example, select `WebLogic Server` entity type.
3. Click **Save**.
4. Enter the name for the saved search alert.
5. Click the **Create alert rule** check box.
6. Enter the rule details and save the search.

You can now view the saved search alert that you created in the alert rules list. The alert is now associated with a specific entity type and not `Saved search` entity type.

Export the Search Results

Oracle Log Analytics lets you export search results in Comma-separated Values (CSV) or JavaScript Object Notation (JSON) format.

To export search results:

1. Search for logs for a set of entities. See [Search Logs by Entities](#).
2. Click **Export**.
3. For the file format, select **Comma-Separated Values** or **JavaScript Object Notation**.
4. Enter a name for the file and click **Export**.

In the case of the Records and Histogram visualizations, the search result based on time, original log content, and all the selected display fields is exported. In the case of Table visualization, the search result based on the time and selected display field is exported. For any other visualization, the results of the query displayed in the selected visualization is exported.


11

Create An Alert Rule

Create an alert rule that generates an alert when an anomaly or a deviation from the fixed threshold is detected in the log data.

Other Topics:

- [View and Edit Alert Rules](#)
- [Generate Inline Alerts](#)
- [View the Entity Details for an Alert](#)

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Alerts Rules**.
2. In the **Service** list, select **Log Analytics**.
3. Click **Create Alert Rule** on the top right corner of the window.

The **Create Alert Rule** dialog box opens.

4. In the **Rule Name** field, enter the rule name.
5. Click **Add Description**, and provide details about the rule that you're creating.
6. In the **Search Name** list, select the name of the saved search that the alert rule must be associated with.
7. In the **Rule type** field,
 - to create a scheduled alert, select **Scheduled** alert option.
 - a. For **Condition Type**, select **Fixed Threshold** or **Anomaly**.

The anomaly based alert rule will be automatically enabled after the data is collected for 30 intervals.
 - b. In **Results**, specify the details of the condition in the **Operator**, **Warning Threshold**, and **Critical Threshold** fields.
 - c. Enter the periodicity of the rule in **Schedule Interval**.

You can select any value between 15 minutes to 7 days as the Schedule Interval. Your saved search runs automatically based on the interval that you specify.

You can save a maximum of 50 scheduled alerts.
 - to create a real time alert that's triggered by the presence of a label in the log records, select **Real Time** alert option.
 - a. In the **Entity Type** field, click the down arrow, and select your entity type.
 - b. In the **Label** field, click the down arrow, and select the label for which you want to generate the alert.
 - c. In the **Log Source** field, enter the name of the log source.

 **Note:**

Alerts would get generated for the logs only if their entity is specified.

8. If you want to customize your alert message, then under **Customize Message Format**, select **Use custom message**. You can customize any or all of the following messages:
 - **Warning:** This message is generated when an alert is marked as a warning alert. The warning alert is triggered when the metric associated with it violates the warning threshold value as defined in the rule.
 - **Critical:** This message is generated when an alert is marked as a critical alert. The critical alert is triggered when the metric associated with it violates the critical threshold value as defined in the rule.
 - **Clear:** This message is generated when an alert is cleared. The clear message is sent when the metric associated with it no longer violates the warning or critical thresholds.

Format of the Custom Message:

You can enter any text in the text field next to the type of the message. Additionally, you can insert system values in the message by using the predefined tokens **Available Message Tokens**, each of which will be substituted in the actual message by the value it refers to. Expand the **Available Message Tokens** section to view the table that lists the tokens and provides their details. For example, to create the following custom message for the critical alert:

A critical alert has been generated because the value **2000** exceeds the designated threshold of **1500**.

Enter the following text in the **Critical** text field to generate the above message:

```
A critical alert has been generated because the value %{sys.value}%
exceeds the designated threshold of %{sys.criticalThreshold}%.
```

In the above message, the token `%{sys.value}%` is replaced by the actual value **2000**, and the token `%{sys.criticalThreshold}%` is replaced by its actual value **1500**.

Important: Enclose the tokens in the *percentage - curly bracket* characters, for example `%{some-token}%`.

9. Under **Notifications**, you can specify the recipients to receive notifications when any result violates the specified threshold.

Notification Channels: Classes of notification destinations are called *notification channels*. Notification channels allow you to set up and reuse functional groups of notification recipients, such as regional administrators, IT managers, or other Web servers without having to specify large numbers of individual destinations repeatedly. Once you set up a notification channel, you can reuse the channels across different alert rules.

- **Email:** Specify the email address or email notification channels. To create a new email channel:


- a. Click **Email Channel**.
 - b. In **Channel Name**, enter the name of the new email channel that you're creating.
 - c. In **Email Addresses**, enter a comma-separated list of recipient email addresses to include in the channel that you're creating.
 - d. Click **Create**.
- **Mobile:** Specify the user names or mobile notification channels. To create a new email channel:
 - a. Click **Mobile Channel**.
 - b. In **Channel Name**, enter the name of the new mobile channel that you're creating.
 - c. In **OMC User Names**, enter a comma-separated list of user names to include in the channel that you're creating.
 - d. Click **Create**.

 **Note:**

Oracle Management Cloud Mobile app must be installed and signed into before a user can receive a push notification. The Oracle Management Cloud Mobile app can be downloaded on the app store.

- **Integrations:** From the list, select the integration notification channel.
In addition to notifying people, Oracle Log Analytics can also send relevant information to third-party web applications (such as *Slack* or *Hipchat*) if an alert is raised, thus allowing you extend Oracle Log Analytics functionality by having third-party applications carry out actions in response to an alert notification. This type of system integration is achieved using *WebHooks*; an HTTP POST message containing a JSON payload that is sent to a destination URL. When an alert is raised, you can have that alert sent to *PagerDuty* or *ServiceNow* for incident management.
To create an integration notification channel, see *Set Up Notification Channels in Using Oracle Infrastructure Monitoring*.
10. Under **Remediation Action**, from the list, select the remediation action that must be performed automatically in response to an alert.
You can create a Remediation Action using the Event Service API. Contact your Oracle Support or Sales Representative for more information about accessing and using the Event Service API.
 11. Click **Save**.

View and Edit Alert Rules

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Alerts**.
2. Click **Alert Rules** on the top right corner of the window.
3. Click the name of the alert rule to view and edit.

**Note:**

You can also delete an alert rule by clicking the Delete icon next to the alert rule name.

Generate Inline Alerts

You can define alerts such that the anomalies are detected based on the inline content of the logs. This can be done by associating an alert with a label that's tagged for the log records from a specific log source and entity type.

To generate inline alerts, first edit the log source to add a label on detecting the specific content in the log record. Next, associate the log source with an entity type. Lastly, define a real time alert rule on the specific target type, label and log source. For example, edit the source `mvHostSrc2` and add a label `invalid_usr` that tags the user name `anonymous`. Next, associate the log source `mvHostSrc2` with the entity `Host (Linux)`. Lastly, create a real time alert rule that raises an alert every time a log record containing the user name `anonymous` is encountered by associating the alert with the label `invalid_usr`, log source `mvHostSrc2`, and entity `Host (Linux)`.

1. Edit the log source, and add a label for the specific log record content. For example, add a label `invalid_usr` when the user name is `anonymous`. See [Use Labels in Log Sources](#).
2. Associate the log source with an entity type. See [Work with Entity Associations](#).
3. Create an alert rule for the specific log source, label, and entity type. See [Create An Alert Rule](#).

In the **Rule type** field in the **Create Alert Rule** dialog box, select **Real time** alert option. The following are some example values that you can use while creating the alert rule:


- In the **Rule Name** field, enter `testAlertRule2`.
- In the **Entity Type** list, select `Host (Linux)`.
- In the **Label** list, select `invalid_usr`.
- In the **Log Source** field, enter `mvHostSrc2`.

When the tag that you specified in the log source is encountered, an alert is raised. For example, an `invalid_usr` alert is raised for the log record when the user name is `anonymous`.

Click the message to view the alert details.

View the Entity Details for an Alert

To analyze the alert and identify the log entry that corresponds to the alert:

1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Alerts**.

You can view the list of alerts with details such as **Severity**, **Message**, **Entity**, **Entity Type**, **Last Updated**, and **Duration**.

2. In the row corresponding to your alert, hover your cursor on the entity name.
A pop up window with the entity name opens.
3. Click the **View More** icon.
A pop up window opens with details of the entity.
4. Click the down arrow next to **View Entity**. Select **Log Analytics**.
The Entity page opens in Oracle Log Analytics.

You can now view the details of the entity that corresponds to the alert.

Transform Logs into Operational Insight

Oracle Log Analytics lets you transform log data into operational insight, to understand the performance of your entity and apply corrective actions, if required.

Topics:

- [Typical Workflow for Developing Operational Insights](#)
- [Use Sample Log Data](#)
- [Compare the Log Records](#)
- [Use Out-of-the-Box Widgets](#)
- [Create Custom Dashboards](#)
- [Generate Log Metrics](#)

Typical Workflow for Developing Operational Insights


Here are the common tasks for transforming log data into operational insight.

Task	Description	More Information
Save and share log searches.	Save a search query as a widget so that you can run the widget to retrieve latest results.	See Save and Share Log Searches and Export the Search Results .
Visualize data.	Present Search results graphically for easier analysis.	See Visualize Data Using Charts and Controls .
Create widgets.	Create a widget by saving a search or customize out-of-the-box widgets to suit your requirement.	See Save and Share Log Searches and Use Out-of-the-Box Widgets .
Create custom dashboards.	Create custom dashboards by using widgets.	See Create Custom Dashboards .

Use Sample Log Data

Use the sample log data that's available in Oracle Log Analytics at no additional cost to see the working of the features end-to-end.

The sample log data is available to explore the working of the Log Explorer and the dashboard reporting features. By using the sample log data, some sample entities, saved searches, and dashboards are created which are visible in other services of Oracle Management Cloud.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Administration Home**.
2. Under **Other Links**, click **Sample Data**.
The Sample Data page opens.

3. Enable the Sample Data button.

Your data remains separated from the sample log data at all times. To design your queries around the sample log data, prefix `demomode |` in your queries.

You can disable the use of sample log data at any time.

Compare the Log Records

Some visualization options in Oracle Log Analytics let you compare two log records and display the changes in patterns.

Some sample use cases where you may want to compare log records are:

- Determine what was different in the log stream right after a failure as compared to a normal period.
- Compare events right after a software deployment.

You can compare log records in the following visualizations only:

- Records with Histogram
- Records
- Table with Histogram
- Table

1. Search for logs for a set of entities. See [Search Logs by Entities](#).**2. Select a supported visualization.****3. Right-click a record that you want to compare and select **Add To Compare**.**

A floating window with the selected record appearing on the left side is displayed.

4. Right-click the record with which you want to compare the first selected record and select **Add To Compare.**

The right side of the floating window is populated with the newly selected records.

5. Click **Compare in the floating window.**

The Log Entry Comparison window displays the comparison.

Use Out-of-the-Box Widgets

Oracle Log Analytics provides a set of out-of-the-box widgets that you can use in a dashboard.

General	Oracle Fusion Middleware	Oracle Database
Top Log Sources	Oracle Middleware Logs Trend	Oracle Database Log Trend
All Logs Trend	Oracle WebServer Top Accessed Pages	Oracle Database Errors Trend
Critical Incidents by Target Type	Oracle WebServer Failed HTTP Requests	Oracle Database Top Errors
Oracle Enterprise Manager Log Trend	Top Oracle WebServer Targets by Requests	Top Oracle Database Targets with Errors
Host Log Trend	Top Oracle Middleware Error Codes	

General	Oracle Fusion Middleware	Oracle Database
Invalid User Login Attempts	Top ECIDs with BEA-x Error Codes	
Failed Password Attempts	Top Oracle Fusion Middleware Targets with Errors	
Top Commands Run with SUDO	Oracle WebServer Top Accessed Pages (Excluding Assets)	
Top Hosts by Log Entries	Oracle WebServer Top Users by Pages (Excluding Assets)	
Top Host Log Sources		
Top SUDO Users		
Access Log Error Status Codes		
OIC Adapter Error Groups		

Create Custom Dashboards

You can create custom dashboards on the **Dashboards** page by adding out-of-the-box widgets or the custom widgets you've created. You can also create a duplicate of one of the available dashboards and customize it to meet your requirements.

1. Select **Dashboards** in the Management Cloud navigation menu.
2. On the Dashboards page, click **Create**.
3. In the **Create Dashboard** dialog box, select **Dashboard (a single dashboard)**.
4. Specify the name of the dashboard and optionally, a description, and click **Create**.
5. On the new dashboard page, click **Edit** to add widgets.

The **Add Widget** pane is displayed.

6. Click a widget in the **Add Widget** pane to add it to the dashboard.

After you've added a widget, you can click the **Content Settings** (⚙️) icon on the widget to perform various actions such as altering the size or placement of the widget, hiding or adding the title, linking the title of the widget to an Oracle Management Cloud page or dashboard, and removing the widget.

7. Optionally, you can click **Text/HTML Widget** at the bottom of the **Add Widget** pane to customize your dashboard by adding a text widget with a header, your company logo, or an HTML or email link.
8. Click **Done Editing** to save the new dashboard.

After you click **Done Editing**, the dashboard is in View mode. At a later time, if you want to make changes to the contents of your dashboard, click **Edit** to go to the Edit mode.

After you've created a dashboard, you can use the following options on the Dashboards page to perform other tasks:

- Click the **Open in Data Explorer** (🔗) icon on a widget to open the widget in Data Explorer and make changes. If you want to make changes to a widget created by another user or to an out-of-the-box widget, you can click the **Save As** option in Data Explorer to create a copy of the widget and make changes to meet your requirements. Note that the

copy of the widget isn't automatically added to the dashboard and you'll have to edit the dashboard and add it.

- Click the **Favorite the dashboard** (☆) icon next to the title of the dashboard. The dashboard is then displayed in the Dashboards menu and is easier to access.
- Click **Filter** to enable expression-based filtering by Entity Type, Entity Status, and Tags.
- Click the **Auto-refresh** drop-down list to set a time for auto-refresh or disable it.
- Click **More** to print the dashboard, set the dashboard as your home page, share the dashboard with other users, duplicate the dashboard, set global entity and time selector options, and delete the dashboard.


Dashboard Collaboration Options


As mentioned above, after creating a dashboard, you can click **More > Share with Others (view-only) > On** to allow all the users in the same tenant to view your dashboard. In addition, you can use REST API to perform advanced dashboard collaboration tasks such as sharing your dashboard with selected users in the same tenant and allowing them to edit the dashboard. For information on REST API for dashboard collaboration, see [Working with Dashboards](#) in *Oracle Management Cloud Common REST API*.


Note that if you have the OMC Administrator role, you can view and edit dashboards created by other users in the same tenant.

Generate Log Metrics

Identify the key performance indicators from your logs to monitor automatically and generate metrics with them for ready access. These metrics can be stored longer than the original logs to save cost.

1. From Oracle Log Analytics, click the OMC Navigation  icon on the top left corner of the interface. In the OMC Navigation bar, click **Log Admin**, and click **Log Metrics**.
2. In the **Log Metrics** section, click **Create**. The Create Log Metric page is displayed.
3. Select the **Entity Type** of your logs. You can define up to 25 metrics for each entity type.
4. Optionally, select the **Log Source** to refine the set of logs to generate the metric.
5. Enter a **Metric Name** and optionally, provide a description to help you identify the metric from the list at a later point.
6. To enable the metric collection, check the status **Enabled** check box.
7. To define the aggregation function that must be employed to generate the metric, under the **Metric is calculated as...**,
 - **Aggregation Function:** From the menu, select **Count**, **Sum**, or **Average**. This is the operation that must be performed on the selected set of logs.
 - **Aggregation Interval:** Enter a number between 60 and 600 seconds to define the interval.
 - **Metric Unit:** Select the unit of measurement that must be used while displaying the metric in a visualization.

- **Does this metric need to be grouped:** Select the **Yes** button if you want the metric to be grouped by a field, and select the field from the menu.
8. You can optionally define a condition that must be recorded as an event in the metric, by specifying the **Field**, **Operator**, and **Condition**. Each time the condition is satisfied, the event is recorded on the metric.
 9. Click **Save**.
The log metric page opens and the new metric that you created is displayed in the table.
 10. To view the visualization of the metric, click the  icon next to the metric name and select **View in Log Explorer**.

You can edit or delete a metric at any point in time by clicking the  icon next to the metric name.

Part IV

Typical Use Cases

Review some of the scenarios where you can use Oracle Log Analytics.

Topics:

- [Example Scenario: Perform Dynamic Log Analysis](#)
- [Example Scenario: Detect Anomalies Using Outliers](#)
- [Perform Advanced Analytics with Link](#)
- [Parse Log Records with Multiple Timestamps](#)
- [Perform Advanced Analytics with Cluster Compare](#)
- [Machine Learning Based Query Enrichment](#)
- [Examples of Semantic Clustering Using Natural Language Processing](#)

Also, see *Analyzing Host Log Trends to Proactively Monitor Infrastructure* ( [Tutorial](#)).

Using the *link* and *link by clusters* features, you can detect anomalies, potential issues, and outliers in your logs. You can also generate alerts and get notifications when any of these events occur. See [Generate Link Alerts](#) and [Generate Alerts for Cluster Utilities](#).

Example Scenario: Perform Dynamic Log Analysis

You can explore logs to diagnose and troubleshoot issues at any time.

Procedures and scenarios described in this chapter use an example application named RideShare targeted at customers interested in carpool and vanpool services. As a DevOps administrator, you're responsible for troubleshooting problems related to this application, which is critical to your business. When customers book rides using the RideShare web application, logs related to this transaction are sent to Oracle Log Analytics, and the RideShare application dashboard is updated in near real time. The updates include the number of rides accepted by users, the category of cars or rides being requested by users, and the regions around the country from where the rides are being requested.

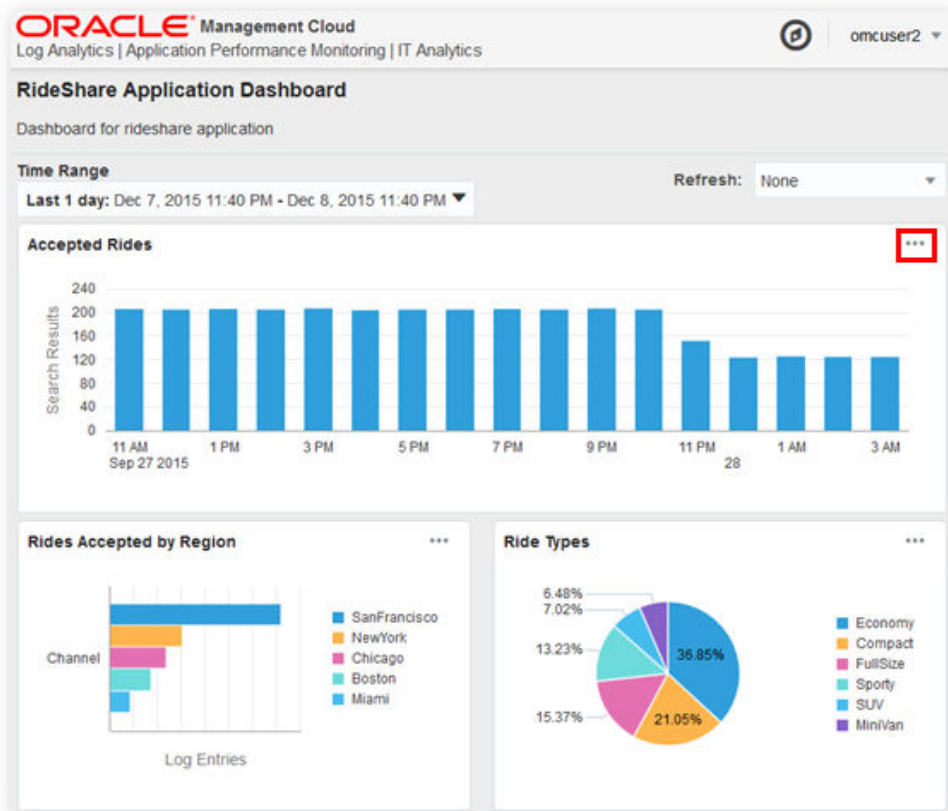
John, one of your ride operators, receives complaints from users that they can't book rides. He contacts your application support team and requests help. As a DevOps administrator, you have to troubleshoot this problem, because it affects your business.


You've built a custom dashboard, the RideShare application dashboard in Oracle Management Cloud, to help you manage routine administration tasks. The dashboard helps you understand the following aspects of your online ride-sharing application:

- Number of rides being processed every hour
- Types of rides that are being requested, such as Economy, Compact, SUV, and so on
- Regionwise location of the customers

Start troubleshooting by:

1. Open the RideShare Application dashboard and click the **Configure widget** icon (the three dots) on the top right corner of the Accepted Rides widget and select **Edit** to view the log entries for the processed rides in the Oracle Log Analytics Data Explorer.



2. From the **Visualize** panel, select **Records with Histogram** .
3. From the **Pinned** section of the **Fields** panel, click **Log Source**.
 - In the Log Source dialog box, select the required log sources for that entity, and click **Apply**.
4. From the **Pinned** section of the **Fields** panel, click **Severity**.
 - In the Severity dialog box, select the required entry (**ERROR** in this case), and click **Apply**.

You select **ERROR** because you deduce that the incomplete bookings are due to some errors in the application servers.

Oracle Log Analytics displays all the transactions that have errored out. In this example scenario, you saw errors related to the application server infrastructure used by the RideShare application. Drill down to logs related to the application server instances by selecting a specific application server target or a group of targets. See [Search Logs by Entities](#).

Example Scenario: Detect Anomalies Using Outliers

Using Oracle Log Analytics, you can:

- Reduce millions of log events into a smaller set of patterns
- Rapidly troubleshoot problems by identifying log records that're behaving different when compared to the expected behavior and intermittent errors


Intelligent drill-down and pivoting gives you additional insight into the cause of the problem by showing a chronological log of entries preceding and following events of interest.

Learn how to use Oracle Log Analytics to troubleshoot the cause for the drop in the number of rides on the online application RideShare.

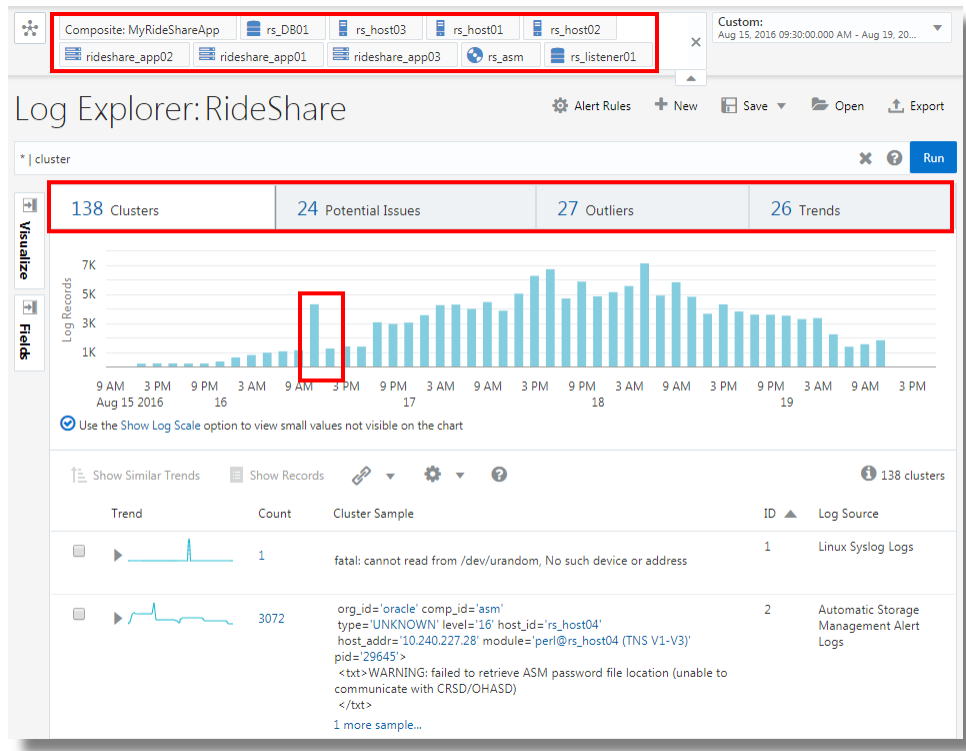
Jane, one of your ride operators, notices that between 10 p.m. and 11 p.m., there was a sudden drop in the number of rides that were processed. She contacts your application support department and requests help. As the DevOps administrator, you have to troubleshoot this critical problem, because it affects your business.

1. In the RideShare application dashboard, start by filtering the number of processed rides between 10 p.m. and 11 p.m.

The dashboard shows a sudden dip in the number of processed rides between 10 p.m. and 11 p.m. To troubleshoot this issue, drill down into the details to find out the problem with the application.

2. Click **Open in Data Explorer**  on the top right corner of the Accepted Rides widget and select **Edit** to view the log entries for the processed rides in the Oracle Log Analytics Data Explorer.
3. In the **Fields** panel, click **Entity**, in the **Entity** dialog box, select all the three hosts and the three applications, and then click **Submit** to expand your search to include the hosts on which the applications are running.

Note that the search returned more than 29,000 log entries. Because it's difficult to analyze so many log entries, try to look for any patterns in these entries by using the `cluster` command.



4. In the **Search** field, enter `* | cluster` and press **Enter**.

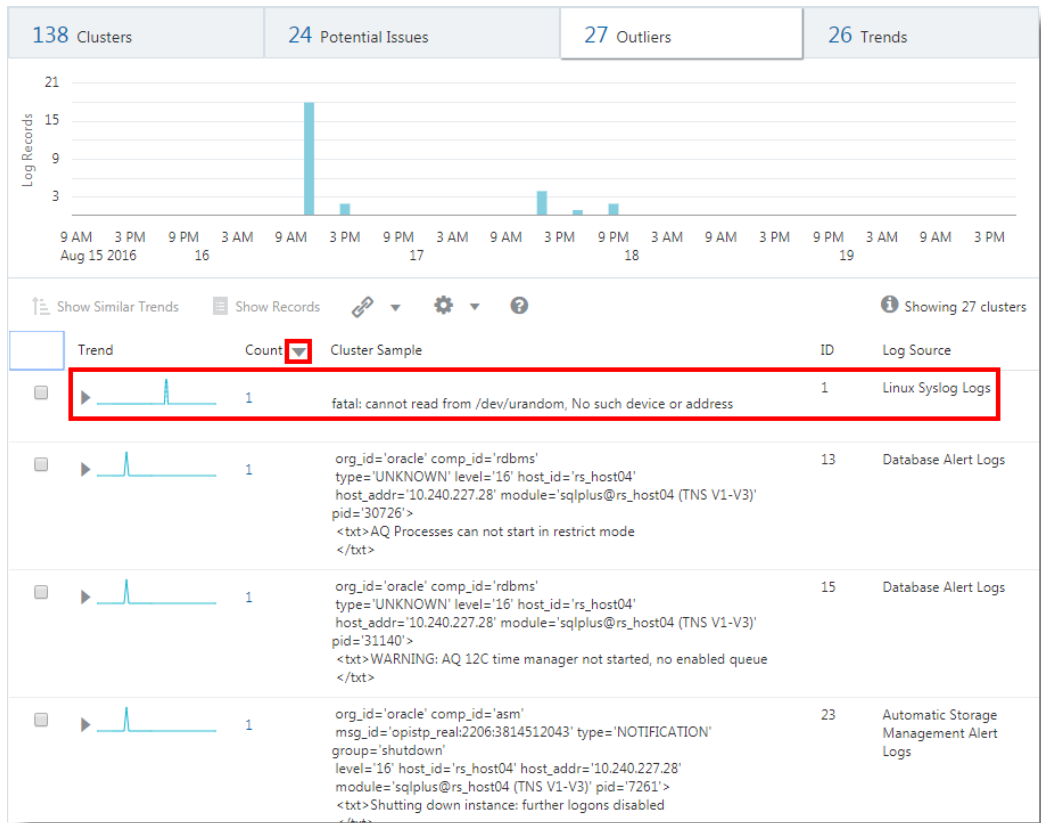
The `cluster` command uses machine learning to group log records together based on how similar they are to each other. See *Cluster Command* in *Oracle Log Analytics Search Language Reference*.

Here, the `cluster` command reduces the large number of log entries into a small number of patterns.

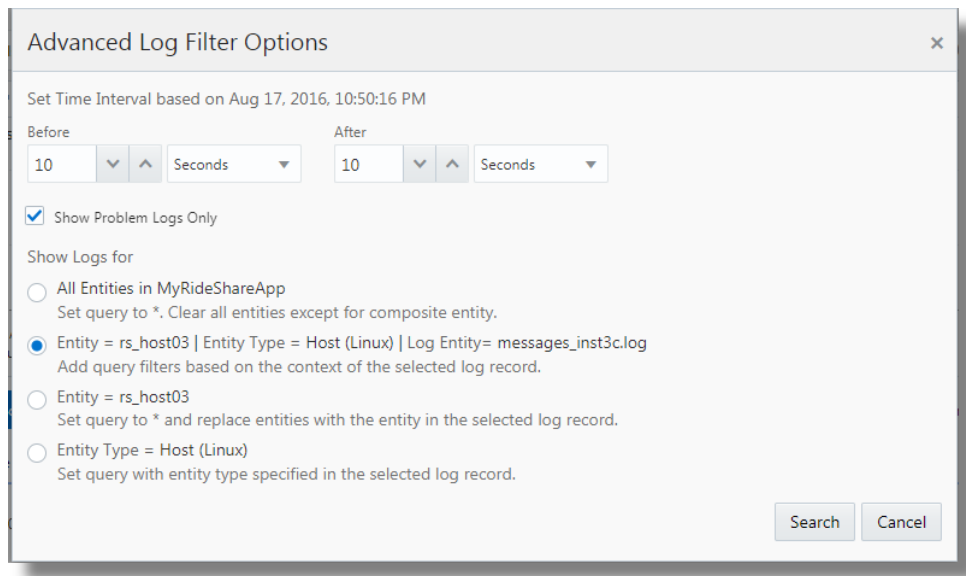
5. Click the right end of the **Count** column header to sort the messages in reverse order to see which patterns have a small number of entries.

After the log entries are sorted in the reverse order of message count, you can see some outlier signatures. Outliers are events that occur rarely. Drill down into an outlier to explore further.

You can see that a log message has returned a fatal error.



6. In the **Count** column of the log message with the fatal error, click **1** to display the relevant record.
7. In the log records section, click the menu icon or right-click the record and select **Show Logs +/- 1 Minute** to see more context for this outlier entry.
You can see all the log entries that were generated in that 1-minute context.
You can see that someone had run the `chmod` command to change permissions on some files. That's probably the cause of the problem.
8. To investigate further, you can use the advanced log filter options to set a time interval for displaying the log data from a specific entity type or entity. In the log records section, click the menu icon or right-click the record and select **Advanced Log Filter Options**.
The **Advanced Log Filter Options** dialog box gives you options to filter the logs:



You can use the advanced options to drill down on a specific time range and quickly pick options to add as a query filter.

- **All Entities:** If you previously selected specific entities, then this option will clear the selection. If a composite entity is selected, then it will be retained. The query is set to * to filter all the logs for all the entities. This will find all the logs for the specified time range.
- **Entity = <entity> | Entity Type = <entity type> | Log Entity = <log entity>:** This option includes the specific filters for the selected row including entity, entity type and log entity, and add them to the query. This is similar to the 1-minute, 5-minutes, and 10-minutes contexts, but helps you to set a specific time range. With this selection, you can find all the logs matching the row selection criteria.
- **Entity = <entity>:** This option includes only the entity filter and adds that to the query. With this selection, you can find all the logs for the selected entity.
- **Entity Type = <entity type>:** This option includes only the entity type filter and adds that to the query. With this selection, you can find all the logs for a selected entity type.
- **Show Problem Logs Only:** Select this to view only the problem logs from the result of the other selections.

Perform Advanced Analytics with Link

Understand the application of the Link feature in performing advanced analytics with the use-case discussed in this topic.

For the steps to use the Link feature to analyze your log records, see [Link Visualization](#).

Example Scenarios:

Use Case	Link Feature	Example Logs
Visualize Time Series Data Using the Link Trend Feature	<i>Link Trend</i>	EBS Concurrent Request Logs
Use <code>timestats</code> Command for Time Series Analysis	<i>Using <code>timestats</code> command after <code>link</code> command</i>	Application Access Logs
Cluster Similar Time Series	<i>Using <code>timecluster</code> command after <code>link</code> command</i>	-
Analyze the Access Logs of Oracle WebLogic Server	<i>Link basic features</i>	FMW WLS Server Access Logs
Use Dictionary Lookup in Link	<i>Annotate Link results</i>	FMW WLS Server Access Logs
Generate Charts with Virtual Fields	<i>Using virtual fields for charts</i>	SAR CPU Logs
Link by Using SQL Statement as the Field of Analysis	<i>Using SQL statement as a field</i>	Database Audit Logs, Database Audit XML Logs, Oracle Unified DB Audit Log Source Stored in Database 12.1
Analyze the Time Taken Between Steps in a Transaction	<i>Time analysis</i>	Access Logs
Generate Charts for Multiple Fields and their Values	<i>Charts for multiple fields and their values</i>	-
Second Level Aggregation Using <code>Eventstats</code> Command in Link	<i>Second level aggregation</i>	Access Logs
Use Link Navigation Functions to Identify Events in a Database	<i>Navigation functions</i>	Database Alert Logs
Use the Currency Symbols in Your Log Analysis	<i>Using currency symbol in groups table and charts</i>	Gasoline Prices

Visualize Time Series Data Using the Link Trend Feature

Link is used to group the log records by specific fields. The various statistics that you can extract from these groups can be visualized using the bubble chart visualization. The bubble chart visualization is now enhanced to support the Time field as an axis.

The following steps explain how to use the trend feature to analyze the job duration for Oracle E-Business Suite (EBS) Concurrent Requests. Oracle Log Analytics provides out-of-the-box support for EBS Concurrent Request Logs.

Consider the following sample log in the filepath `/u01/oracle/appl_top/req/17474445.req`:

```
Human Resources: Version : 12.2
```

```
Copyright (c) 1998, 2013, Oracle and/or its affiliates. All rights reserved.
```

```
AME_MIGRATIONB: Approvals Management Post Upgrade Process
```

```
+-----+
```

Current system time is 24-JUL-2018 01:04:29

+-----+
-----+

Starts24-JUL-2018 01:04:30

Ends24-JUL-2018 01:04:30

Migration of item class usages successful

+-----+
-----+

Start of log messages from FND_FILE

+-----+
-----+

End of log messages from FND_FILE

+-----+
-----+

+-----+
-----+

No completion options were requested.

Output file size:

0

Deleting empty output file.

+-----+
-----+

Concurrent request completed successfully

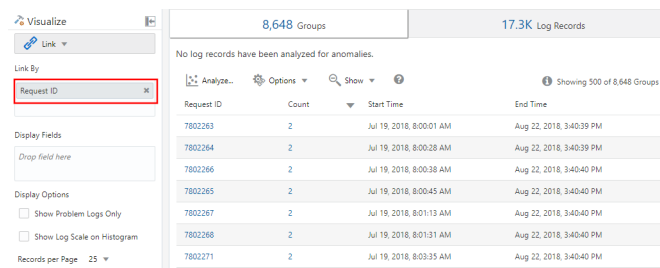
Current system time is 24-JUL-2018 01:04:32

+-----+
-----+

The out-of-the-box log source *EBS Concurrent Request Logs - Enhanced* extracts the *Request ID* field from the filepath. For example, the numeric data 7474445 is the Request ID extracted from the filepath of the above sample log. The log source also extracts the associated metadata for each Request ID.

1. Select the log source and switch to Link visualization:

In the **Fields** panel, click **Log Source** > Select the **EBS Concurrent Request Logs - Enhanced** log source > Switch to the **Link** visualization > Drag and drop the **Request ID** field to **Link By** panel to get the list of requests:



The auto-generated query looks like this:

```
'Log Source' = 'EBS Concurrent Request Logs - Enhanced' | link 'Request ID'
```

2. Extract the request start and end time:

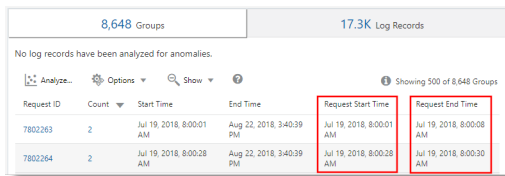
Each request has a start time and an end time printed in the file. If the end time is absent, then the time at which the file is updated is considered as the end time. The log source is configured to capture these values as *Event Start Time* and *Event End Time* fields.

Edit the query to extract these fields:

```
'Log Source' = 'EBS Concurrent Request Logs - Enhanced'  
| link 'Request ID'  
| stats earliest('Event Start Time') as 'Request Start Time',  
latest('Event End Time') as 'Request End Time'
```

`earliest` is a function of `stats` command. This sorts the records of each Request ID by time and returns the oldest *Event Start Time*. Similarly, `latest` returns the last *Event End Time*.

You can now view the new fields in the records table:



Request ID	Count	Start Time	End Time	Request Start Time	Request End Time
7802263	2	Jul 19, 2018, 8:00:01 AM	Aug 22, 2018, 3:40:39 PM	Jul 19, 2018, 8:00:01 AM	Jul 19, 2018, 8:00:08 AM
7802264	2	Jul 19, 2018, 8:00:28 AM	Aug 22, 2018, 3:40:39 PM	Jul 19, 2018, 8:00:28 AM	Jul 19, 2018, 8:00:30 AM

Request Start Time and *Request End Time* are automatically detected as timestamps and formatted in your local timezone. When the files are collected, the agent uses the EBS database timezone to interpret the timestamps.

Note:

To ensure that the database timezone is displayed as expected in Oracle Infrastructure Monitoring configuration home, and to avoid mismatch in the values, provide the timezone during the upload.

3. Compute request duration:

Now that we have the start and end times for each request, we can compute the duration as the difference between these two fields.

Change the query suitably:

```
'Log Source' = 'EBS Concurrent Request Logs - Enhanced'  
| link 'Request ID'  
| stats earliest('Event Start Time') as 'Request Start Time',  
latest('Event End Time') as 'Request End Time'  
| eval 'Time Taken' = 'Request End Time' - 'Request Start Time'
```


Time Taken is a new field created for each Request ID group. This would contain the difference between the request start and end Time.

Request ID	Count	Start Time	End Time	Request Start Time	Request End Time	Time Taken
7802263	2	Jul 19, 2018, 8:00:01 AM	Aug 22, 2018, 3:40:39 PM	Jul 19, 2018, 8:00:01 AM	Jul 19, 2018, 8:00:08 AM	7 sec
7802264	2	Jul 19, 2018, 8:00:28 AM	Aug 22, 2018, 3:40:39 PM	Jul 19, 2018, 8:00:28 AM	Jul 19, 2018, 8:00:30 AM	2 sec

Note:

Oracle Log Analytics automatically detects *Time Taken* as a duration field, since it is produced by the difference between two timestamp fields. Therefore, it is automatically formatted in a human readable way.

4. Trend for the time taken by the EBS Concurrent Requests :


The *Time Taken* field can now be analyzed for trends. Click the **Analyze** icon  > Select the fields *Request Start Time* and *Time Taken* in the Analyze dialog box > Click **OK**.

This would automatically change the query to:

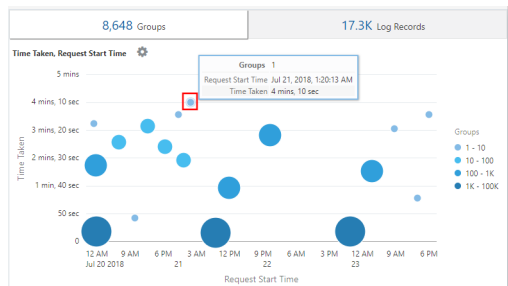
```
'Log Source' = 'EBS Concurrent Request Logs - Enhanced'
| link 'Request ID'
| stats earliest('Event Start Time') as 'Request Start Time',
latest('Event End Time') as 'Request End Time'
| eval 'Time Taken' = 'Request End Time' - 'Request Start Time'
| classify topcount = 300 'Request Start Time', 'Time Taken'
```

`Classify` command takes two fields, clusters the results, and marks the anomalies where applicable. The results are displayed in the bubble chart.

When *Time* is selected for an axis, the bubble chart automatically switches to the

Trend option. To modify the chart options, click the **Chart Options** icon  and change the required parameters.

In the resulting bubble chart, *Request Start Time* is plotted along the x-axis and clusters of *Time Taken* is plotted along the y-axis:



The time is shown in the local time zone. The size of the bubbles indicate the number of requests.

In the above bubble chart, the request duration of more than four minutes is noticed on the 21st July, 2018. Majority of the requests finished in less than two minutes.

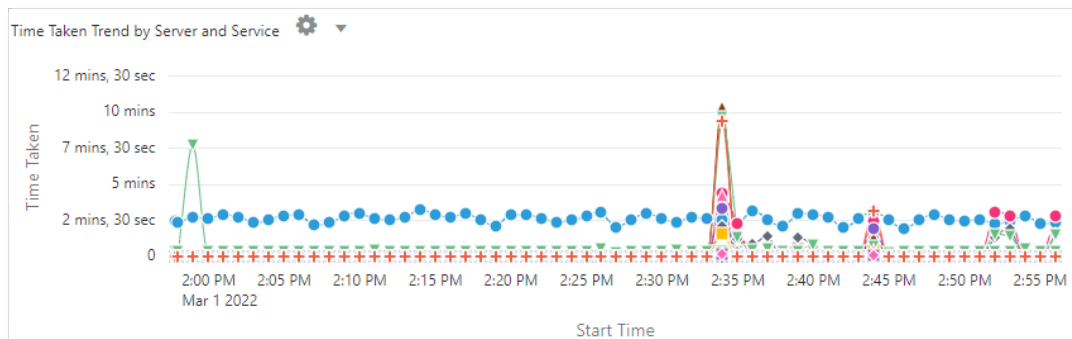
You can click on one or more bubbles to drill down to view the specific requests.

Use `timestats` Command for Time Series Analysis

Use the `timestats` command after the `link` command to generate time series data for analyzing statistical trends over time.

In the following example, the *Application Access Logs* are first grouped by the fields *Time*, *Server*, and *Service* using the `link` command. The output of the `link` command is then used to project the trend for *Time Taken* field, using the `timestats` command. The *Time Taken* field values are plotted for each unique combination of *Server* and *Service*.

```
'Log Source' = 'Application Access Logs'  
| eval 'Duration (sec)' = unit(Duration, second)  
| link Time, Server, Service  
| timestats name = 'Time Taken Trend by Server and Service'  
    avg('Duration (sec)') as 'Time Taken' by Server, Service
```



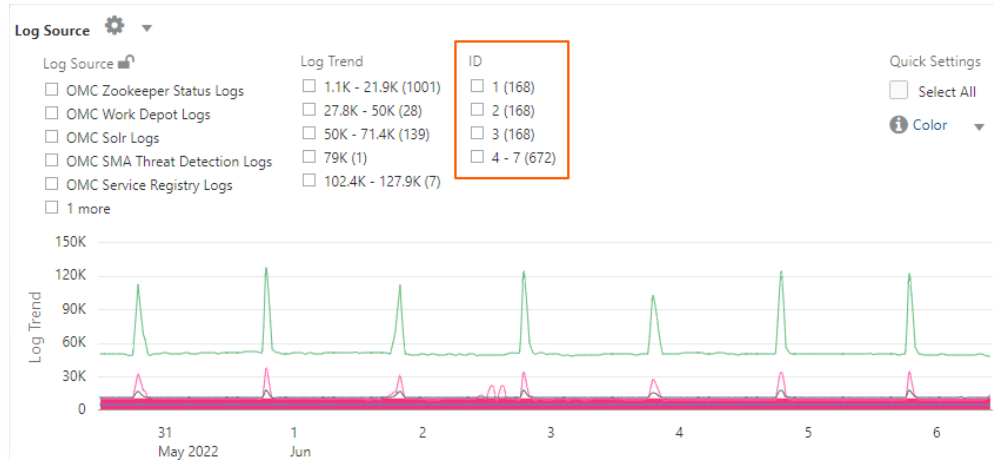
In the above example, the X-axis shows the values from the *Start Time* field. The Y-axis shows the time taken using the average of the numeric field *Duration* in seconds. Each series in the chart represents a unique combination of *Server* and *Service*.

Cluster Similar Time Series

Use the `timecluster` command after `link` to cluster similar time series together. See [Timecluster Command in Using Oracle Log Analytics Search](#).

The following query clusters Log Sources that have a similar trend. You can use the *ID* field to view individual clusters.

```
* | link span = 1hour Time, 'Log Source' | timecluster sum(Count) as 'Log  
Trend' by 'Log Source'
```




The above image shows clusters of log sources that have similar trend grouped by ID.

Analyze the Access Logs of Oracle WebLogic Server

Consider the example of a data set consisting of Oracle WebLogic Server Access Logs from the log source `FMW WLS Server Access Logs`. The log records contain data about the access to Oracle WebLogic Server by the users over a specific period of time. These individual log records can be analyzed to get meaningful insight into the usage statistics, the popularity of the URLs, the most active users, and more such data. From the logs, learn to obtain the following results by analyzing the log records with the selection of specific fields for each result:


1. Display the top URLs by Number of Hits
2. Display the anomalies by Number of Hits
3. Display the anomalies by Access Duration
4. Identify the URLs by Upload Size
5. Identify the URLs by Download Size
6. Analyze the correlation between Number of Hits and Download Size
7. Determine the Most Visited Pages
8. Identify the Top Users
9. Identify the Top Users and their Favorite Pages
10. Identify the entry page that drives maximum visits
11. Identify the Entry and Exit path for most users

 **Note:**

- Use the `rename` command to change the name of the field to one that's more relevant for the use-case.
- The `classify` command lets you analyze the groups, and displays the result in the form of a bubble chart. To simply view the result of the execution of a query in the tabular format, remove the `classify` command from the query, and re-run it.
- Click the anomalous bubble in the chart to view the details of the anomalous groups. To return to the original result after investigating the bubble, click the **Undo**  icon.
- When you run the `link` command, the group duration is shown in a readable format in the bubble chart, for example, in *minutes* or *seconds*. However, if you want to run a `where` command after the `link` command to look for transactions that took more than the specified number of seconds (say, 200 seconds), then the unit that you must use is *milliseconds*.

To retrieve the data set, select a suitable date range, specify the log source, and run the query:

```
'Log Source' = 'FMW WLS Server Access Logs'
```

Select **Link**  from the **Visualize** panel. This'll display the 'FMW WLS Server Access Logs' groups table and the bubble chart.

1. **To display the top URLs by Number of Hits**, group the log records by the value of the URL in the log record, obtain the total count for the URL in each group, rename the default fields in the groups table to suitable values, and display the result in the tabular format. With this analysis, you can determine the URLs that're most used.
 - a. Drag and drop the field **URI** to **Link By**, remove the field **Log Source** from **Link By**, and click the check mark to submit the query.
 - b. After the query is executed, in the command-line, change the names of the fields **Count** to **Number of Hits**, **Start Time** to **First Access**, **End Time** to **Last Access**, and **Group Duration** to **Access Duration**.
 - c. Remove the `classify` command from the command-line, and submit the query.

The query will be as follows:

```
'Log Source' = 'FMW WLS Server Access Logs' | link URI | rename Count  
as 'Number of Hits', 'Start Time' as 'First Access', 'End Time' as  
'Last Access', 'Group Duration' as 'Access Duration'
```

On running the query, you can determine the top URLs by number of hits in the table. The columns are renamed as specified in the rename command.

2. **To display the anomalies by Number of Hits**, group the log records by the value of the URL in the log record, rename the default fields in the groups table to suitable values,

and analyze the groups for the URL's number of hits. With this analysis, you can separate the unusual pattern in accessing the URLs.

Click **Analyze**, select **Number of Hits**, and click **OK**.

The query must change to the following:

```
'Log Source' = 'FMW WLS Server Access Logs' | link URI | rename
Count as 'Number of Hits', 'Start Time' as 'First Access', 'End
Time' as 'Last Access', 'Group Duration' as 'Access Duration' |
classify topcount = 300 'Number of Hits'
```

This query triggers analysis of the 'Number of Hits' column and creates bubbles representing the commonly seen ranges. The majority of the values are treated as the baseline. For example, a large bubble can become the baseline, or a large number of smaller bubbles clustered together can form the baseline. Bubbles that are farthest from the baseline are marked as anomalies.

So, this displays the anomalous URLs grouped into separate bubbles in the bubble chart. To view the percentage of URLs in each range of number of hits, hover the cursor on the bubbles.

3. **To display the anomalies by Access Duration**, group the log records by the value of the URL in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the access duration of the URL. With this analysis, you can separate the unusual pattern in the time spent in accessing the URLs. In continuation to step 2:

Click **Analyze**, select **Access Duration**, and click **OK**.

Access Duration is an indication of the duration for which each URL was accessed. This is computed as the difference between the last timestamp and the first timestamp in the log file for each URL.

4. **To identify the URLs by Upload Size**, group the log records by the value of the URL in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the size of the data uploaded. With this analysis, you can identify the URLs that have unusual size of the data uploaded. In continuation to step 3:

- a. Drag and drop the field **Content Size In** to **Display Fields**.
- b. Rename the field **Content Size In** to **Bytes Uploaded** by altering the query on the command-line, and run the query.
- c. Click **Analyze**, select **Bytes Uploaded**, and click **OK**.

The query will be as follows:

```
'Log Source' = 'FMW WLS Server Access Logs' | link URI | stats
avg('Content Size In') as 'Bytes Uploaded' | rename Count as
'Number of Hits', 'Start Time' as 'First Access', 'End Time' as
'Last Access', 'Group Duration' as 'Access Duration' | classify
topcount = 300 'Bytes Uploaded'
```

The Analyze chart displays the groups of URLs by the bytes uploaded.

- d. To correlate the **Bytes Uploaded** data across the time range, you can selectively hide or show charts in the Histogram Chart Options. Explore the other visualization options besides the bar chart.

5. **To identify the URLs by Download Size**, group the log records by the value of the URL in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the size of the data downloaded. With this analysis, you can identify the URLs that have unusual size of the data downloaded. In continuation to step 4:

- a. Drag and drop the field **Content Size Out** to **Display Fields** and remove **Content Size In** from **Display Fields**.
- b. Rename the field **Content Size Out** to **Download Size** by altering the query on the command-line, and run the query.
- c. Click **Analyze**, select **Download Size**, and click **OK**.

The query will be as follows:

```
'Log Source' = 'FMW WLS Server Access Logs' | link URI | stats  
avg('Content Size Out') as 'Download Size' | rename Count as 'Number  
of Hits', 'Start Time' as 'First Access', 'End Time' as 'Last  
Access', 'Group Duration' as 'Access Duration' | classify topcount =  
300 'Download Size'
```

The Analyze chart displays the groups of URLs by the download size.

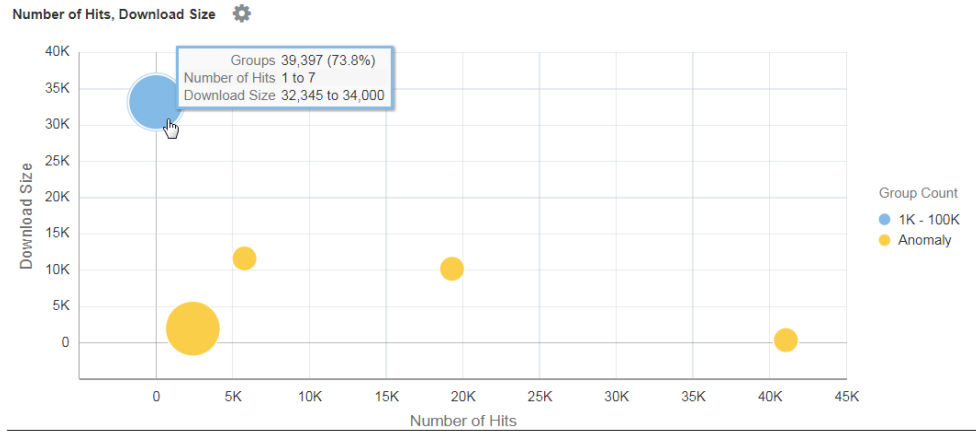
- d. To correlate the **Download Size** data across the time range, you can selectively hide or show charts in the Histogram Chart Options. Explore the other visualization options besides the bar chart.
6. **To analyze the correlation between Number of Hits and Download Size**, group the log records by the value of the URL in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the size of the data downloaded and the number of hits. With this analysis, you can identify the URLs that have unusual patterns of size of data downloaded and number of hits. In continuation to step 5:

- a. Click **Analyze**, select the fields **Number of Hits**, **Download Size**, and click **OK**.
- b. Remove `topcount=300` from the query to see all the bubbles, and run the query.

The query will be as follows:

```
'Log Source' = 'FMW WLS Server Access Logs' | link URI | stats  
avg('Content Size Out') as 'Download Size' | rename Count as 'Number  
of Hits', 'Start Time' as 'First Access', 'End Time' as 'Last  
Access', 'Group Duration' as 'Access Duration' | classify 'Download  
Size', 'Number of Hits'
```

In the bubble chart, the field **Number of Hits** is plotted along the x-axis and **Download Size** along the y-axis.



The bubbles can be interpreted as follows:

- 73.8% of the URLs were accessed one to seven times.
 - Average download size for the 73.8% of URLs is between 32,345 to 34,000. This tight range implies that a large number of URLs have very uniform behavior with reference to the download size.
 - Since 73.8% is the large majority, the rest of the points are marked as anomalies.
 - With real data, it is common for the system to group .css, .js and image files separately from other URLs because they tend to have different download behaviors.
7. **To determine the Most Visited Pages**, group the log records by the value of the URL in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the number of unique visitors. With this analysis, you can identify the URLs that're most visited by the unique visitors. In continuation to step 6:
- Drag and drop the field **User Name** to **Display Fields**.
 - Click the down arrow next to the field name, change the function from **Unique** to **Distinct Count**. See the other functions you can select for a numeric field:
 - Rename the field **User Name** to **Number of Unique Users**, remove the `classify` command by altering the query on the command-line, and run the query. The query will be as follows:

```
'Log Source' = 'FMW WLS Server Access Logs' | link URI | stats
avg('Content Size In') as 'Bytes Uploaded', avg('Content Size
Out') as 'Download Size', distinctcount('User Name') as 'Number
of Unique Users' | rename Count as 'Number of Hits', 'Start
Time' as 'First Access', 'End Time' as 'Last Access', 'Group
Duration' as 'Access Duration'
```

- Click **Analyze**, select the field **Number of Unique Users**, and click **OK**.

The table lists the URLs and the corresponding number of unique users, helping us to identify the URLs that were most visited by unique users. From the table, you can also determine the number of hits that each URL has.

The analysis shows that more than 99% of the URLs have 0 or 1 unique users. This would be the case for URLs that don't need a login, or are seldom accessed. Drilling down to any of the smaller bubbles will point to the specific pages, how many hits they typically have, and how many unique visitors.

8. **To identify the Top Users**, group the log records by the value of the user name in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the number of hits. With this analysis, you can identify the most active users.
 - a. Edit the command-line to remove all the filters: `'Log Source' = 'FMW WLS Server Access Logs' | link URI`
 - b. Drag and drop the field **User Name** to **Link By**, remove **URI**, and run the query.
 - c. Remove the `classify` command, rename the default fields in the command-line, and run the following query:

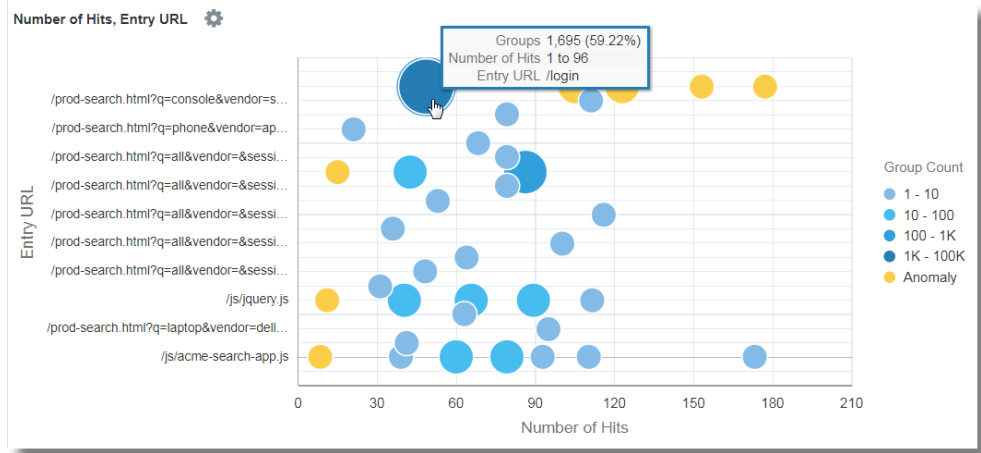
```
'Log Source' = 'FMW WLS Server Access Logs' | link 'User Name' |  
rename Count as 'Number of Hits', 'Start Time' as 'First Access',  
'End Time' as 'Last Access', 'Group Duration' as 'Access Duration'
```

The table is sorted by the number of hits by the user.

- d. To view the user behavior by access, click **Analyze**, select the field **Number of Hits**, and click **OK**.
 - e. Click the anomalies to identify the users who have recorded higher or lower number of hits compared to the other users.
9. **To identify the Top Users and their Favorite Pages**, group the log records by the value of the user name in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the number of unique pages. With this analysis, you can identify the least and most active users, and their favorite pages. In continuation to step 8:
 - a. Drag and drop the field **URI** to **Display Fields**. Change the function from **Unique** to **Distinct Count**.
 - b. Rename the field **URI** to **Number of Unique Pages** by altering the query in the command-line, and run the query.
 - c. Click **Analyze**, select the field **Number of Unique Pages**, and click **OK**.
10. **To identify the entry page that drives maximum visits**, group the log records by the value of the user name in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the values of the entry URLs and number of hits to the URLs. With this analysis, you can identify the pages that the users hit first. In continuation to step 9:
 - a. To get the entry URLs, change the function of the field **URI** from **Distinct Count** to **Earliest**.
 - b. Rename the field **URI** to **Entry URL** by altering the query in the command-line, and run the query.
 - c. Click **Analyze**, select the fields **Number of Hits** and **Entry URL**, select the **topcount** as **20**, and click **OK**.

```
'Log Source' = 'FMW WLS Server Access Logs' | link 'User Name' |  
stats earliest(URI) as 'Entry URL' | rename Count as 'Number of
```

```
Hits', 'Start Time' as 'First Access', 'End Time' as 'Last
Access', 'Group Duration' as 'Access Duration' | classify
topcount = 20 'Number of Hits', 'Entry URL'
```



This displays the first URL used by the users in relation to the number of hits. For example, */login* is the first URL majority of the users use.

11. **To identify the Entry and Exit path for most users**, group the log records by the value of the user name in the log record, rename the default fields in the groups table to suitable values, and analyze the groups for the values of the entry URLs and exit URLs. With this analysis, you can identify

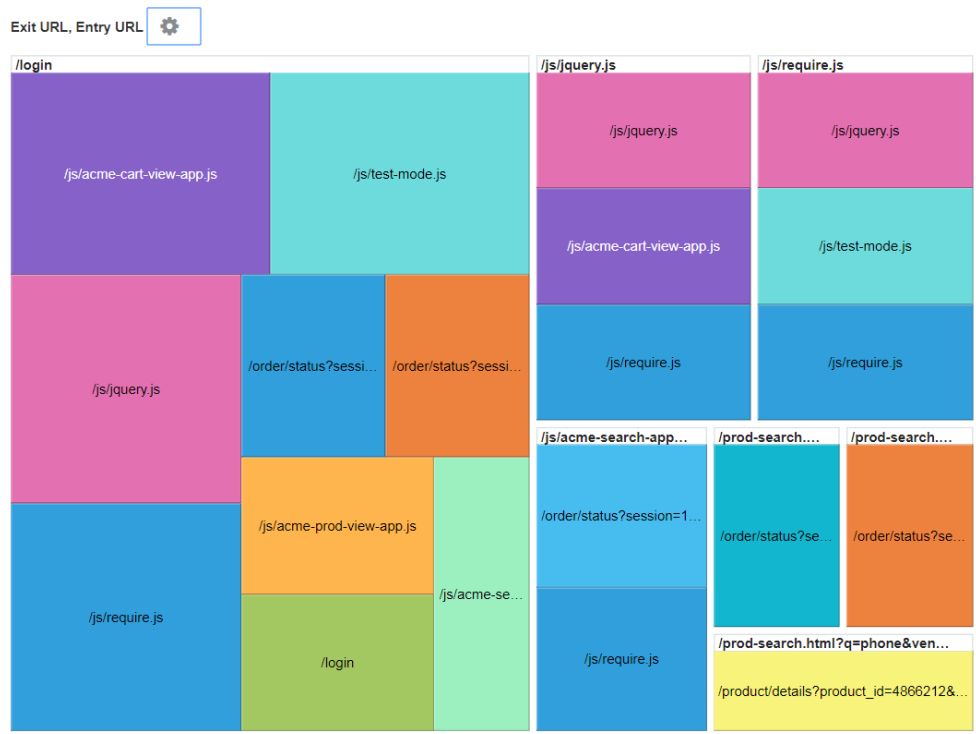
- The most common paths taken by the users to transit through the website
- The most popular product pages from where the users are exiting the website
- The most common exit URLs, like the product checkout pages or the payment gateway
- The unusual exit URLs, and root cause the unexpected exits

In continuation to step 10:

- a. Drag and drop the field **URI** to **Display Fields**.
- b. To get the exit page, change the function of the field **URI** from **Unique** to **Latest**.
- c. Edit the command-line and rename the field **latest(URI)** to **Exit URL** and submit the query.
- d. Click **Analyze**, select the fields **Entry URL** and **Exit URL**, select the **topcount** as **20**, and click **OK**.

```
'Log Source' = 'FMW WLS Server Access Logs' | link 'User Name' |
stats earliest(URI) as 'Entry URL', latest(URI) as 'Exit URL' |
rename Count as 'Number of Hits', 'Start Time' as 'First
Access', 'End Time' as 'Last Access', 'Group Duration' as
'Access Duration' | classify topcount = 20 'Entry URL', 'Exit
URL'
```

- e. Increase the size of the chart by using the Analyze Chart Options.



This tree map shows the relationship between the entry and exit URLs in a site. This would be very useful for the retail sites where the service providers would want to identify the entry URLs that lead the customers to the checkout pages, and the product URLs that're causing users to not proceed to checkout.

Generate Charts with Virtual Fields

To create a new virtual field, you can use the `eval` command in the link feature. The `eval` query on the command-line will generate a line chart for the virtual field and enable tracking it over time.

To create a new virtual field, you can use the `eval` command in the link feature. The `eval` query on the command-line will generate a line chart for the virtual field and enable tracking it over time.

Examples:

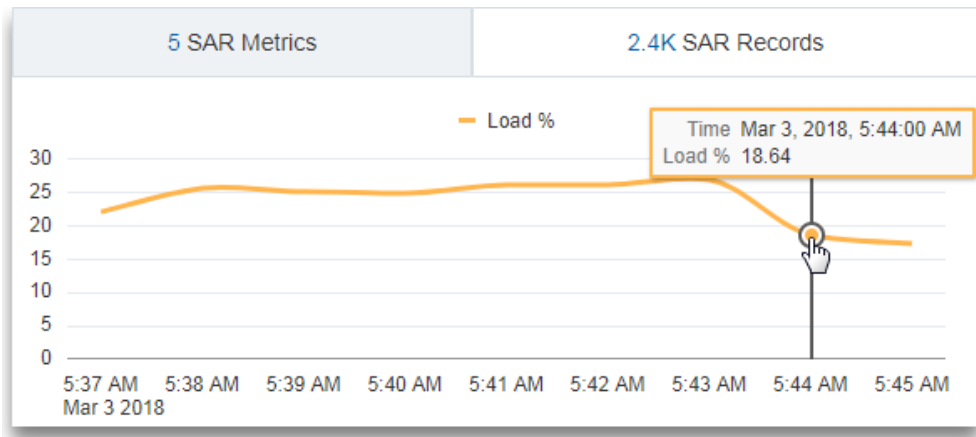
- Consider the scenario where the log records from the log source SAR CPU Logs are grouped by the host name and the CPU. To determine the load experienced by the CPU of the server over time, the `eval` command creates a virtual field `Load %` and generates the line chart.

```
'Log Source' = 'SAR CPU Logs' | rename Instance as CPU | link 'Host Name (Server)', CPU | stats avg('CPU Idle Time (%)') as 'CPU Idle Time (%)' |
eval 'Load %' = 100 - 'CPU Idle Time (%)'
```

To view the line chart:

1. Click the **Histogram** tab.

2. Click the down arrow next to the **Chart options** (⚙️) icon. Click **Hide / Show Charts**. Select **Load %**.
3. Click the down arrow next to the **Chart options** (⚙️) icon. Click **Chart Options**. From the **Chart Type** list, select **Line Without Marker**. Click **Close**.

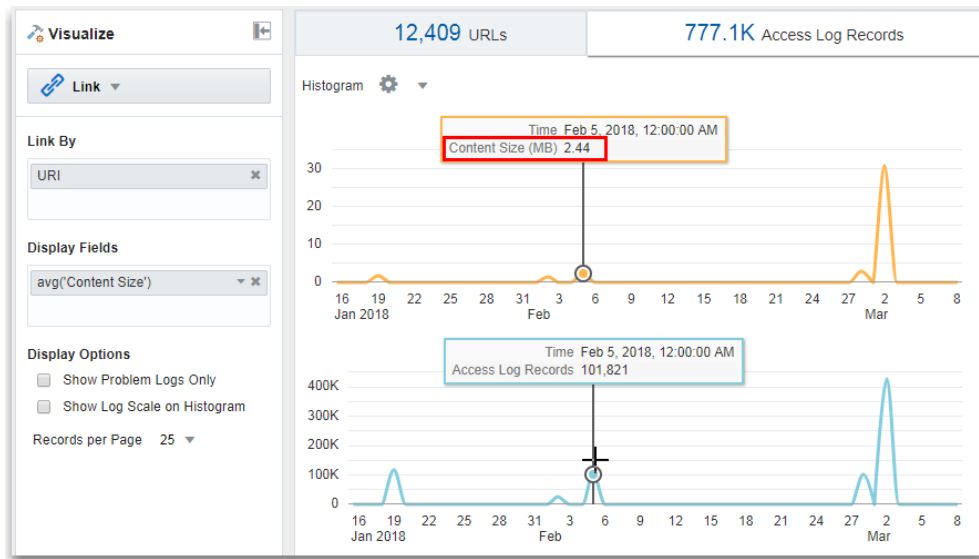


- Consider the scenario where the log records from the log source OMC WLS Server Access Logs are grouped by the URI. To determine the size of the data accessed over time in megabytes, the eval command creates a virtual field Content Size (MB), calculates the content size in megabytes based on the value of the field Content Size, and generates the line chart.

```
'Log Source' = 'WLS Server Access Logs' | link URI | stats
avg('Content Size') as 'Content Size Bytes' | eval 'Content Size
(MB)' = 'Content Size Bytes' / 1024
```

To view the line chart:

1. Click the **Histogram** tab.
2. Click the down arrow next to the **Chart options** (⚙️) icon. Click **Hide / Show Charts**. Select **Content Size (MB)** and **Access Log Records**.
3. Click the down arrow next to the **Chart options** (⚙️) icon. Click **Chart Options**. From the **Chart Type** list, select **Line Without Marker**. Click **Close**.



Link by Using SQL Statement as the Field of Analysis

Link supports **SQL Statement** as a field for analysis. **SQL Statement** contains the SQL that's executed, and is captured by log sources like Database Audit XML Logs and Oracle Unified DB Audit Log Source Stored in Database 12.1.

You can use link 'SQL Statement' to group SQLs and analyze their behavior and identify anomalies.

Example:

Consider the following query that links the log records based on the field **SQL Statement**:

```
'Log Source' in ('Database Audit Logs', 'Database Audit XML Logs')
| rename 'Host Name (Server)' as 'DB Server', 'User Name (Originating)'
as 'OS User', 'User Name' as 'DB User'
| link 'SQL Statement'
| rename Count as 'Number of Runs', 'Start Time' as 'First Run', 'End
Time' as 'Last Run', 'Group Duration' as Age
| addfields [ Object = dual | stats count as 'dual Table Access' ],
[ Object like 'all_%' | stats count as 'ALL_ Table Access' ],
[ Object like 'dba_%' | stats count as 'DBA_ Table Access' ],
[ Object like 'user_%' | stats count as 'USER_ Table Access' ],
[ Object like 'v$%' | stats count as 'VDollar Table Access' ],
[ Object = null | stats count as 'No Table Access' ],
[ Action = '2' | stats count as 'Insert Count' ],
[ Action = '3' | stats count as 'Select Count' ],
[ Action = '6' | stats count as 'Update Count' ],
[ Action = '7' | stats count as 'Delete Count' ],
[ Type = '8' | stats count as 'Connect Count' ],
[ 'Status Code' = 1 | stats count as Failures ]
| eval 'Object Type' = if('dual Table Access' > 0, Dual,
'ALL_ Table Access' > 0, System,
'DBA_ Table Access' > 0, System,
'USER_ Table Access' > 0, System,
```

```

        'VDollar Table Access' > 0, System,
        'No Table Access' > 0, 'No Table', Other)
| eval 'SQL Type' = if('Insert Count' > 0, Insert,
        'Select Count' > 0, Select,
        'Update Count' > 0, Update,
        'Delete Count' > 0, Delete,
        'Connect Count' > 0, Connect, Other)
| stats distinctcount(Object) as Objects, distinctcount('Database
ID') as 'Number of DBs',
        distinctcount(Session) as 'Number of Sessions'
| fields -'dual Table Access', -'No Table Access', -'ALL_ Table
Access',
        -'USER_ Table Access', -'DBA_ Table Access', -'VDollar Table
Access', -'Insert Count',
        -'Select Count', -'Update Count', -'Delete Count', -'Connect
Count', -'SQL Type', -'Object Type'
| classify Age
| classify 'Number of Sessions'
| classify 'Number of DBs'
| classify 'Number of Runs', 'Object Type'
| classify 'Object Type', 'SQL Type'

```

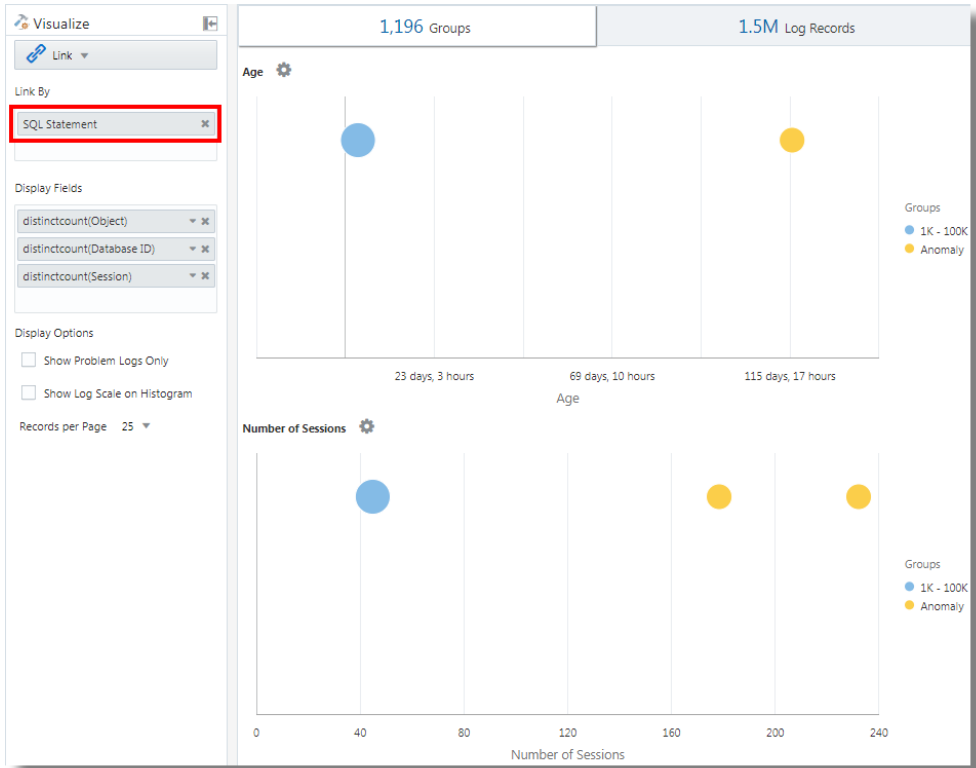
 **Note:**

`addfields` is a function available with link visualization to add virtual fields to the query. It takes a query and pipes the output to a `stats` command. The resulting virtual field is available in the table as well as in the time series chart.

For the syntax and other details of the `addfields` command, see `Addfields Command` in *Using Oracle Log Analytics Search*.

By executing the above query, the following results can be observed:

- Based on the `classify` command, the bubble charts for Age, Number of Sessions, Number of DBs, Number of Runs, Object Type, and Object Type, SQL Type are generated.





In the bubble charts, the log records are grouped based on the number of SQLs that fall under each set of parameters. The `Object Type` and `SQL Type` parameters are determined using the `eval` command in the query.

- The **Line with Area** histogram charts illustrate the occurrence of fields like `dual` Table Access, `No Table Access`, `ALL_ Table Access`, `USER_ Table Access`, `DBA_ Table Access`, `VDollar Table Access`, `Insert Count`, `Select Count`, `Update Count`, `Delete Count`, `Connect Count`, and Log Records plotted against time.
 1. In the histogram chart tab, click the down arrow next to the **Chart options** (⚙️) icon.
 2. Select to show the charts of all the fields.
 3. Under **Chart Type**, select **Line With Area**.
 4. Adjust the width to display two charts per line.



- The **Groups Table** lists the groups identified by link based on the **SQL Statement** field. You can observe that for each SQL, the table lists the number of time that the SQL was run, the start time, the end time, and the group duration. Click on each group and view the log records for more details. You can also view the groups in the cluster visualization for further analysis.

SQL Statement	Number of Runs	First Run	Last Run	Age
select TENANT_ID, OBJ_PATH, MESSAGE_TIME, TOPIC_ID, STATUS from EMS_LOGAN_HDFS_TENANT_SA where LP_ID is null and MESSAGE_TIME >= :1 and TENANT_ID not in (select TENANT_ID from EMS_LOGAN_DEPRI_TENANT where PAUSED > 0 and DATA_TYPE = :2) order by MESSAGE_TIME asc	163,616	May 16, 2018 8:11:36:08 AM	May 17, 2018 10:00:15:3 AM	22 hour 1.25 mins
select TENANT_ID, OBJ_PATH, MESSAGE_TIME, TOPIC_ID, STATUS from EMS_LOGAN_HDFS_TENANT where LP_ID is null and MESSAGE_TIME >= :1 and TENANT_ID not in (select TENANT_ID from EMS_LOGAN_DEPRI_TENANT where PAUSED > 0 and DATA_TYPE = :2) order by MESSAGE_TIME asc	163,550	May 16, 2018 8:11:36:08 AM	May 17, 2018 10:00:15:3 AM	22 hour 1.25 mins
COMMIT	117,171	Jan 19, 2018 8:10:32:23 PM	May 17, 2018 10:00:14:44 AM	117 days, 11 hour 1 s
begin dbms_application_info.set_module(1, :2) end;	109,606	Jan 19, 2018 8:11:49:18 PM	May 17, 2018 10:00:14:44 AM	117 days, 10 hour 1 s
select LP_ID from EMS_LOGAN_LP_HEARTBEAT	81,984	May 16, 2018 8:11:36:07 AM	May 17, 2018 10:00:15:3 AM	22 hour 1.25 mins
select count(distinct OBJ_PATH), LP_ID, TENANT_ID, TOPIC_ID from EMS_LOGAN_HDFS_TENANT_SA where LP_ID is not null and STATUS in (:1, :2) and MESSAGE_TIME >= :3 group by LP_ID, TENANT_ID, TOPIC_ID	81,808	May 16, 2018 8:11:36:08 AM	May 17, 2018 10:00:15:3 AM	22 hour 1.25 mins
select count(distinct OBJ_PATH), LP_ID, TENANT_ID, TOPIC_ID from EMS_LOGAN_HDFS_TENANT where LP_ID is not null and STATUS in (:1, :2) and MESSAGE_TIME >= :3 group by LP_ID, TENANT_ID, TOPIC_ID	81,808	May 16, 2018 8:11:36:07 AM	May 17, 2018 10:00:15:3 AM	22 hour 1.25 mins

Analyze the Time Taken Between Steps in a Transaction

The link feature gives you the ability to analyze user sessions, extract the various time parameters by grouping, and deduce data about the transaction time to help you in getting business insights.

Consider this unordered data set taken from an Access Log file. The following fields indicate the information about a user session and the actions performed by the user:

Time	Session ID	Action
T2	1	Login
T1	5	Login
T6	1	addtocart
T3	1	productlisting
T4	1	purchase
T9	1	purchase
T7	5	addtocart
T5	1	addtocart
T8	5	purchase

The actions like Login, addtocart, productlisting, and purchase are recorded in a random order T1 through T9, and have occurred in two sessions with session ID 1 and 5.

To, perform similar time analysis of your Access Logs, extract the `Session ID` from the logs into a field. Extract the intermediate steps of the session from the Access Logs by applying a regular expression to obtain the `URL` from the logs.

In a generic context, the sessions in this example represent any user transactions, and the actions represent the intermediate steps performed by the user to complete a transaction.

To analyze this unordered data and to extract the required information, the following example query can be run:

```
'Upload Name' = logadmin
| link 'Session ID'
| rename 'Group Duration' as 'Session Duration'
| addfields
  [ Action = addtocart | stats earliest(Time) as 'First Add To Cart Time' ],
  [ Action = purchase | stats latest(Time) as 'Last Purchase Time' ]
| eval 'Time Taken for Purchase (Secs)' = ('Last Purchase Time' - 'First Add
To Cart Time') / 1000
| fields -'First Add To Cart Time',
        -'Last Purchase Time'
| classify 'Time Taken for Purchase (Secs)'
```

- `link 'Session ID'` groups the Access Logs records by the Session ID, creating two groups:

Time	Session ID	Action
T2	1	Login
T6	1	addtocart
T3	1	productlisting
T4	1	purchase
T5	1	addtocart
T9	1	purchase
T1	5	Login
T7	5	addtocart
T8	5	purchase

- `addfields` is run against each of these groups. The first `addfields` picks up the records where `Action = addtocart`. The result of this query is as below for both the groups:

Time	Session ID	Action
T6	1	addtocart
T5	1	addtocart
T7	5	addtocart

- `stats earliest(Time)` sorts the above result by time, for each group:

Time	Session ID	Action
T5	1	addtocart
T6	1	addtocart
T7	5	addtocart

- Then the specified field, which is `Time`, is picked up from the **first** record:

```
'First Add To Cart Time' = T5 for Group = 1
'First Add To Cart Time' = T7 for Group = 5
```

- The second `addfields` runs on `Action = purchase`, extracting the following records:

Time	Session ID	Action
T4	1	purchase
T9	1	purchase
T8	5	purchase

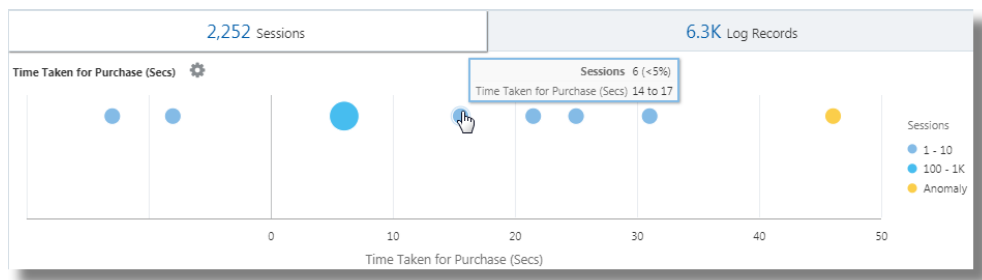
- `latest(Time)` also sorts the above records by `Time`:

Time	Session ID	Action
T4	1	purchase
T9	1	purchase
T8	5	purchase

- `latest(Time)` picks up the **last** record and extract the specified field, which is `Time`:

```
'Last Purchase Time' = T9 for Group = 1
'Last Purchase Time' = T8 for Group = 5
```

- At this point, both the groups have the values for `First Add to Cart Time` and `Last Purchase Time` set. These are timestamps. `eval` subtracts one from another to get the elapsed time.
- In effect, you can get the time taken from *Adding to the Cart* to the *Purchase* step for each session. This can now be used in `classify` to analyze the variance of this *Elapsed Time* across sessions.




For the syntax and other details of the `addfields` command, see [Addfields Command in Using Oracle Log Analytics Search](#).

Generate Charts for Multiple Fields and their Values

You can use the `addfields` command in the query to specify multiple fields to generate separate charts. Now, you can also use the histogram **Add Chart** option in the UI to perform the same operation as the `addfields` command.


Typically, you would want to compare the charts of a single field with various values, for example, values of the field *Severity* like *Error*, *Critical*, *Alert*, and *Warning*. The Add Chart option allows you to generate multiple charts to compare side-by-side by specifying the field and its values in the dialog box.


Alternatively, you can type and update the query with the command. The Add Chart option enables you to perform the operation faster than composing the query with `addfields` command.

1. From the link UI, go to **Log Records** tab > click the **Add Chart**  option, to automatically update the query with the `addfields` command.

The **Add Charts** dialog box opens.

2. Next to **Subquery**, select the field from the drop-down menu, for example, *Severity*.
Select the relevant operator.

Click the edit icon  to select the value of the field from the available options, for example, *alert*.

3. Next to **Stats**, select the **Function** to perform on the field and the **Function Field** from the drop down menu, for example, *count*.
4. Click **Add Chart** to view the resulting query. Click the edit icon  to edit the query.

Add Charts ✕

To add a chart, specify a field and value for the subquery (e.g., action = get), then click Add Chart.

[Show Tips](#)

Subquery: Field: Severity, Operator: =, Value: alert ✎

Stats: Function: count, Function Field: Optional, Chart Name: 'Alert Count'

Add Chart ➔

Charts	Edit	Delete
Severity = alert stats count as 'Alert Count'	✎	✖
Severity = error stats count as 'Error Count'	✎	✖
Severity = critical stats count as 'Critical Count'	✎	✖

Page 1 of 1 (1-3 of 3 items) ⏪ < 1 > ⏩

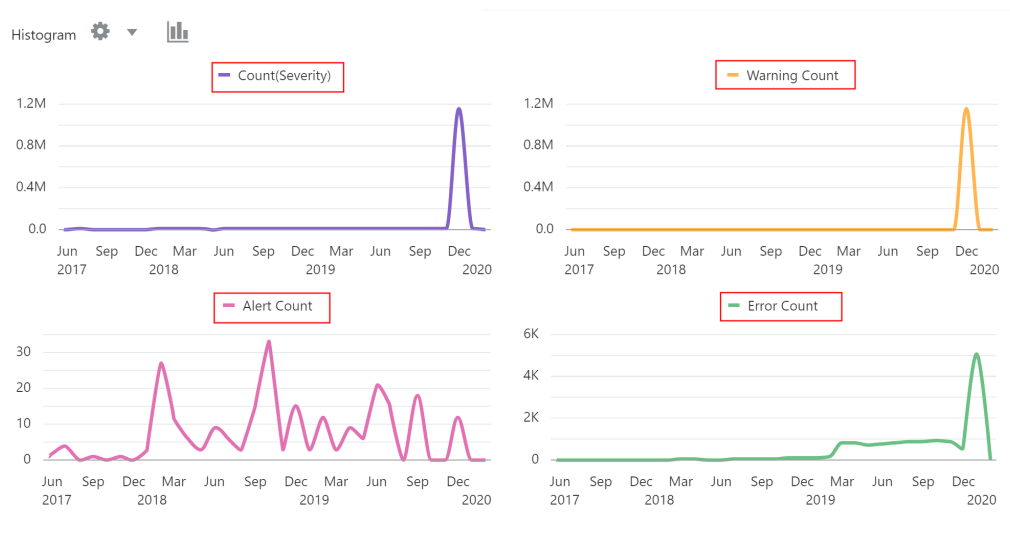
OK Cancel ➔

5. Repeat steps 2 through 4 to add more charts, for example, to generate charts for the values error, critical, and warning of the field Severity.

Click **OK**.

6. Click the down arrow next to the **Chart options** (⚙️) icon and ensure that new charts that you've generated are included and selected in the **Hide/Show** option. You can further select the type of chart and size from the chart options, for example, Line without marker. See [Histogram Chart Options](#).

You can now see the customized charts of select fields and their select values in the **Log Records** tab, and compare them visually.



Second Level Aggregation Using Eventstats Command in Link

Link is used to group the log records using one or more unique keys. For example, you can group all the log records belonging to a transaction using the unique *transaction ID*. Statistics can be generated on each group using the `stats` command. `eventstats` is a new command that can further aggregate these statistics. The following examples illustrate the use cases for `eventstats`.

Consider the following **Access Logs Dataset** throughout the examples:

```
1-Jan-2020 10:00:00 PST, chicago_dc1 /index.html 100
1-Jan-2020 10:00:00 PST, chicago_dc1 /index.html 100
1-Jan-2020 10:00:00 PST, chicago_dc1 /index.html 50
1-Jan-2020 10:00:00 PST, chicago_dc1 /index.html 50
1-Jan-2020 10:00:00 PST, chicago_dc2 /index.html 200
1-Jan-2020 10:00:00 PST, chicago_dc2 /index.html 200
1-Jan-2020 10:00:00 PST, austin_dc7 /report/download 5000
1-Jan-2020 10:00:00 PST, austin_dc7 /users/auth 50
1-Jan-2020 10:00:00 PST, amsterdam_dc1 /index.html 350
1-Jan-2020 10:00:00 PST, amsterdam_dc1 /report/download 1024
```

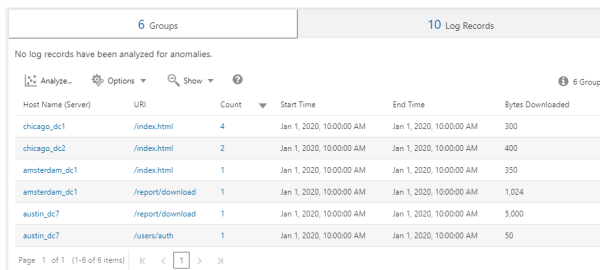
The dataset has these fields:

- **Time:** For example, *1-Jan-2020 10:00:00 PST*.
- **Host Name (Server):** The host that served this request, for example, *chicago_dc1*.
- **URI:** The URL of the request, for example, */index.html*.
- **Content Size Out:** The number of bytes downloaded, for example, *100*.

Simple Grouping:

```
* | link 'Host Name (Server)', URI
  | stats sum('Content Size Out') as 'Bytes Downloaded'
```

The above query groups the log records using the distinct combination of **Host Name (Server)** and **URI** fields. The **Content Size Out** field of each record is then summed up per group into the new field **Bytes Downloaded**.

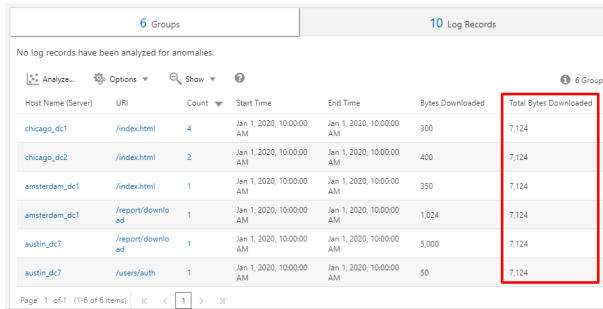


Host Name (Server)	URI	Count	Start Time	End Time	Bytes Downloaded
chicago_dc1	/index.html	4	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	300
chicago_dc2	/index.html	2	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	400
amsterdam_dc1	/index.html	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	350
amsterdam_dc1	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	1,024
austin_dc7	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	5,000
austin_dc7	/users/auth	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	50

Global Sum Using Eventstats

The bytes downloaded in the previous example is for each server and URL combination. A simple use case of `eventstats` is to compute the total data downloaded across all the servers and URLs:

```
* | link 'Host Name (Server)', URI
  | stats sum('Content Size Out') as 'Bytes Downloaded'
  | eventstats sum('Bytes Downloaded') as 'Total Bytes Downloaded'
```



The screenshot shows a log analysis interface with a table of log records. The table has columns for Host Name (Server), URI, Count, Start Time, End Time, Bytes Downloaded, and Total Bytes Downloaded. The 'Total Bytes Downloaded' column is highlighted with a red box. The data in the table is as follows:

Host Name (Server)	URI	Count	Start Time	End Time	Bytes Downloaded	Total Bytes Downloaded
chicago_dc1	/index.html	4	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	300	7,124
chicago_dc2	/index.html	2	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	400	7,124
amsterdam_dc1	/index.html	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	350	7,124
amsterdam_dc1	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	1,024	7,124
austin_dc7	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	5,000	7,124
austin_dc7	/users/auth	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	50	7,124

In the above example, `eventstats` aggregates values for each group to produce a single global roll up. This can now be passed to `classify` or `eval`, as well as used in the `where` clause.

Multiple eventstats Commands:

Multiple `eventstats` can be grouped together or chained as in the following example:

```
.. | eventstats sum('Content Size In') as 'Bytes Uploaded',
  sum('Content Size Out') as 'Bytes Downloaded'
  | eventstats avg('Duration') as 'Global Average Duration'
```

Grouping Using Eventstats

The command `eventstats` also has a *group by* mode. Consider the following query:

```
* | link 'Host Name (Server)', URI
  | stats sum('Content Size Out') as 'Bytes Downloaded'
  | eventstats sum('Bytes Downloaded') as 'Total Bytes Downloaded' by
  URI
```

Instead of computing a single value, `eventstats` now computes one value per unique URI:

Host Name (Server)	URI	Count	Start Time	End Time	Bytes Downloaded	Total Bytes Downloaded
chicago_dc1	/index.html	4	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	300	1,050
chicago_dc2	/index.html	2	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	400	1,050
amsterdam_dc1	/index.html	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	350	1,050
amsterdam_dc1	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	1,024	6,024
austin_dc7	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	5,000	6,024
austin_dc7	/users/auth	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	50	50

The sum is produced by first getting the distinct URIs and then performing the aggregation:

```
index.html      -> 300 + 400 + 350 = 1050
/report/download -> 5000 + 1024      = 6024
/users/auth     -> 50                = 50
```

Eventstats with Eval

The command `eventstats` can also operate on a field produced by an `eval` command. For example, instead of the URL, we can produce the totals against the data center:

```
* | link 'Host Name (Server)', URI
  | stats sum('Content Size Out') as 'Bytes Downloaded'
  | eval offset = indexof('Host Name (Server)', _)
  | eval Datacenter = substr('Host Name (Server)', 0, offset)
  | eventstats sum('Bytes Downloaded') as 'Total Bytes Downloaded' by
Datacenter
  | fields -offset
```

Host Name (Server)	URI	Count	Start Time	End Time	Bytes Downloaded	Datacenter	Total Bytes Downloaded
chicago_dc1	/index.html	4	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	300	chicago	700
chicago_dc2	/index.html	2	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	400	chicago	700
amsterdam_dc1	/index.html	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	350	amsterdam	1,374
amsterdam_dc1	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	1,024	amsterdam	1,374
austin_dc7	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	5,000	austin	5,050
austin_dc7	/users/auth	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	50	austin	5,050

The `sum` function is executed after the grouping by substrings:

```
chicago_dc1 = 300
chicago_dc2 = 400
-> chicago = 300+400 = 700

amsterdam_dc1 = 350
amsterdam_dc1 = 1024
-> amsterdam = 350 + 1024 = 1374
```

```

austin_dc7 = 5000
austin_dc7 = 50
-> austin = 5000 + 50 = 5050

```

Grouping can be performed by using one or more properties. The properties are the group keys, or string values produced by `stats` or `eval`.

Compute Percentages for Group Comparison

A very important application for `eventstats` command is to produce a global value, and identify the high percentage or low percentage contribution from various groups:

```

* | link 'Host Name (Server)', URI
  | stats sum('Content Size Out') as 'Bytes Downloaded'
  | eval offset = indexof('Host Name (Server)', '_')
  | eval Datacenter = substr('Host Name (Server)', 0, offset)
  | eventstats sum('Bytes Downloaded') as 'Total Bytes Downloaded' by
Datacenter
  | eval 'Download Contribution %' = 100 / ('Total Bytes
Downloaded' / 'Bytes Downloaded')
  | fields -offset

```

Host Name (Server)	URI	Country	Start Time	End Time	Bytes Downloaded	Datacenter	Total Bytes Downloaded	Download Contribution %
chicago_dc1	/index.html	4	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	300	chicago	700	42.857
chicago_dc2	/index.html	2	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	400	chicago	700	57.143
amsterdam_dc1	/index.html	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	350	amsterdam	1,374	25.473
amsterdam_dc1	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	1,024	amsterdam	1,374	74.527
austin_dc7	/report/download	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	5,000	austin	5,050	99.01
austin_dc7	/users/auth	1	Jan 1, 2020, 10:00:00 AM	Jan 1, 2020, 10:00:00 AM	50	austin	5,050	0.99

Download Contribution % is computed using the global value produced by the `eventstats` command and the value per group produced by `stats`:

```

chicago_dc1, index.html          => 100/(700/300)   = 42.857
chicago_dc2, index.html          => 100/(700/400)   = 57.143
amsterdam_dc1, index.html         => 100/(1374/350)  = 25.473
amsterdam_dc1, /report/download  => 100/(1374/1024) = 74.527
austin_dc7, /report/download       => 100/(5050/5000) = 99.01
austin_dc7, /users/auth           => 100/(5050/50)   = 0.99

```

This query allows you to see which URLs cause the highest download traffic compared to the other URLs in the same data center. **Download Contribution %** field can be used to filter the groups using:

- the `where` clause
- `sort` command for ranking
- `classify` command for anomaly detection

Use Link Navigation Functions to Identify Events in a Database

Use Link to create structured data from log records and display the data as an ordered table. Statistical functions can be applied to columns of the table using the `stats` command, to create derived columns. These derived columns can be further aggregated using the `eventstats` command.

Navigation Functions

Navigation functions are useful to fetch values of a specific column from a specific row. They produce different results depending on the preceding sort command.

The following navigation functions can be used with the `eventstats` command in link:

Function	Description
<code>rownum</code>	Create a row number column
<code>first()</code>	Get the first value for the specified field
<code>last()</code>	Get the last value for the specified field
<code>nthval()</code>	Get the column value for the specified row
<code>lag()</code>	Get the column value for the previous row
<code>lead()</code>	Get the column value for the next row

For more information about the functions, see `Eventstats Command` in *Using Oracle Log Analytics Search*.

Get Context for an Event

Oracle Log Analytics provides out-of-the-box labels for the *Database Alert Logs*. The Label *Abnormal Termination* indicates a serious issue causing the database to shutdown. A typical triage involves analyzing the sequence of events that happened before such a shutdown. It is also useful to know the events after a shutdown.

The following sections explain the steps to triage by using some of the `eventstats` functions for *Database Alert Logs*.

Link Events in Database Alert Logs

Run the following query to link the events for a selected database:

```
'Log Source' = 'Database Alert Logs' and Label != null and Entity = MyDB
| rename Entity as Database
| link span = 1minute Time, Database, Label
| sort Database, 'Start Time'
```

This creates a unique row for each **Label** in the **Database**. Since we have included the **Time** column, there would be multiple rows for the same **Label**, if they repeat at different times.

The `sort` command sorts the table by the order of **Label**, with the oldest one at the first row.

Add Row Number

Run the following query to add a number to each row:

```
'Log Source' = 'Database Alert Logs' and Label != null and Entity =
MyDB
| rename Entity as Database
| link span = 1minute Time, Database, Label
| sort Database, 'Start Time'
| eventstats rownum as 'Row Number' by Database
```

If the query had more than one database, then the **Row Number** would reset for each **Database**, due to the `by Database` clause.

Identify the Row with Database Crash Event

The Label *Abnormal Termination* indicates that the database crashed. Identify such rows with the following query:

```
'Log Source' = 'Database Alert Logs' and Label != null and Entity =
MyDB
| rename Entity as Database
| link span = 1minute Time, Database, Label
| sort Database, 'Start Time'
| eventstats rownum as 'Row Number' by Database
| addfields
  [ * | where Label = 'Abnormal Termination'
    | eventstats last('Row Number') as 'Crash Row'
  ]
```

`addfields` is used to identify a subset of the log records. In this case, `addfields` searches through several rows of the table. The matching rows are passed to `eventstats`, and `last('Row Number')` picks up the last matching row's **Row Number**. This is now populated as a new field **Crash Row**. Note that **Crash Row** will have a value only for those rows that match the condition specified in `addfields`.

Crash Row is populated only for specific rows. Use another `eventstats` to populate all the rows with the value:

```
'Log Source' = 'Database Alert Logs' and Label != null and Entity =
MyDB
| rename Entity as Database
| link span = 1minute Time, Database, Label
| sort Database, 'Start Time'
| eventstats rownum as 'Row Number' by Database
| addfields
  [ * | where Label = 'Abnormal Termination'
    | eventstats last('Row Number') as 'Crash Row'
  ]
| eventstats max('Crash Row') as 'Event Row' by Database
```

This creates the column **Event Row** in every row, and contains the row that had the last Database Crash.

Identify Events near Database Crash

The table still has several events, for example, hundreds. To identify few events before the Event Row, and few events after the Event Row, change the query to filter the rows:

```
'Log Source' = 'Database Alert Logs' and Label != null and Entity = MyDB
| rename Entity as Database
| link span = 1minute Time, Database, Label
| sort Database, 'Start Time'
| eventstats rownum as 'Row Number' by Database
| addfields
  [ * | where Label = 'Abnormal Termination'
      | eventstats last('Row Number') as 'Crash Row'
  ]
| eventstats max('Crash Row') as 'Event Row' by Database
| eval 'Start Row' = 'Event Row' - 3
| eval 'End Row' = 'Event Row' + 2
| where 'Row Number' >= 'Start Row' and 'Row Number' <= 'End Row'
```

The table now shows which events happened before **Abnormal Termination**. We can also see the events that happened after **Abnormal Termination**.

Previous and Next Events

`lag()` can be used to get the previous event. An optional row number can be passed to get a specific previous row. `lead()` can similarly be used to get the next row:

```
'Log Source' = 'Database Alert Logs' and Label != null and Entity = MyDB
| rename Entity as Database
| link span = 1minute Time, Database, Label
| sort Database, 'Start Time'
| addfields
  [ *
    | where Label != null
    | eventstats lag(Label) as 'Previous Event',
                 lead(Label) as 'Next Event'
  ]
```

Further, `nthVal()` can get the value from a specific row.

Use the Currency Symbols in Your Log Analysis

You can use the `unit` function in `eval` command to mark a field as containing currency. You can then use that field value in your analysis and display corresponding currency symbol in the visualizations and groups table.

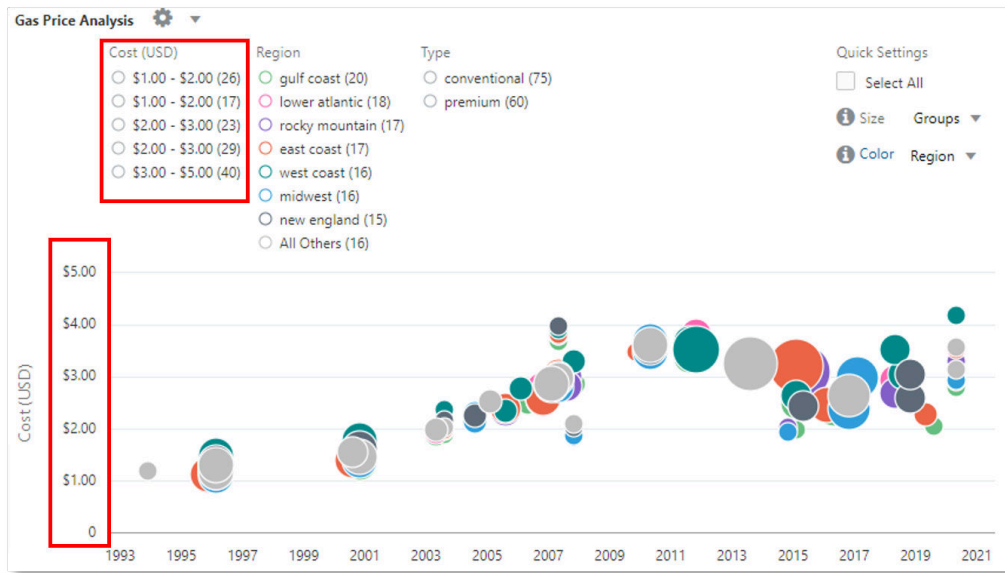
You can first specify the currency unit using the format defined in Eval Command. After that, the link table and charts will display the right currency symbols.

In the following example, the value of the field *Price* is used for calculating the values of the new fields *Price (USD)*, *Price (GBP)*, *Price (JPY)*, *Price (CNY)*, and *Price (INR)* and marking

them as containing currency. The same new fields are used for analysis in obtaining the region-wise average price of gasoline over a period of several years.

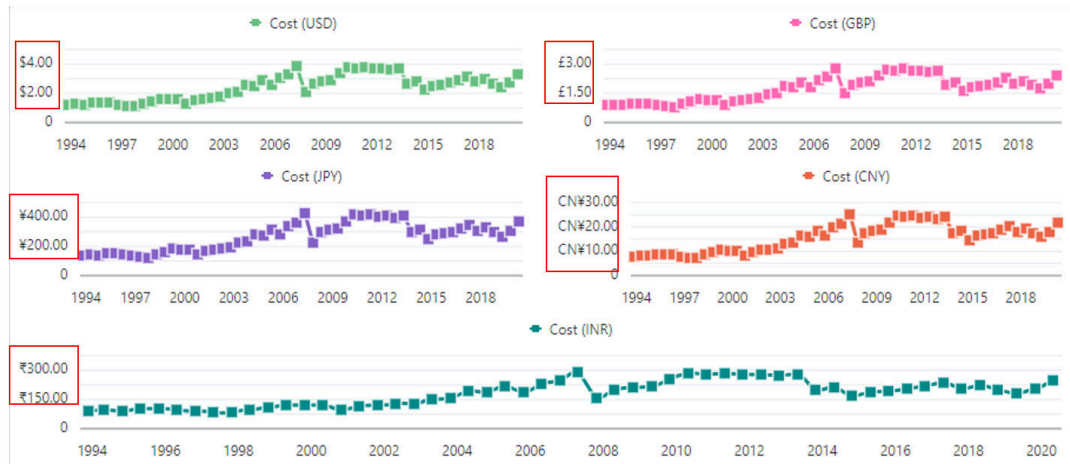
```
'Log Source' = 'Gasoline Prices'
| eval 'Price (USD)' = unit(Price, currency_usd)
| eval 'Price (GBP)' = unit(Price * 0.72, currency_gbp)
| eval 'Price (JPY)' = unit(Price * 110.6, currency_jpy)
| eval 'Price (CNY)' = unit(Price * 6.47, currency_cny)
| eval 'Price (INR)' = unit(Price * 74.79, currency_inr)
| link Time, Type, Region
| stats avg('Price (USD)') as 'Cost (USD)',
        avg('Price (GBP)') as 'Cost (GBP)',
        avg('Price (JPY)') as 'Cost (JPY)',
        avg('Price (CNY)') as 'Cost (CNY)',
        avg('Price (INR)') as 'Cost (INR)'
| classify 'Start Time', 'Cost (USD)', Region, Type as 'Gas Price
Analysis'
```

In the following image, the groups are identified based on region, time and type of gasoline. The average price band of gasoline is used to plot the bubbles along y-axis.



In the following image, the groups table shows the average price of gasoline in various currencies. The charts show the variation of the cost across several years for each currency value.

Type	Region	Count	Start Time	End Time	Cost (USD)	Cost (GBP)	Cost (JPY)	Cost (CNY)	Cost (INR)
conventional	central atlantic	27	May 2, 1999, 5:00:00 PM	Oct 31, 1999, 4:00:00 PM	\$1.25	£0.90	¥137.80	CN¥8.06	₹93.18
conventional	east coast	27	May 2, 1999, 5:00:00 PM	Oct 31, 1999, 4:00:00 PM	\$1.18	£0.85	¥130.57	CN¥7.64	₹88.30
conventional	gulf coast	27	May 2, 1999, 5:00:00 PM	Oct 31, 1999, 4:00:00 PM	\$1.17	£0.84	¥129.18	CN¥7.56	₹87.35



Parse Log Records with Multiple Timestamps

Some log records can have multiple timestamps like the log entry time, the start time, and end time of a process or transaction that you may want to capture into their own fields. After capturing the start and end time into their own fields, you can use the `eval` command to perform date manipulation on such fields, for example, to get the duration between the two times. See `Eval Command` in *Using Oracle Log Analytics Search*.

Oracle Log Analytics has several out-of-the-box fields that let you store time and date information that are not the actual log entry time:

- Event End Time
- Event Start Time
- Event Generation Time
- First Event Time
- Contact End Time
- Contract Start Time
- Alert Raised Time
- Collection Time
- Detected Time

The data that is stored into these fields must be in the ISO-8601 format:

```
2018-07-04T23:43:34.000Z
```

While creating the parser, you can use the `{TIMEDATE}` macro only once to express the log entry time. For the additional time fields, you must extract the data using one of the methods below depending on your use case.

Case 1: Your log already has the time and date information in ISO-8601 format

If your log already has additional time and date information in ISO-8601 format, then you can extract them as strings in the base parser of Extended Field Definition.

Consider the following example log:

```
July 4, 2018 23:43:12 Server1 ModuleA Transaction completed.  
Start=2018-07-04T23:45:34.000Z, End=2018-07-04T23:46:39.000Z
```

The log contains time and date information for log entry, start time, and end time.

1. To obtain the log entry time, create the base parser. See [Create a Parser](#).

```
{TIMEDATE}\s(\S+)\s(\S+)\s(.*)
```

2. Open the **Create Log Source** dialog box. See [Create a Log Source](#).
3. Select the base parser that you created in step 1.
4. Provide the file path for the example log.
5. In the **Extended Fields** tab, add the extended field definitions to the log source to extract the time and date fields:
 - From **Message** field: Start={Event Start Time:\S+}
 - From **Message** field: End={Event End Time:\S+}
6. Save the new log source that you created.

Event End Time	1530728723000
Event Start Time	1530725123000

You'll now notice that the two fields **Event Start Time** and **Event End Time** are populated with the values from the log. In the Log Explorer, you can see the times as milliseconds since epoch.

Case 2: Your log does not have the time and date information in ISO-8601 format

If the additional time fields that you want to extract are not in the ISO-8601 format, then you must follow these steps for parsing:

Consider the example log file where the entire file is a single log entry:

```
+-----+  
-----+  
Application Object Library: Version : 12.2  
Copyright (c) 1998, 2013, Oracle and/or its affiliates. All rights  
reserved.  
FNDWFBG: Workflow Background Process  
+-----+  
-----+
```

```

Current system time is 04-JUL-2018 17:25:23
+-----+
**Starts**04-JUL-2018 17:25:23
**Ends**04-JUL-2018 18:25:23
+-----+
Start of log messages from FND_FILE
+-----+
+-----+
End of log messages from FND_FILE
+-----+
Successfully resubmitted concurrent program FNDWFBG with request ID
239834523 to start at 04-JUL-2018 18:30:23 (ROUTINE=IERKWEP)
+-----+
No completion options were requested.
Output file size:
0
Output is not being printed because:
The print option has been disabled for this report.
+-----+
Concurrent request completed successfully
Current system time is 04-JUL-2018 18:30:23
+-----+

```

1. Create the base parser. See [Create a Parser](#).

```

.*?Current system time is {TIMEDATE}.*\**Starts\**\*(\[S\ ]+).*?
\**Ends\**\*(\[S\ ]+).*

```

The fields that you must select for parsing are:

- Version
- Event Start Time
- Event End Time

For the above example log, select the **Handle entire file as a single log entry** check box. No header regex is required.

2. Open the **Create Log Source** dialog box. See [Create a Log Source](#).

3. Select the base parser that you created in step 1.

4. Provide the file path for the example log.

5. Navigate to **Data Filters** tab.

6. **Convert Month Short Name to Number:**

If your log already has a numeric month number instead of a month name, then you can skip this step and go to step 7.

If your log has the short name for the month instead of the month number, then, to convert the month short name to month number, add twelve data filters of the type **MASK**.

For each calendar month, the data filter will have similar details as for January month below:

- **Name:** Jan to 01

- **Type:** Mask
- **Find Expression:** (**\w+**\d{2}-) (JAN) (-\d{4})
- **Replace Expression:** \$101\$3

Order	Name	Type	Find Expression	Replace Expression
1	jan to 01	Mask	(**\w+**\d{2}-)(JAN)(-\d{4})	\$101\$3
2	feb to 02	Mask	(**\w+**\d{2}-)(FEB)(-\d{4})	\$102\$3
3	mar to 03	Mask	(**\w+**\d{2}-)(MAR)(-\d{4})	\$103\$3
4	apr to 04	Mask	(**\w+**\d{2}-)(APR)(-\d{4})	\$104\$3
5	may to 05	Mask	(**\w+**\d{2}-)(MAY)(-\d{4})	\$105\$3

The data mask finds occurrences of the time pattern in the log:

```
**Starts**04-JAN-2018 17:25:23
```

It captures the data before JAN, the value JAN, and the data after JAN into three capture groups. The capture groups are indicated with the three pairs of parentheses ().

Then in the replace expression, the value from the first capture group is replaced using \$1, the value JAN is replaced with 01, and the third capture group is replaced using \$3.

After the data filter is implemented, the time and data information appears as follows:

```
**Starts**04-01-2018 17:25:23
```

7. Rewrite the time and date information in ISO-8601 format:

Now that the time and date information is available in the right data type, rewrite the time and date data to be in the ISO-8601 format using two data filters for the example log:

These two data filters *must* be positioned after the twelve data filters that you created to convert month short name to month number. This'll ensure that the time and date data format is evaluated after the month short name is converted to month number. Use the up and down arrows to change the order of the data filters.

Create the following two data filters to convert the start time and end time data to ISO-8601 format:

- **Name:** Change shape of Starts
 - **Type:** Mask
 - **Find Expression:** **Starts**(\d+)-(\d+)-(\d{4})\s(\d{2}:\d{2}:\d{2})
 - **Replace Expression:** **Starts**\$3-\$2-\$1T\$4.000Z
- **Name:** Change shape of Ends

- **Type:** Mask
- **Find Expression:** `**Ends**(\d+)-(\d+)-(\d{4})\s(\d{2}:\d{2}:\d{2})`
- **Replace Expression:** `**Ends**$3-$2-$1T$4.000Z`

Order	Name	Type	Find Expression	Replace Expression
13	Change shape of Starts	Mask	<code>**Starts**(\d+)-(\d+)-(\d{4})\s(\d{2}:\d{2}:\d{2})</code>	<code>**Starts**\$3-\$2-\$1T\$4.000Z</code>
14	Change shape of Ends	Mask	<code>**Ends**(\d+)-(\d+)-(\d{4})\s(\d{2}:\d{2}:\d{2})</code>	<code>**Ends**\$3-\$2-\$1T\$4.000Z</code>

In the find expression, each element of time and date is captured. In the replace expression, the order of the time and date elements are changed. The values \$1, \$2, \$3, and \$4 correspond to the capture groups in the find expression. The capture groups are indicated with the pairs of parentheses ().

The static `.000Z` is added in the replace expression to store the time and date value in the field. This effectively stamps the time and date in **Coordinated Universal Time** (UTC) time zone. If your log entry instead was in Pacific Standard Time (PST) time zone, then its time zone is artificially changed to UTC, but without the actual shift in the hour.

 **Note:**

Currently, it is not possible to shift the time and date value into a different time zone at ingest time. But you can do this from the Log Explorer by using the **eval** command:

- After you have stored the event start and event end time, subtract the event end time from the event start time to get the event duration.
- Add or subtract the duration of time difference between the time zones calculated in milliseconds.
- Convert the output from number of milliseconds to the time and date format.

See Eval Command in *Using Oracle Log Analytics Search*.

8. In the **Extended Fields** tab, add the extended field definitions to the log source to extract the time and date fields:
 - From **Message** field: `**Starts**{Event Start Time:\S+}`
 - From **Message** field: `**Ends**{Event Start Time:\S+}`
9. Save the new log source that you created.

Event End Time	1530728723000
Event Start Time	1530725123000

You can now notice that the two fields **Event Start Time** and **Event End Time** are populated with the date and time values from the log. In the Log Explorer, you can see the times as milliseconds since epoch.

Perform Advanced Analytics with Cluster Compare

Following are some typical scenarios for using the Cluster Compare utility. You can compare two sets of log data by reducing the duplicates and showing only the unique clusters found in each set. This can possibly find the root-cause for an issue by removing the duplicate clusters.

Topics:

- [Cluster Compare by Time Shift](#)
- [Cluster Compare by Custom Time](#)
- [Cluster Compare by Current Time](#)

For steps to use the Cluster Compare utility, see [Use Cluster Compare Utility](#).

For the syntax and other details of the `clustercompare` command, see [Clustercompare Command](#) in *Using Oracle Log Analytics Search*.

Cluster Compare by Time Shift

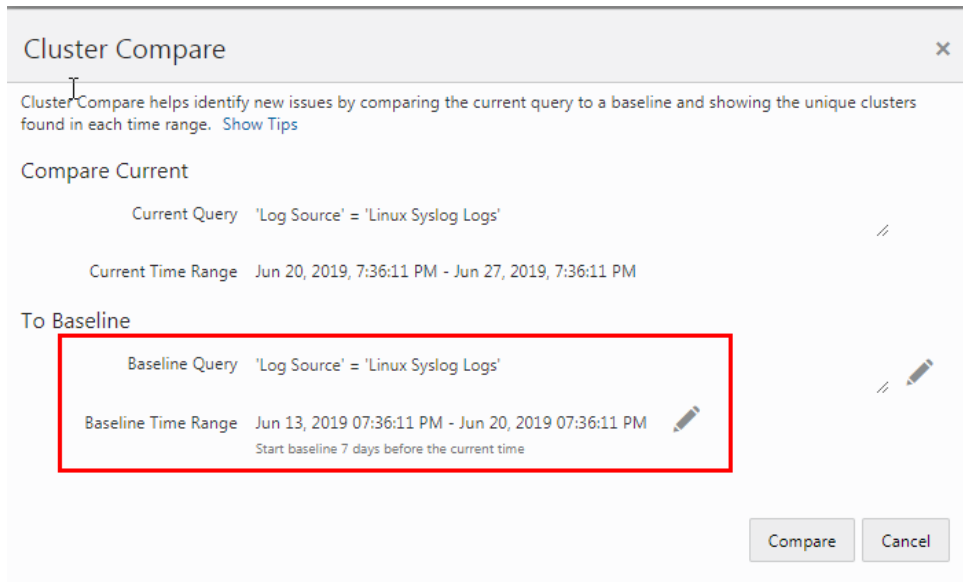
To generate useful analytics by reducing the number of clusters to only the clusters that are unique in the current time period, then use the Time Shift option. This is the default option available with the cluster compare utility.

Consider that we want to compare the log data from the log source `Linux Syslog Logs` collected over the current week, and the past week.

```
|=====|=====|
  Baseline Time Range      Current Time Range
<----Use the same query in both the time ranges---->
```

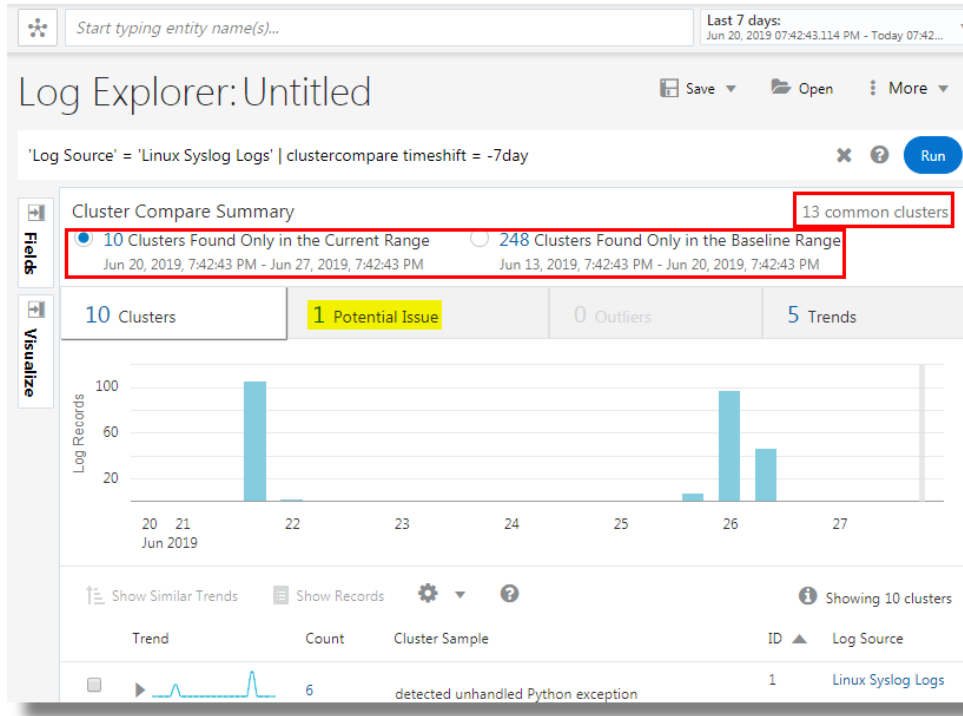
Select the current time range from the time selector as `Last 7 days` and specify the query `'Log Source' = 'Linux Syslog Logs' | cluster`. For the cluster compare utility, this qualifies as the current time range and current query.

Click **Cluster Compare** and notice that the baseline query is the same as the current query. Also, note that the baseline time range is already selected by default, which is a week before the current week. Click **Compare**.



The Cluster Compare summary is displayed as follows:

- 10 clusters are found only in the current range
- 248 clusters are found only in the baseline range
- 13 common clusters are found in both the ranges



Using this data, you can identify the unique potential issue in the current week, and find a root-cause. Narrow down your selection of log records to those are the cause for the potential issue.

Note: The time shift value is subtracted from the start and end of the current time. If the time shift is less than the duration of the current time, there will be an overlap. This will show all the common (duplicate) clusters from that overlap period. A message will be shown when this is detected. In such a case, the baseline query is the same as the current query.


Cluster Compare by Custom Time

If you want to compare the log data from the same source but over two custom time ranges, then use the Custom Time option in the cluster compare utility.

Consider that we want to compare the log data of the entity type `Host (Linux)` collected over the current time range in the month June 2019 and the baseline time range in the month August 2016.

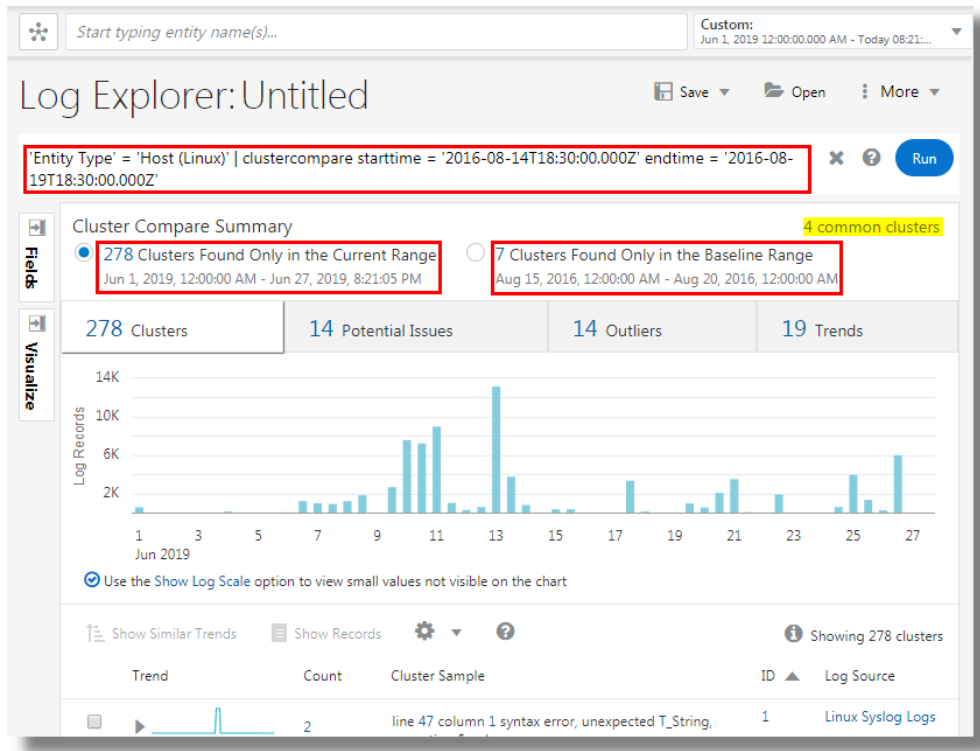
```
|=====|
=====|
Baseline Time Range                               Current Time
Range
<----->Use the same query in both the time
ranges<----->
```

Select the current time range from the time selector for the period `June 1, 2019 12:00 AM to June 27, 2019 8:21 PM` and specify the query `'Entity Type' = 'Host (Linux)' | cluster`. For the cluster compare utility, this qualifies as the current time range and current query.

Click **Cluster Compare** and notice that the baseline query is the same as the current query. Click the  icon next to the **Baseline Time Range** and select **Use Custom Time**. Specify the custom time range `Aug 15, 2016 12:00 AM to Aug 20, 2016 12:00 AM`. Click **Compare**.

The Cluster Compare summary is displayed as follows:

- 278 clusters are found only in the current range
- 7 clusters are found only in the baseline range
- 4 common clusters are found in both the ranges



This analysis can enable you to compare the syslog data from the entity type over the two periods, eliminate the common clusters, and view the unique clusters. In this case, the increase in the number of logs pertaining to the potential issues from the baseline range to current time range can be analyzed by viewing the logs pertaining to the potential issues in the current time range.

Cluster Compare by Current Time

If you want to compare the logs from different sources in the same time range, then use Cluster Compare by current time and select the logs from different entity types or log sources.

Consider a case where an error is reported on the node of a Rideshare application `rs_host01` but not on the node `rs_host03`. Both the nodes can then be compared using the same time range Aug 14, 2016, 9:30:00 AM to Aug 20, 2016, 9:30:00 AM to detect variations and identify issues which can then be root-caused. Both the nodes have approximately 20,000 log records to compare and analyze.


```


|=====|
<----Baseline Time Range = Current Time Range---->
<-----Baseline Query----->
<-----Current Query----->

```

Select the current time range from the time selector as Aug 14, 2016, 9:30:00 AM to Aug 20, 2016, 9:30:00 AM and specify the query `Entity = rs_host01`. For the cluster compare utility, this qualifies as the current time range and current query.

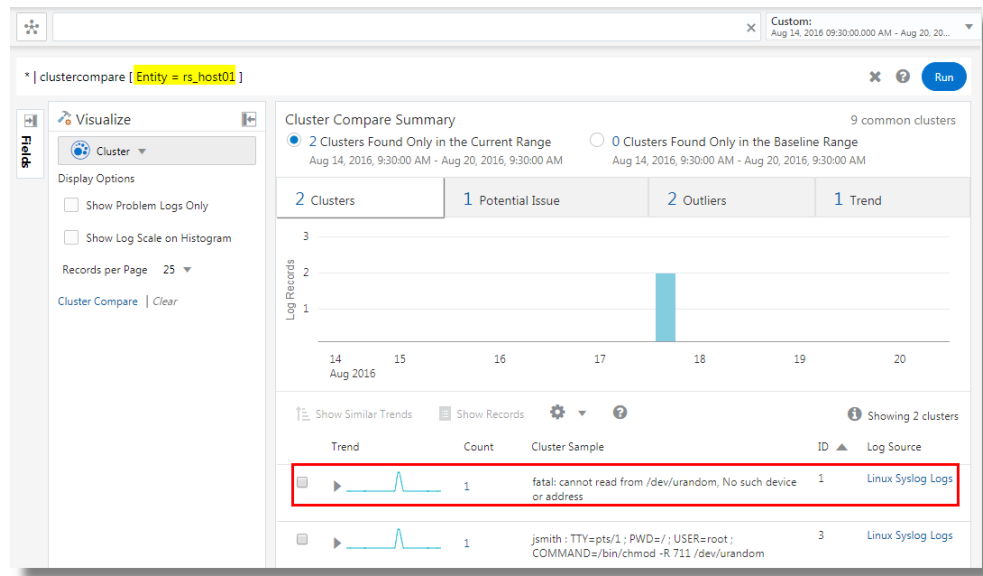
Click **Cluster Compare** and notice that the baseline query is the same as the current query.

Click  and modify the baseline query to `Entity = rs_host03`. By default, the baseline time

range is time shifted. Click  next to the baseline time range and select the option **Use Current Time**. Click **Compare**.

The Cluster Compare summary is displayed as follows:

- 2 clusters are found only in the current range
- 0 clusters are found only in the baseline range
- 9 common clusters are found in both the ranges



Note that in the same time range, the two Rideshare nodes have 9 common clusters, and the node `rs_host01` has 2 unique clusters. Evidently, the cluster table lists the fatal error which caused the issue in the node that's analyzed.

This analysis eliminates the complexity of comparing 20,000 records from both the nodes by removing the common clusters, and identifying unique clusters resulting in fewer number of records to analyze.

Machine Learning Based Query Enrichment

Typically, you can derive rich insights about the log records using the *Cluster* and *Link* features. But it is not possible to use the insights generated from these analytical tools in other Oracle Log Analytics visualizations. Now use the new **insights** feature that auto-analyzes the results of a search, and returns a set of ML-derived fields that capture the insights.

Generate Insights Fields Using the `addinsights` Command

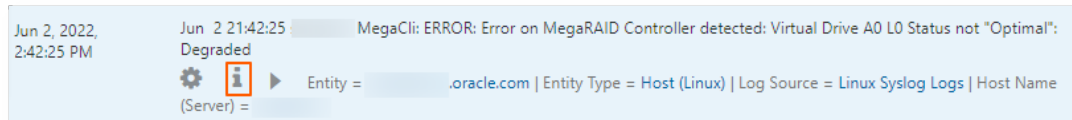
Run the `addinsights` command after a search to analyze the given query and automatically enrich the query results with additional insight information for each log record. The following insights fields are returned by the command: *Cluster Record*

Count, Shape Record Count, Shape Cluster Count, Potential Issue, and Shape ID. See Addinsights Command in Using Oracle Log Analytics Search.


Following is an example query with `addinsights` for Linux Syslog Logs:

```
'Log Source' = 'Linux Syslog Logs' | addinsights
```

The following image shows the result of running the example query.



Click on the *info* icon to view the insights fields that are auto-generated based on the analysis of the query search results.

Insight Field Name	Insight Field Value
Cluster Record Count	4183
Shape Record Count	11318
Shape Cluster Count	3
Issue	Potential
Cluster Trend	

Use the Insights to View Similar Log Records

This time, run a similar query on Database Alert Logs.

```
'Log Source' = 'Database Alert Logs' | addinsights
```

After the query is run with the `addinsights` command, scroll down the search result to the log record that you are interested in, expand to view the fields, click **Add To Search** on the *Cluster Record Count* field view.

Original Log Content

```
Sat Oct 01 06:43:00
Errors in file /... \C2_j001_10993.trc:
ORA-12012: error on auto execute of job "... CORECARDS_MV"
ORA-02049: timeout: distributed transaction waiting for lock
ORA-06512: at "SYS.DBMS_SNAPSHOT", line 2821
ORA-06512: at "SYS.DBMS_SNAPSHOT", line 3058
ORA-06512: at "SYS.DBMS_SNAPSHOT", line 3017
ORA-06512: at "...", line 55
ORA-06512: at line 1
```

Entity = ... | Entity Type = Oracle Database Instance | Log Source = Database Alert Logs

Field Name	Field Value
Cluster Record Count	34097
Cluster Trend	Add To Search
Entity	Exclude From Search
Entity Type	
Error ID	ORA-12012
Error Text	ORA-12012: error on auto execute of job "... RECORDS_MV" ORA-02049: timeout: distributed transaction waiting for lock

This now updates the query to the one below, showing only the log records that have similar Cluster Count.

```
'Log Source' = 'Database Alert Logs' | addinsights | where 'Cluster Record Count' = 34097
```

Switch to the Cluster visualization. It shows all the clusters that are similar to the selected message. You can click on a variable to see the specific values for that variable.

Trend	Count	Cluster Sample
	14880	Errors in file ... IRAC2/trace/... IRAC2_j001_12 12.trc: ORA-12012: error on auto execute of job "... IT_B_34893618" ORA-20001: Bug ... Bug ... is locked. try again later. ... ORA-06512: at "..._ERROR", line 45 ORA-06512: at "...DB", line 936 ORA-06512: at "...DB", line 529 ORA-06512: at line 1 3 more samples...

Examples of Semantic Clustering Using Natural Language Processing

The `nlp` command can be used to extract keywords from a string field, or to cluster records based on these extracted keywords. Keyword extraction can be controlled

using a custom NLP dictionary. If no dictionary is provided, the default out-of-the-box dictionary is used.

Topics:

- [Cluster Kernel Errors in Linux Syslog Logs](#)
- [Cluster the Database Alert Logs](#)

For more information on semantic clustering, see [Semantic Clustering Using Natural Language Processing](#).

Cluster Kernel Errors in Linux Syslog Logs

The following query clusters Kernel messages in Linux Syslog Logs:

```
'Log Source' = 'Linux Syslog Logs' and kernel
| link cluster()
| where 'Potential Issue' = '1'
| nlp table = 'iSCSI Errors' cluster('Cluster Sample') as 'Cluster ID',
           keywords('Cluster Sample') as Summary
| sort 'Cluster ID'
```

In the above query:

- `link cluster()` runs the traditional cluster and returns a `Cluster Sample` field.
- `nlp cluster('Cluster Sample')` processes each `Cluster Sample` and assigns a `Cluster ID`. Messages that have similar meaning would get the same `Cluster ID`.
- `keywords('Cluster Sample')` extracts the keywords used in clustering. This is returned in the `Summary` field.

The following image shows the link results returned:

Cluster Sample	Count	Start Time	End Time	Potential Issue	Cluster ID	Summary
[8358765.214932] EXT4-fs warning (device dm-0): ext4_end_bio:322: I/O error -5 writing to inode 1311730 (offset 0 size 0 starting block 5517537)	7	Jun 17, 2020, 7:52:04 AM	Jun 17, 2020, 7:52:05 AM	1	-1545762356	block, device, end, error, offset, size, starting, warning, writing
[36064444.898398] blk_update_request: I/O error, dev sdc, sector 2048	837	Jun 16, 2020, 12:24:03 PM	Jun 19, 2020, 11:56:34 AM	1	-1965012324	error, request, sector, update
[22949543.841882] Buffer I/O error on dev dm-11, logical block 0, async page read	260	Jun 16, 2020, 2:05:12 PM	Jun 19, 2020, 11:56:34 AM	1	-393301095	block, buffer, error, logical, page, read
[10791670.239623] Buffer I/O error on dev dm-11, logical block 11206675, lost async page write	41	Jun 17, 2020, 7:52:04 AM	Jun 18, 2020, 7:38:02 PM	1	-393301095	block, buffer, error, logical, lost, page, write
Kernel reported iSCSI connection 2i0 error (1021 - ISCSI_ERR_SCSI_EH_SESSION_RST: Session was dropped as a result of SCSI error recovery) state (1)	4	Jun 17, 2020, 6:12:05 AM	Jun 17, 2020, 7:53:22 AM	1	-401516367	connection, err, error, kernel, recovery, reported, result, scsi, session, state

- The first and second rows are not similar, and hence get different cluster IDs.
- The third and fourth rows have similarity in the Cluster Sample. This can be seen in the overlap of keywords extracted in the Summary field.
- By default, a 70% overlap is required to form a cluster. This can be overridden using the similarity parameter to cluster.
- The Cluster ID generated is deterministic. Thus, the Cluster ID can be used as a shortcut for the list of keywords shown in the Summary column.

Use similarity to Control the Number of Clusters

Running cluster using the default dictionary and a lower similarity threshold would produce fewer clusters:

```
'Log Source' = 'Linux Syslog Logs' and kernel
| link cluster()
| where 'Potential Issue' = '1'
| nlp similarity=0.2 cluster('Cluster Sample') as 'Cluster ID',
      keywords('Cluster Sample') as Summary
| sort 'Cluster ID'
```

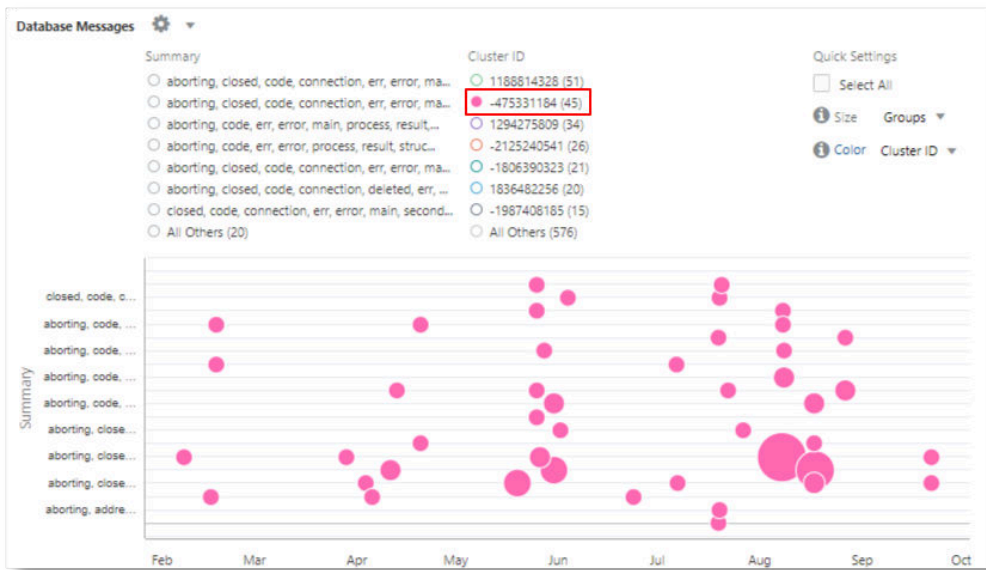
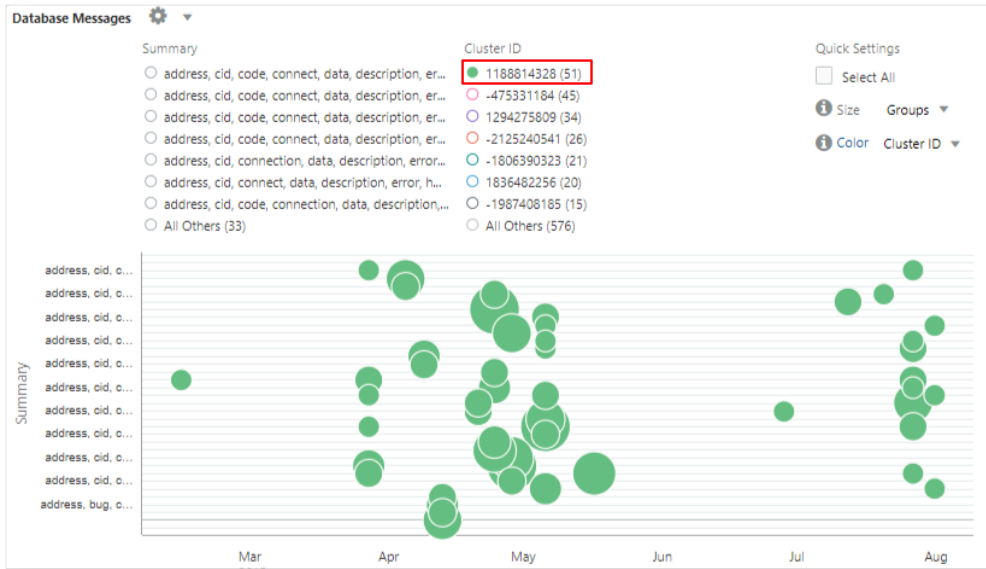
This merged some of the rows into the existing clusters, as well as reduced the number of clusters:

Cluster Sample	Count	Start Time	End Time	Potential Issue	Cluster ID	Summary
[22949543.841882] Buffer I/O error on dev dm-11, logical block 0, async page read	260	Jun 16, 2020, 2:05:12 PM	Jun 19, 2020, 11:56:34 AM	1	-393301095	block, buffer, error, logical, page, read
[10791670.239623] Buffer I/O error on dev dm-11, logical block 11206675, lost async page write	41	Jun 17, 2020, 7:52:04 AM	Jun 18, 2020, 7:38:02 PM	1	-393301095	block, buffer, error, logical, lost, page, write
[8358765.214932] EXT4-fs warning (device dm-0): ext4_end_bio:322: I/O error -5 writing to inode 1311730 (offset 0 size 0 starting block 5517537)	7	Jun 17, 2020, 7:52:04 AM	Jun 17, 2020, 7:52:05 AM	1	-393301095	block, device, end, error, offset, size, starting, warning, writing
[36064444.898398] blk_update_request: I/O error, dev sdd, sector 2048	837	Jun 16, 2020, 12:24:03 PM	Jun 19, 2020, 11:56:34 AM	1	-485003274	error, request, sector, update
[22874587.450168] EXT4-fs (dm-11): previous I/O error to superblock detected	13	Jun 17, 2020, 7:52:04 AM	Jun 18, 2020, 2:43:58 PM	1	-485003274	detected, error, previous, superblock

Cluster the Database Alert Logs

The following query shows an example of semantically clustering *Database Alert Logs*:

```
'Log Source' = 'Database Alert Logs'
| link cluster()
| nlp cluster('Cluster Sample') as 'Cluster ID',
  keywords('Cluster Sample') as Summary
| where Summary != null
| classify 'Start Time', Summary, 'Cluster ID' as 'Database Messages'
```



A

Out-of-the-Box Log Sources

Oracle Log Analytics provides the following out-of-the-box log sources that you can use.

AIX Audit Logs

AIX Cron Logs

AIX Dynamic System Optimizer Logs

AIX HACMP Cluster Logs

AIX SU Logs

AIX Syslog Logs

Apache Cassandra DB Garbage Collection Logs

Apache Cassandra DB System Logs

Apache Hadoop Standard Logs

Apache Hive Logs

Apache HTTP Server Access Logs

Apache HTTP Server Error Logs

Apache HTTP Server SSL Access Logs

Apache HTTP SSL Request Logs

Apache Kafka Logs

Apache Spark Logs

Apache Tomcat Access Logs

Apache Tomcat Catalina Logs

Apache Tomcat Error Logs

Apache Tomcat Host Logs

Apache Zookeeper Logs

ArcSight Common Event Format Source

Automatic Storage Management Alert Logs

Automatic Storage Management Trace Logs

AVDF Alert Linux Syslog

AVDF Event in Oracle Database

Bluecoat Proxy Squid Logs

Bluecoat Proxy W3C Logs

Check Point Firewall LEA Syslog Logs
Citrix Netscaler Logs
Clusterware Disk Monitor Logs
Clusterware Ready Services Alert Logs
Clusterware Ready Services Daemon Logs
Database Alert Logs
Database Audit Logs
Database Audit XML Logs
Database Incident Dump Files
Database Listener Alert Logs
Database Listener Trace Logs
Database Trace Logs
EBS Concurrent Manager Logs
EBS Concurrent Request Logs
EBS Conflict Resolution Manager Logs
EBS Internal Concurrent Manager Logs
EBS Output Post Processor Logs
EBS Transaction Manager Logs
EBS Workflow Notification Mailer Logs
EM Cloud Control Agent AJTS Logs
EM Cloud Control Agent EMCTL Logs
EM Cloud Control Agent Host Target Event Logs
EM Cloud Control Agent JVMGC Logs
EM Cloud Control Agent Logs
EM Cloud Control Agent PFU Logs
EM Cloud Control Agent STDOUT Logs
EM Cloud Control OMS Access Logs
EM Cloud Control OMS Diagnostics Logs
EM Cloud Control OMS Logs
EM Cloud Control OMS STDOUT Logs
EM Cloud Services Agent AJTS Logs
EM Cloud Services Agent EMCTL Logs
EM Cloud Services Agent Host Target Logs

EM Cloud Services Agent JVMGC Logs
EM Cloud Services Agent Log Collector Logs
EM Cloud Services Agent Logs
EM Cloud Services Agent PFU Logs
EM Cloud Services Agent STDOUT Logs
F5 Big IP ASM WAF Syslog CEF Logs
F5 Big IP Logs
FMW BI JBIPS Logs
FMW BI Publisher Logs
FMW OAM Embedded LDAP Access Logs
FMW OHS Access Logs (V11)
FMW OHS Access Logs (V12)
FMW OHS Admin Access Logs (V12)
FMW OHS Diagnostic Logs (V11)
FMW OHS Error Logs
FMW OHS OPMN Logs (V11)
FMW OHS Server Logs (V12)
FMW OID Audit Logs
FMW OID Directory Control Logs
FMW OID Directory Dispatcher Server Logs
FMW OID Directory Replication Server Logs
FMW OID Directory Server Logs
FMW OID Monitor Logs
FMW OID OPMN Logs
FMW WLS Node Manager Log
FMW WLS Server Access Logs
FMW WLS Server Diagnostic Logs
FMW WLS Server Logs
FMW WLS Server STDOUT Logs
Fusion Apps Diagnostic Logs
IBM DB2 Audit Logs
IBM DB2 Diagnostic Logs
IBM Websphere Application Server (Classic) Logs

IBM Websphere Application Server (Classic) System Error
Identity and Access Management Audit Database
IPTables Logs
JBOSS EAP Log Source
Juniper SRX Syslog Logs
KSplice Logs
Linux Audit Logs
Linux Cron Logs
Linux Exadata Cell Alert Logs
Linux Exadata Cell Management Server Logs
Linux Exadata Cell Management Server Trace Logs
Linux Mail Delivery Logs
Linux Secure Logs
Linux Syslog Logs
Linux YUM Logs
McAfee VirusScan Enterprise Logs
Microsoft Active Directory Distributed File System Replication Logs
Microsoft Active Directory Installation Wizard Logs
Microsoft Active Directory Netsetup Logs
Microsoft Active Directory NtFrsApi Logs
Microsoft DHCP (IPv4) Logs
Microsoft DHCP (IPv6) Logs
Microsoft DNS Logs
Microsoft Exchange Active Monitoring Trace Logs
Microsoft Exchange Authentication Admin Logs
Microsoft Exchange Database Availability Logs
Microsoft Exchange Diagnostics Service Logs
Microsoft Exchange Outlook Web Access Probe Logs
Microsoft IIS Log Source for FTP format logs
Microsoft IIS Log Source for IIS format logs
Microsoft IIS Log Source for NCSA format logs
Microsoft IIS Log Source for W3C format logs
Microsoft SharePoint Logs

Microsoft SQL Server Agent Error Log
Microsoft SQL Server Error Log Source
Microsoft .Net Log4Net Logs
MongoDB Logs
MySQL Database Audit XML Logs
MySQL Error Logs
MySQL General Log Source Stored in Database
MySQL General Query Logs
MySQL Slow Query Logs
NetApp Syslog Logs
NGINX Access Logs
NGINX Error Logs
Node.js Log4js Logs
OCI Audit Logs
OCI VCN Flow Logs
OMC Compliance Assessment Result Logs
OMC Orchestration Service Output Logs
OMC Security Monitoring Analytics Event Format (XML) Source
Oracle Access Manager Audit Logs
Oracle EBS Transaction Logs
Oracle DB Audit Log Source Stored in Database
Oracle DB Audit Log Source Stored in Database for Unified Audit Trail
PeopleSoft Analytics Engine Server Logs
PeopleSoft Application Analytics Engine Server Logs
PeopleSoft Application server domain Application Server (APPSRV) Process Logs
PeopleSoft Application server domain Monitor Server (MONITORSRV) Process Logs
PeopleSoft Application server domain Watch Server (WATCHSRV) Process Logs
PeopleSoft Application Tuxedo Access Logs
PeopleSoft Application Tuxedo User Logs
PeopleSoft Integration Gateway Error Logs
PeopleSoft Integration Gateway Message Logs
PeopleSoft Master Scheduler Server Logs
PeopleSoft Process Scheduler App Engine Server Logs

PeopleSoft Process Scheduler Distribution Agent Logs
PeopleSoft Process Scheduler Master Scheduler Logs
PeopleSoft WLS Server Access Logs
PeopleSoft WLS Server Logs
PeopleSoft WLS Server STDOUT Logs
PeopleSoft WLS Servlet Logs
PostgreSQL Logs
SAP Application Startup Logs
SAP Application Transport Logs
SAP Dev Dispatcher Logs
SAP Dev ICM Security Logs
SAP Dev Message Server Logs
SAP Dev RD Logs
SAP Java Server Application Logs
SAP Java Server Default Trace Logs
SAP VMC Available Logs
Siebel Component Logs
Siebel Gateway Name Server Audit Logs
Siebel Gateway Server Logs
Solaris Audit Logs
Solaris ILOM Configuration Logs
Solaris Install Logs
Solaris SMF Daemon Logs
Solaris SU Logs
Solaris Syslog Logs
Squid Proxy Access Logs
SUDO Logs
Ubuntu Secure Logs
Ubuntu Syslog Logs
Windows Application Events
Windows Security Events
Windows Setup Events
Windows System Events

 **Note:**

The preceding list is evolving. Check with the product user interface for the latest list of log sources.

B

Understand the Search Commands

The Search Language for analyzing the logs allows you to specify what action to perform on the search results.

Commands can be either search commands or statistical commands.

Search Commands

Search commands are those commands which further filter the available log entries.

The following table lists the search commands and provides a brief description of each.

Command	Description
<code>addfields</code>	Use this command to generate aggregated data within groups identified by the <code>link</code> command. See Addfields Command in <i>Using Oracle Log Analytics Search</i> .
<code>addinsights</code>	Use this command to view additional insight information in each log record. See Addinsights Command in <i>Using Oracle Log Analytics Search</i> .
<code>bottom</code>	Use this command to display a specific number of results with the lowest aggregated value as determined by the specified field. See Bottom Command in <i>Using Oracle Log Analytics Search</i> .
<code>bucket</code>	Use this command to group the log records into buckets based on the range of values of a field. See Bucket Command in <i>Using Oracle Log Analytics Search</i> .
<code>classify</code>	Use this command to cluster properties of groups identified by the <code>link</code> command. See Classify Command in <i>Using Oracle Log Analytics Search</i> .
<code>cluster</code>	Use this command to group similar log records. See Cluster Command in <i>Using Oracle Log Analytics Search</i> .
<code>clustercompare</code>	Use this command to compare one cluster collection with another, and for viewing the clusters that exist exclusively in the current range versus clusters that exist exclusively in the baseline range. See Clustercompare Command in <i>Using Oracle Log Analytics Search</i> .
<code>clusterdetails</code>	Use this command to return similar log records. See Clusterdetails Command in <i>Using Oracle Log Analytics Search</i> .
<code>clustersplit</code>	Use this command to view the log data within a cluster for specific classify results in the tabular format. See Clustersplit Command in <i>Using Oracle Log Analytics Search</i> .
<code>compare</code>	Use this command to compare properties generated by the <code>link</code> command over the comparison intervals specified. See Compare Command in <i>Using Oracle Log Analytics Search</i> .
<code>createview</code>	Use this command to define a subquery to create a subset of groups identified by the <code>link</code> command. See Createview Command in <i>Using Oracle Log Analytics Search</i> .

Command	Description
distinct	Use this command to remove duplicates from the returned results. See Distinct Command in <i>Using Oracle Log Analytics Search</i> .
eval	Use this command to calculate the value of an expression and display the value in a new field. See Eval Command in <i>Using Oracle Log Analytics Search</i> .
eventstats	Use this command to obtain overall summary statistics, optionally grouped by fields, on properties of groups identified by the <code>link</code> command. Its output will include one field for each aggregation. See Eventstats Command in <i>Using Oracle Log Analytics Search</i> .
fields	Use this command to specify which fields to add or remove from the results. See Fields Command in <i>Using Oracle Log Analytics Search</i> .
fieldsummary	Use this command to return data for the specified fields. See Fieldsummary Command in <i>Using Oracle Log Analytics Search</i> .
head	Use the <code>head</code> command to display the first <i>n</i> number of results. See Head Command in <i>Using Oracle Log Analytics Search</i> .
highlightgroups	Use this command to match strings or search criteria on the properties of the groups identified by the <code>link</code> command, and causes them to be highlighted in the link visualization. See Highlightgroups Command in <i>Using Oracle Log Analytics Search</i> .
highlightrows	Use this command to match a string or a list of strings, and highlight the entire row in the Log UI. See Highlightrows Command in <i>Using Oracle Log Analytics Search</i> .
highlight	Use this command to match a string or a list of strings, and highlight them in the Log UI. See Highlight Command in <i>Using Oracle Log Analytics Search</i> .
link	Use this command to group log records into high level business transactions. See Link Command in <i>Using Oracle Log Analytics Search</i> .
lookup	Use this command to invoke field value lookups. See Lookup Command in <i>Using Oracle Log Analytics Search</i> .
map	Use this command to join a view with the groups identified by the <code>link</code> command to create new properties. See Map Command in <i>Using Oracle Log Analytics Search</i> .
nlp	Use this command to apply natural language processing algorithms to a text field. See NLP Command in <i>Using Oracle Log Analytics Search</i> .
regex	Use this command to filter data according to a specified regular expression. See Regex Command in <i>Using Oracle Log Analytics Search</i> .
rename	Use this command to change the name of a field. See Rename Command in <i>Using Oracle Log Analytics Search</i> .
search	Use this command to retrieve a specific logical expression from the available log data. See Search Command in <i>Using Oracle Log Analytics Search</i> .
searchLookup	Use this command to retrieve contents from a lookup table. See SearchLookup Command in <i>Using Oracle Log Analytics Search</i> .
sort	Use this command to sort logs according to specified fields. See Sort Command in <i>Using Oracle Log Analytics Search</i> .

Command	Description
tail	Use this command to display the last <i>n</i> number of results. See Tail Command in <i>Using Oracle Log Analytics Search</i> .
timecluster	Use this command to group the time-series charts together based on how similar they are to one another. See Timecluster Command in <i>Using Oracle Log Analytics Search</i> .
top	Use this command to display a specified number of results with the highest aggregated value as determined by the specified field. See Top Command in <i>Using Oracle Log Analytics Search</i> .
where	Use this command to calculate the value of an expression to be true or false. See Where Command in <i>Using Oracle Log Analytics Search</i> .

Statistical Commands

Statistical commands perform statistical operations on the search results.

The following table lists the supported statistical commands, and provides a short description for each.

Commands	Description
distinct	Use this command to remove duplicate entries from the search results. See Distinct Command in <i>Using Oracle Log Analytics Search</i> .
stats	Use this command to provide summary statistics for the search results, optionally grouped by a specified field. See Stats Command in <i>Using Oracle Log Analytics Search</i> .
timestats	Use this command to generate data for displaying statistical trends over time, optionally grouped by a specified field. See Timestats Command in <i>Using Oracle Log Analytics Search</i> .

C

Entity Types Modeled in Oracle Log Analytics

Oracle Log Analytics supports the following types of entities.



Note:

This list of entities will constantly evolve as and when new entity types are added. Users can also create their own entity types.

Entity Types

- Amazon Web Services (S3)
- Apache Hadoop
- Apache Hive
- Apache Kafka
- Apache Zookeeper
- Automatic Storage Management
- Automatic Storage Management Instance
- Cassandra
- Cisco ASA
- Cisco Ethernet Switch
- Cluster
- Container
- DB2
- Docker Container
- Docker Engine
- EMC VMAX
- EMC VNX
- F5 BigIP
- Generic System
- Group
- Host

Host (AIX)
Host (HP-UX)
Host (Linux)
Host (Solaris)
Host (Windows)
Hosted Target
Hyper-V
IBM Websphere
IBM WebSphere MQ
J2EE Application
Java Application Server
Java EE Application Server
Juniper SRX
LDAP Server
Listener
Load Balancer
Microsoft .NET Server
Microsoft AD
Microsoft Exchange
Microsoft IIS
Microsoft Internet Information Services
Microsoft Internet Information Services Web Site
Microsoft SharePoint
Microsoft SQL Server Database
Microsoft SQL Server Database Instance
Middleware Cluster
Middleware Domain
MongoDB
MySQL Database
MySQL Database Instance
NetApp FAS
NetApp FlexPod
NGINX

Node.js
OpenStack
Operating System
Oracle Access Management Cluster
Oracle Access Management Server
Oracle Business Intelligence (BI)
Oracle Cluster Node
Oracle Clusterware
Oracle Database
Oracle Database Cluster Listener
Oracle Database Instance
Oracle Database Listener
Oracle E-Business suite
Oracle Exadata Database Machine
Oracle Exadata Storage Server
Oracle Exadata Storage Server Grid
Oracle Hadoop Cluster
Oracle Hadoop HDFS
Oracle Hadoop Yarn
Oracle Home
Oracle HTTP Server
Oracle ILOM Server
Oracle InfiniBand Switch
Oracle Internet Directory
Oracle JD Edwards
Oracle PeopleSoft Application Server
Oracle PeopleSoft Internet Architecture
Oracle PeopleSoft Process Scheduler
Oracle PeopleSoft System
Oracle Power Distribution Unit (PDU)
Oracle Rack
Oracle Utilities Customer Care and Billing
Oracle VM

PostgreSQL
Ruby on Rails
SAP System
SAPNW Application Server Instance
SAPNW Application Server JAVA Server Process
Service Bus
Siebel Component
Siebel Enterprise
Siebel Server
SOA Infrastructure
Storage Manager
Storage Server
Switch
Sybase Adaptive Server Enterprise
System
Target
Tibco
Tomcat
Traffic Director Configuration
Traffic Director Instance
Virtual Platform
Virtual Server
VMWare
Web Application Server
WebLogic Cluster
WebLogic Domain
WebLogic Server

**Note:**

Please note that the preceding list is evolving.

D

SQL Query Guidelines

You should use the SQL queries that are used to extract the data carefully.

Follow these guidelines when writing SQL queries for extracting log data:

- Use read-only queries only.
- The credentials provided to execute the queries should have only the required privileges to extract the necessary data.
- The query performance is also an important consideration, because it can affect both the target database and other software running on the same host.
- The query should include at least one column that can be used to order the database records. This can be either some kind of a sequence number or a time-stamp column. Every new entry should have a value for this column that's equal to or greater than the one in older records. The SQL query will be run at regular intervals to extract new data. Oracle Log Analytics will use this column to identify the new records that have been introduced since the previous collection. It's recommended that this column should have an index to avoid full table scans.

E

List of Non-Facetable Fields

The following fields can't be filtered using the **Fields** panel.

- Alert Raised Time
- Call Stack Trace
- Data Received
- Data Sent
- Data
- Event End Time
- Error Stack Dump
- Event Generation Time
- First Event Time
- Message
- Resource Limit Settings
- SQL Bind Variables
- SQL Statement
- Stack Trace
- Supplemental Detail
- Supplemental Filename

F

Commonly Used Oracle Log Analytics Entities

This appendix lists the commonly used entities available with Oracle Log Analytics.

Entity Type	Entity Internal Name
Oracle Database	omc_oracle_db
Microsoft SQL Server Database	omc_sqlserver_db
MySQL Database	omc_mysql_db
Oracle WebLogic Server	omc_weblogic_j2eeserver
Oracle WebLogic Domain	omc_weblogic_domain
Oracle HTTP Server	omc_oracle_apache
Tomcat Server	omc_tomcat
Oracle Automatic Storage Management	omc_oracle_asm
Oracle Access Manager	omc_oracle_oam
Oracle Internet Directory	omc_oracle_oid
Websphere Server	omc_websphere_j2eeserver
Docker Engine	omc_docker_engine
Oracle Clusterware	omc_oracle_clusterware

G

Additional Entities in Oracle Log Analytics

This appendix lists the additional entities available with Oracle Log Analytics.

Entity Type (Display Name)	Property Names	Property Display Names
omc_apache_hive (Apache Hive)	host_name version hive_home log_dir	Host Name Version Install Home Location Logs directory of Hive
omc_apache_kafka (Apache Kafka)	kafka_broker_id kafka_listen_port kafka_broker_log_dir	Kafka Broker ID Kafka Broker Listen Port Kafka Broker Log Location
omc_apache_spark (Apache Spark)	host_name spark_home log_directory log_filename_pattern	Host Name Spark Home Log Directory Log File Name Pattern
omc_apache_zookeeper (Apache ZooKeeper)	host_name version listen_port zookeeper_home log_dir tracelog_dir	Host Name Version Zookeeper listening port Install Home Location Logs directory Trace logs directory
omc_aws_cloud_service_instance (AWS Cloud Service)	service_name credential_name access_url	Cloud Service Name Credential Name Access URL
omc_aws_ec2_instance (Amazon EC2 Instance)	unique_id datacenter_id entity_type_group tags credential_name	Cloud Unique Identifier Cloud Service Datacenter ID Entity Type Group Resource Tags Credential Name
omc_cassandra_db (Cassandra Database)	host_name cassandra_home cassandra_log_dir	Host Name Cassandra home/installation directory Cassandra log directory
omc_cisco_ace (Cisco Application Control Engine)	host_name version	Host Name Version
omc_cisco_asa (Cisco Adaptive Security Appliance)	host_name version	Host Name Version
omc_cisco_eth_switch (Cisco Ethernet Switch)	N/A	N/A

Entity Type (Display Name)	Property Names	Property Display Names
omc_citrix_netscaler (Citrix NetScaler)	admin_host_name admin_port log_directory	admin_host_name admin_port log_directory
omc_docker_container (Docker Container)	ContainerId IsKeyStoreProvided	Container ID Is Key Store Provided
omc_docker_engine (Docker Engine)	BaseURI DockerVersion host_name IsKeyStoreProvided	Base URL Docker Version Host Name Is Key Store Provided
omc_emc_vmax (EMC VMAX)	symmetrix_id install_dir	Symmetrix ID Installation Directory
omc_emc_vnx (EMC VNX)	host_name install_dir	Host Name Installation Directory
omc_f5_bigip (F5 BIG-IP)	host_name log_home shared_log_home	Server Name Log Home Shared Log Home
omc_flexpod (FlexPod)	fas_host_name fex_host_name ucs_host_name	NetApp Server Name Fabric Extender Server Name UCS Server Name
omc_generic_host (Host)	host_name	Host Name
omc_generic_load_balancer (Generic Load Balancer)	host_name	Host Name
omc_generic_relational_db (Generic Relational Database)	host_name	Host Name
omc_generic_virtual_server (Virtual Server)	N/A	N/A
omc_generic_web_application_server (Generic Web Application Server)	host_name port	Host Name Port
omc_host_aix (Host: AIX)	is_remote host_name	Is Remote Host Name
omc_host_hpux (Host: HP-UX)	is_remote host_name	Is Remote Host Name
omc_host_linux (Host: Linux)	is_remote host_name	Is Remote Host Name
omc_host_solaris (Host: Solaris)	is_remote host_name	Is Remote Host Name
omc_host_windows (Host: Windows)	is_remote host_name	Is Remote Host Name
omc_ibm_db2_database (IBM DB2 Database)	host_name inst_home	Host Name Instance Home Directory

Entity Type (Display Name)	Property Names	Property Display Names
omc_ibm_websphere_mq (IBM WebSphere MQ)	host_name mq_name mq_home mq_log_home mq_qm_name	Server Name MQ Name MQ Home MQ Log Home MQ QM Name
omc_jboss_j2eeserver (JBoss Server)	host_name http_port_list https_port_list jboss_home version server_log_dir	Host Name JBoss HTTP Port List JBoss HTTPS Port List JBoss Home Directory Version JBoss Server Log Dir
omc_jdedwards_eone (JD Edwards EnterpriseOne Server)	instance_name host_name port install_dir	Instance Name Host Name Port Number Installation Directory
omc_juniper_srx (Juniper SRX)	ip_address	IP Address
omc_microsoft_active_directory (Microsoft Active Directory)	host_name port log_dir	Fully Qualified Host Name AD Running Port Active Directory Log Directory
omc_microsoft_dnsserver (Microsoft DNS Server)	host_name system_root	Host Name System Root
omc_microsoft_dotnet_server (Microsoft .NET Server)	host_name system_root	Host Name System Root
omc_microsoft_exchange (Microsoft Exchange)	host_name exchange_home msg_tracking_log_home smtp_send_log_home smtp_receive_log_home	Host Name Application Installation Directory Message Tracking Log Location Send Connector Log Location Receive Connector Log Location
omc_microsoft_hyper_v_virtual_platform (Microsoft Hyper-V Virtual Platform)	host_name log_directory	Host Name Log Directory
omc_microsoft_hyper_v_virtual_server (Microsoft Hyper-V Virtual Server)	host_name log_directory	Host Name Log Directory
omc_microsoft_iis (Microsoft IIS)	system_drive site_id_folder	System Drive Site ID folder
omc_microsoft_iis_web_site (Microsoft Internet Information Services Web Site)	host_name home_dir http_port_list https_port_list base_log_dir site_id	Host Name Home Directory HTTP Port List HTTPS Port List Logging Directory Service Site ID
omc_microsoft_sharepoint (Microsoft SharePoint)	host_name common_program_files	Server Name Common Program File Directory

Entity Type (Display Name)	Property Names	Property Display Names
omc_mongodb (MongoDB)	host_name port install_home log_dir	Host Name Port Number MongoDB installation directory MongoDB log directory
omc_mysql_db (MySQL Database)	url jdbcdriver host_name database_name port is_cluster	URL JDBC Driver Server Name Database Name Port Number Is Cluster
omc_mysql_db_instance (MySQL Database Instance)	host_name instance_name data_dir	Server Name Instance Name Data Directory
omc_netapp_fas (NetApp FAS)	host_name	Host Name
omc_nginx Nginx	server_name listen_port install_home version main_error_log_dir http_error_log_dir stream_error_log_dir server_error_log_dir location_error_log_dir mail_error_log_dir http_access_log_dir server_access_log_dir location_access_log_dir	Server Name Nginx Listen Port Nginx install home Nginx version Nginx Main Context Error Log Location Nginx Http Context Error Log Location Nginx Stream Context Error Log Location Nginx Server Context Error Log Location Nginx Location Context Error Log Location Nginx Mail Context Error Log Location Nginx Http Context Access Log Location Nginx Server Context Access Log Location Nginx Location Context Access Log Location
omc_nodejs (Node.js)	host_name port_list full_module_name log_directory log_filename_pattern	Fully Qualified Host Name Port Module Name Log Directory Log File Name Pattern
omc_oc4j (OC4J)	host_name instance_name oracle_binary_home version oracle_home	Host Name OC4J Instance Name Oracle Binary Home Version of OC4J Oracle home path

Entity Type (Display Name)	Property Names	Property Display Names
omc_oracle_apache (Oracle HTTP Server)	host_name port ohs_home component_name protocol config_path oracle_home version	Host Name Listen Port Instance Home Location (11g) / Domain Home Location (12c) Component Name Protocol Configuration Path Oracle Home Path Version
omc_oracle_bi (ORACLE BI)	host_name obiee_home oracle_instance oracle_home	Host Name OBIEE Home Oracle Instance Oracle Home
omc_oracle_db_instance (Oracle Database Instance)	host_name instance_name audit_dest adr_home	Host Name Instance Name Audit Dest ADR Home
omc_oracle_db_listener (Oracle Database Listener)	hosted_vip_name host_name lsnr_port lsnr_protocol oracle_home lsnr_alias lsnr_ora_dir preferred_net_address log_dir_path trace_dir_path lsnr_version lsnr_type use_ssh	Host VIP Name Host Name Port Listener Protocol Oracle Home Alias Listener Ora Directory Preferred Net Address Log Directory Trace Directory Listener Version Listener Type Use SSH
omc_oracle_ebiz (Oracle E-business suite)	host_name version inst_top log_home appl_top oracle_home context_name	Host Name Version Instance Location Logs Home Directory Application Product Files Location Oracle Home Context Name
omc_oracle_hadoop_hdfs (Oracle Hadoop HDFS)	cm_url cm_target_name cm_cluster_name metric_url log_dir	Cloudera Manager URL Cloudera Manager Target Name Hadoop Cluster Name REST Fetchlet Base URL Log Directory

Entity Type (Display Name)	Property Names	Property Display Names
omc_oracle_hadoop_yarn (Oracle Hadoop Yarn)	cm_url cm_target_name cm_cluster_name metric_url log_dir remote_log_dir	Cloudera Manager URL Cloudera Manager Target Name Hadoop Cluster Name REST Fetchlet Base URL Log Directory Remote Log Directory
omc_oracle_oam (Oracle Access Management Server)	weblogic_home server_name app_name	Weblogic Home Server Name Application Name
omc_oracle_oid (Oracle Internet Directory)	ldap_bind_DN ldap_ssl_port oracle_home host UserName connect_descriptor canonical_path server_name service_url oracle_instance ias_internal_name	LDAP Bind DN LDAP SSL Port Oracle home path OID Host Username OID Connect Descriptor Canonical Path Server Name Middleware Administration Server Service URL Oracle Instance Directory Internet Directory Name
omc_oracle_otd_instance (Traffic Director Instance)	has_credentials config_name host_name otd_home version instance_name oracle_home version_category instance_id access_log_path server_log_path tcp_access_log_path	Has Credentials Configuration Name Host Name Instance Home Location (11g) / Domain Home Location (12c) Version Instance Name Oracle Home Version Category Instance Id Access Log Location Server Log Location TCP Access Log Location
omc_oracle_soainfra (SOA Infrastructure)	server_name version optimize_metric service_url	Server Name Version Optimize Metric Upload Service URL
omc_oracle_utilities_ccb (Oracle Utilities Customer Care and Billing)	host_name ouccb_servdir	Fully Qualified Host Name Oracle Utilities Customer Care and Billing Log Home Directory
omc_postgresql_db (PostgreSQL Database)	host_name port log_directory log_filename_prefix	Host Name Port Number Server Log Directory Server Log Filename Prefix

Entity Type (Display Name)	Property Names	Property Display Names
omc_rubyonrails (Ruby on Rails)	host_name application_path log_dir	Host Name Ruby on Rails Application path Ruby on Rails Application log directory
omc_siebel_component (Siebel Component)	siebel_component_name siebel_component_group_name siebel_component_internal_name siebel_component_group_internal_name siebel_component_log_dir siebel_component_log_file_name siebel_component_sarm_log_dir siebel_component_sarm_log_file_name	Siebel Component Siebel Component Group Name Siebel Component Internal Name Siebel Component Group Internal Name Siebel Component Log Directory Siebel Component Log File Name Siebel Component SARM Log Directory Siebel Component SARM Log File Name
omc_siebel_server (Siebel Server)	siebel_server_name siebel_server_install_dir siebel_server_version siebel_server_internal_name siebel_server_log_dir siebel_server_logfile_name	Siebel Server Name Siebel Server Install Directory Siebel Server Version Siebel Server Internal Name Siebel Server Log Directory Siebel Server Logfile Name
omc_sqlserver_db (Microsoft SQL Server Database)	url jdbcdriver host_name database_name port is_cluster	URL JDBC Driver Server Name Database Name Port Number Is Cluster
omc_sqlserver_db_instance (Microsoft SQL Server Database Instance)	host_name instance_name instance_root_dir	Server Name Instance Name Instance Root Directory
omc_sybase_ase_db (Sybase ASE Database)	server_name host_name port sybase_dir sybase_ase_dir	Server Name Host Name Port Number Sybase installation Directory Sybase ASE directory
omc_tibco (TIBCO)	machine_name bc_version engine_work_dir tibco_domain sharepath_dir tra_version	Machine Name BC Version Engine Work Directory TIBCO Domain Share Path Directory TRA Version

Entity Type (Display Name)	Property Names	Property Display Names
omc_tomcat (Tomcat)	host_name jmx_port jmx_user_name jmx_password jmx_protocol jmx_service ssl_trust_store ssl_trust_store_password version catalina_base_directory_path	Host Name JMX Port Number JMX User Name (required when authentication is enabled for JMX) JMX Password (required when authentication is enabled for JMX) Communication Protocol Service Name SSL Trust Store (required when SSL is enabled) SSL Trust Store JMXPassword (required when SSL is enabled) Apache Tomcat Version Catalina Base Directory Path
omc_vmware_virtual_platform (VMware Virtual Platform)	server_name all_users_profile	Server Name All Users Profile
omc_vmware_virtual_server (VMware Virtual Server)	host_name all_users_profile	Fully Qualified Host Name All Users Profile
omc_websphere_j2eeserver (IBM Websphere Server)	host_name http_port_list https_port_list server_name was_home profile_name	Host Name Websphere HTTP Port List Websphere HTTPS Port List Server Name Websphere Application Server Home Profile Name

H

Download and Customize Oracle Log Analytics JSONs

You can add entities by passing input JSONs to the `omcli` commands. Following are the list of entities whose sample JSONs you can download, edit, and use for adding those entities:

- Oracle Database
- Oracle Database Listener
- Oracle WebLogic Domain
- Oracle WebLogic Server
- Oracle HTTP Server
- Oracle Linux Host
- Oracle Access Manager
- Oracle Internet Directory
- Oracle Automated Storage Management
- Microsoft SQL Server Database
- MySQL Database
- IBM Websphere Server
- JBoss Server
- Tomcat Server
- Siebel Enterprise
- Docker Server
- Oracle Clusterware
- Oracle Peoplesoft

Download the set of JSON files provided for Oracle Log Analytics in this zip file:

[Link for downloading sample JSONs](#)



Note:

This zip file contains the JSON files for only a few entity types. For a complete list of entity types supported for monitoring, see Download and Customize Oracle Infrastructure Monitoring JSONs in *Using Oracle Infrastructure Monitoring*.

Tips for Editing JSON Files

- Replace all values that are listed in the file in quotation marks and within brackets (“<text>”) with values that apply to your entity.

- Use the multi-entity file to add multiple types of entities and be sure to remove all the extra entries that you're not using.
- Consider using a JSON validation tool.

Write Performant Extended Field Extraction Expression

The following tips will enable you to write performant Extended Field Extraction Expression:

- Ensure that the Extended Field Extraction Expression does not have a Match all regex (. * or \s*\S*) at the start or the end. The Extended Field Extraction Expression as in the example below is not allowed:

```
.*(?:[-]*)\s*Call\s+Stack\s+Trace\s*(?:[-]*){callstk:[\s\S]*?}(?:[-]*)\s*Binary.*
```

Use the following expression instead:

```
(?:[-]*)\s*Call\s+Stack\s+Trace\s*(?:[-]*){callstk:[\s\S]*?}(?:[-]*)\s*Binary
```

- Extended Field Extraction Expression must restrict the Match all regex (. * or \s*\S*) usage to 4. See the following example:

```
AVDFAlert.*EVENT\S+=\ (AN="\{sefAction:[^"]+\}"\s+AT="\{sefEndEventTime:[^"]+\}"\s+ASE="\{sevlvl:[^"]*\}"\s+URL="\{detailloc:[^"]*\}"\s+STN="\{sefSourceEPName:[^"]*\}"\s+STT="\{sefSourceEPType:[^"]*\}"\s+EN="\{eventid:[^"]*\}"\s+ET="\{sefStartEventTime:[^"]+\}"\s+ES="\{status:[^"]*\}"\s+CC="\{sefCommand:[^"]*\}"\s+UN="\{sefSourceEPAccountName:[^"]*\}"\s+CHN="\{sefActorEPName:[^"]*\}"\s+CIP="\{sefActorEPNwAddress:[^"]*\}"\s+TOBJ="\{eventtarget:[^"]*\}"\s+TTY="\{eventtargettype:[^"]*\}"\s+TS="\{sefSourceEPAccountSummaryRisk:[^"]*\}"\s
```

This expression uses . * 10 times, which is not allowed. Break the expression into multiple expressions to ensure that each expression uses . * up to 4 times.

- The Extended Field Extraction Expression does not use more than 4 conditions or alternatives. See the example below:

```
^\s*\S+\s+:\s+TTY=.*COMMAND=\s*\S*\/(cat|find|ls|more|tail|wc)\s+(-\w+\s+)\s+)?{msecrsrcname:\S+}
```

This expression is not allowed as it has 6 conditions. Break this expression into 2 expressions as follows, in which case each expression has 3 conditions:

```
^\s*\S+\s+:\s+TTY=.*COMMAND=\s*\S*\/(cat|find|ls)\s+(-\w+\s+)\s+)?{msecrsrcname:\S+}\s+:\s+TTY=.*COMMAND=\s*\S*\/(more|tail|wc)\s+(-\w+\s+)\s+)?{msecrsrcname:\S+}
```

- Extended Field Extraction Expression has some static text. See the following example:

```
(?:POST|PUT|DELETE) \s+["']* \s+(?:-)? (\d+)? \s+{contszin:\d+}
```

This expression does not have any static text. Ensure that the expression has at least some minimum static text, if not more.

If any of these rules are violated, the same would be flagged and would have to be fixed before the Extended Field Extraction Expression can be saved.

Once the **Extended Field Extraction Expression** is in accordance with the above rules, the **Test** functionality matches the **Example Base Field Content** with the expression and report on the match status. The match status can be a success, failure or an error. If its a failure or an error, fix the expression and re-test.

J

Write Performant Regular Expressions

Here we discuss some of the important aspects that you must consider while crafting performant regular expressions.

Character Classes

The character classes specify the characters that you're trying or not trying to match. Ensure to replace the `.` in your `.*s` with a more specific character. The `.*` will invariably shift to the end of your input and will then backtrack, that is return to a previously saved state to continue the search for a match. When using a specific character class, you have control over how many characters the `*` will cause the regex engine to consume, giving you the power to stop the rampant backtracking.

Consider the following example regular expression:

```
(\d{4}) (\d{2}) (\d{2}), (\S+), ([\S\s]*), ([\S\s]*), ([\S\s]*), ([\S\s]*),  
([\S\s]*), ([\S\s]*), ([\S\s]*), ([\S\s]*), ([\S\s]*), ([\S\s]*), ([0-9.]+),  
([0-9.]+), ([0-9.]+), ([0-9.]+), ([0-9.]+)
```

For the following input:

```
20150220,201502,16798186260,tvN,Entertainment/  
Music,Female,67,2,Individual,ollehtv Economic,Commercial  
Housing,5587,0,2,0,1,1
```

As the result of the specified regular expression, the match can run into backtracking. This situation is detected by Oracle Log Analytics and the match operation is aborted.

By changing the regular expression to the example below, you can ensure that the match completes faster. Notice that `[\S\s]*` is changed to `[^,]*` which avoids unnecessary backtracking.

```
(\d{4}) (\d{2}) (\d{2}), (\S+), ([^,]*), ([^,]*), ([^,]*), ([^,]*), ([^,]*), ([^,]*),  
([,]*), ([,]*), ([,]*), ([,]*), ([0-9.]+), ([0-9.]+), ([0-9.]+), ([0-9.]+),  
([0-9.]+), ([0-9.]+)
```

Lazy Quantifiers

In many regexes, greedy quantifiers (`.*s`) can be safely replaced by lazy quantifiers (`.*?s`), thus giving the regex a performance boost without changing the result.

Consider the input:

```
Trace file /u01/app/oracle/diag/rdbms/navisdb/NAVISDB/trace/  
NAVISDB_arc0_3941.trc  
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
ORACLE_HOME = /u01/app/oracle/product/11.2.0/db_1  
System name: Linux  
Node name: NAVISDB
```

```

Release: 2.6.18-308.e15
Version: #1 SMP Fri Jan 27 17:17:51 EST 2012
Machine: x86_64
Instance name: NAVISDB
Redo thread mounted by this instance: 1
Oracle process number: 21
Unix process pid: 3941, image: oracle@NAVISDB (ARC0)

```

Consider the following greedy regex for the given input:

```

Trace\sfile\s(\S*).*ORACLE_HOME\s*[:=]
\s*(\S*).*System\sname:\s*(\S*).*Node\sname:\s*(\S*).*Release:\s*(\S*).*
Machine:\s*(\S*).*Instance\sname:\s*(\S*).*Redo\sthread\smounted\sbys\
this\sinstance:\s(\d*).*Oracle\sprocess\snumber:\s*(\d*).*Unix\sprocess
\spid:\s(\d*).*image:\s+([\n\r]*)

```

The regex engine shoots to the end of the input every time it encounters `.*`. The first time that the `.*` appears, it consumes all the input and then backtracks until it gets to `ORACLE_HOME`. This is an inefficient way of matching. The alternative lazy regex is as shown below:

```

Trace\sfileRelease:\s*(\S*).*?Machine:\s*(\S*).*?
Instance\sname:\s*(\S*).*?
Redo\sthread\smounted\sbys\sthis\sinstance:\s(\d*).*?
Oracle\sprocess\snumber:\s*(\d*).*?Unix\sprocess\spid:\s(\d*).*?
image:\s+([\n\r]*)

```

The above lazy regex consumes starting from the beginning of the string until it reaches `ORACLE_HOME`, at which point it can proceed to match the rest of the string.

Note: If the `ORACLE_HOME` field appears toward the beginning of the input, the lazy quantifier should be used. If the `ORACLE_HOME` field appears toward the end, it might be appropriate to use the greedy quantifier.

Anchors

Anchors tell the regex engine that you intend the cursor to be in a particular place in the input. The most common anchors are `^` and `$`, indicating the beginning and end of the input.

Consider the following regexes to find an IPv4 address:

```
\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}
```

```
^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}
```

Notice that the second regex begins with `^` and is specific about the IP address appearing at the beginning of the input.

We're searching for the regex in input that looks like the following example:

```
107.21.20.1 - - [07/Dec/2012:18:55:53 -0500] "GET
    /extension/bsupport/design/cl/images/btn_letschat.png HTTP/1.1" 200
    2144
```

A non-matching input would look similar to the following example:

```
[07/Dec/2012:23:57:13 +0000] 1354924633 GET "/favicon.ico" "" HTTP/1.1 200
82726 "-"
    "ELB-HealthChecker/1.0"
```

The second regex (which starts with `^`) runs faster on the non-matching input because it discards the non-matching input immediately.

The Importance of Alternation

The order of alternation counts, so place the more common options in the front so they can be matched faster. If the rarer options are placed first, then the regex engine will waste time in checking those before checking the more common options which are likelier to succeed. Also, try to extract common patterns. For example, instead of `(abcd|abef)` use `ab(cd|ef)`.

Observe the following regexes:

```
{TIMEDATE}\s{0,1}:\s*(?:\|)(\w+)(?:\|)(?:\|)(\d+)(?:\|)(\.{1,1000}).*
```

```
{TIMEDATE}\s{0,1}:\s*(?:\|)(\w+)(?:\|)(?:\|)(\d+)(?:\|)(\.{1,1000}).*
```

On the following input:

```
2014-06-16 12:13:46.743: [UiServer][1166092608] {0:7:2} Done for
ctx=0x2aaab45d8330
```

The second regex matches faster as the alternation looks for character `[` first, followed by null. As the input has `[`, the match runs faster.

Sample Parse Expressions

You can refer to the following sample parse expressions to create a suitable parse expression for extracting values from your log file.

A log file comprises entries that are generated by concatenating multiple field values. You may not need to view all the field values for analyzing a log file of a particular format. Using a parser, you can extract the values from only those fields that you want to view.

A parser extracts fields from a log file based on the parse expression that you've defined. A parse expression is written in the form of a regular expression that defines a search pattern. In a parse expression, you enclose search patterns with parentheses `()`, for each matching field that you want to extract from a log entry. Any value that matches a search pattern that's outside the parentheses isn't extracted.

For the supported regex constructs, see [Java Regex Package Documentation](#).

Example 1

If you want to parse the following sample log entries:

```
Jun 20 15:19:29 hostabc rpc.gssd[2239]: ERROR: can't open clnt5aa9: No
such file or directory
Jul 29 11:26:28 hostabc kernel: FS-Cache: Loaded
Jul 29 11:26:28 hostxyz kernel: FS-Cache: Netfs 'nfs' registered for
caching
```

Following should be your parse expression:

```
(\S+)\s+(\d+)\s+(\d+):(\d+):(\d+)\s+(\S+)\s(?:\[^:\[\]]+)(?:\[(\d+
\)])?:\s+)?(.+)
```

In the preceding example, some of the values that the parse expression captures are:

- (\S+): Multiple non-whitespace characters for the month
- (\d+): Multiple non-whitespace characters for the day
- ([^\[\]]+): All the characters except [and] for the service name
- (.+): (Optional) Primary message content

Example 2

If you want to parse the following sample log entries:

```
###<Apr 27, 2014 4:01:42 AM PDT> <Info> <EJB> <host> <AdminServer>
<[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-
tuning) '> <OracleSystemUser> <BEA1-13E2AD6CAC583057A4BD>
<b3c34d62475d5b0b:6e1e6d7b:143df86ae85:-8000-000000000000cac6>
<1398596502577> <BEA-010227> <EJB Exception occurred during invocation
from home or business:
weblogic.ejb.container.internal.StatelessEJBHomeImpl@2f9ea244 threw
exception: javax.ejb.EJBException: what do i do: seems an odd quirk of
the EJB spec. The exception is:java.lang.StackOverflowError>
###<Jul 30, 2014 8:43:48 AM PDT> <Info> <RJVM> <example.com> <>
<Thread-9> <> <> <> <1406735028770> <BEA-000570> <Network
Configuration for Channel "AdminServer" Listen Address
example.com:7002 (SSL) Public Address N/A Http Enabled true Tunneling
Enabled false Outbound Enabled false Admin Traffic Enabled true
ResolveDNSName Enabled false>
```

Following should be your parse expression:

```
###<(\p{Upper}\p{Lower}{2})\s+(\[\d]{1,2}),\s+(\[\d]{4})\s+(\[\d]{1,2}):
(\[\d]{2}):(\[\d]{2})\s+(\\p{Upper}{2})(?:\s+(\w+))?
>\s+(\.*)>\s+(\.*)>\s+(\.*)>\s+(\.*)>\s+(\.*)>\s+(\.*)>\s+(\.*)>
\s+(\.*)>\s+<\d{10}\d{3}>\s+(\.*)>\s+(\.*) (?:\n(.*)?)?>\s*
```

In the preceding example, some of the values that the parse expression captures are:

- `(\p{Upper}\p{Lower}{2})`: 3-letter short name for the month; with the first letter in uppercase followed by two lowercase letters
- `([\d]{1,2})`: 1-or-2-digit day
- `([\d]{4})`: 4-digit year
- `([\d]{1,2})`: 1-or-2-digit hour
- `([\d]{2})`: 2-digit minute
- `([\d]{2})`: 2-digit second
- `(\p{Upper}{2})`: 2-letter AM/PM in uppercase
- `(?:\s+(\w+))`: (Optional, some entries may not return any value for this) Multiple alphanumeric characters for the time zone
- `(. *?)`: (Optional, some entries may not return any value for this) One or multiple characters for the severity level; in this case `<INFO>`
- `(. *)`: Any additional details along with the message

Search Patterns

Some of the commonly used patterns are explained in the following table:

Pattern	Description	Example
<code>.</code>	Any character except line break	<code>d.f</code> matches <code>def</code> , <code>daf</code> , <code>dbf</code> , and so on
<code>*</code>	Zero or more times	<code>D*E*F*</code> matches <code>DDEEFF</code> , <code>DEF</code> , <code>DDFF</code> , <code>EEFF</code> , and so on
<code>?</code>	Once or none; optional	<code>colou?r</code> matches both <code>colour</code> and <code>color</code>
<code>+</code>	One or more	<code>Stage \w-\w+</code> matches <code>Stage A-b1_1</code> , <code>Stage B-a2</code> , and so on
<code>{2}</code>	Exactly two times	<code>([\d]{2})</code> matches <code>01</code> , <code>11</code> , <code>21</code> , and so on
<code>{1,2}</code>	One to two times	<code>([\d]{1,2})</code> matches <code>1</code> , <code>12</code> , and so on
<code>{3,}</code>	Three or more times	<code>([\w]{3,})</code> matches <code>ten</code> , <code>hello</code> , <code>h2134</code> , and so on
<code>[...]</code>	One of the characters in the brackets	<code>[AEIOU]</code> matches one uppercase vowel
<code>[x-y]</code>	One of the characters in the range from x to y	<code>[A-Z]+</code> matches <code>ACT</code> , <code>ACTION</code> , <code>BAT</code> , and so on
<code>[^x]</code>	One character that is not x	<code>[^/d]{2}</code> matches <code>AA</code> , <code>BB</code> , <code>AC</code> , and so on
<code>[^x-y]</code>	One of the characters not in the range from x to y	<code>[^a-z]{2}</code> matches <code>A1</code> , <code>BB</code> , <code>B2</code> , and so on
<code>[d\D]</code>	One character that is a digit or a non-digit	<code>([\d\D]+)</code> matches any character, including new lines, which the regular dot doesn't match
<code>\s</code>	A whitespace	<code>(\S+)\s+(\d+)</code> matches <code>AA 123</code> , <code>a_ 221</code> , and so on
<code>\S</code>	One character that is not a whitespace	<code>(\S+)</code> matches <code>abcd</code> , <code>ABC</code> , <code>A1B2C3</code> , and so on
<code>\n</code>	A new line	<code>(\d)\n(\w)</code> matches: 1 A

Pattern	Description	Example
\w	An alphanumeric character	[\w-\w\w\w] matches a-123, 1-aaa, and so on
\p{Lower}	Lowercase letters	\p{Lower}{2} matches aa, ab, ac, bb, and so on
\p{Upper}	Uppercase letters	\p{Upper} matches A, B, C, and so on
\ followed by ?, [], *, .	Escape character; to use the characters after \ as literals	\? returns ?

K

Manually Specify Time Zone and Character Encoding for Files

You can manually specify the properties for the log files by editing the configuration properties in the agent installation folder.

You can perform this task for all the logs except the *Windows events*.

1. If you want to apply the properties on specific patterns and / or log sources, then make a note of the pattern ID and the log source ID. Open the config xml file under the agent installation folder *agent_inst/sysman/ApplicationsState/loganalytics/logrules_os_file.xml*.
 - Pattern ID example: `<Pattern id="495071102827757094" name="/tmp/w*.mgr" include="true">`
 - Log Source ID example: `<LogSource id="-2574377491167724513" name="SS Concurrent Manager Logs" sourceType="os_file"/>`
2. Edit the properties file *emd.properties* in the location *agent_inst/sysman/config/emd.properties* and add the following property to override the default configuration of Oracle Log Analytics:

```
loganalytics.src.override_config=true
```

3. Specify the time zone *tz* and character encoding *enc* properties in the file *emd.properties* by selecting from one of the following examples:

Note:

For the supported encoding values, see [Supported Encodings](#) in *Java Documentation*. Use the names listed under the column *Canonical Name for java.nio API*.

- Apply the properties for all the sources and patterns:

```
loganalytics.src.addl_src_ptn_configs=tz=UTC,enc=EUC-JP
```

- Apply the properties only for specific log sources:

```
loganalytics.src.addl_src_ptn_configs=srcid=-2574377491167724513,tz=UTC,enc=EUC-JP;srcid=-2574377491167724512,enc=UTF-8
```

In this example, the time zone `UTC` and character encoding `EUC-JP` properties are applied for log source `-2574377491167724513`, and character encoding property `UTF-8` is applied for log source `-2574377491167724512`.

- Apply the properties only for specific patterns:

```
loganalytics.src.addl_src_ptn_configs=ptnid=495071102827757094,tz=UTC,enc=EUC-JP;ptnid=495071102827757095,enc=UTF-8
```

In this example, the time zone `UTC` and character encoding `EUC-JP` properties are applied for pattern `495071102827757094`, and character encoding property `UTF-8` is applied for pattern `495071102827757095`.

- Apply the properties only for a combination of specific patterns and log sources:

```
loganalytics.src.addl_src_ptn_configs=srcid=-2574377491167724513,ptnid=495071102827757094,tz=UTC,enc=EUC-JP;srcid=-2574377491167724513,ptnid=495071102827757095,enc=UTF-8
```

In this example, the time zone `UTC` and character encoding `EUC-JP` properties are applied for logs with pattern `495071102827757094` and log source `-2574377491167724513`, and character encoding property `UTF-8` is applied for logs with pattern `495071102827757095` and log source `-2574377491167724513`.

L

Add Entity by Creating a JSON File

Say, you want to monitor and analyze the log files of an application that comprises an Oracle Database and an Oracle WebLogic Administration Server. Then you must add the entities pertaining to the application setup to be able to monitor them.

1. Download the sample JSONs pertaining to Oracle Database and Oracle WebLogic Server to use as templates. See [Download and Customize Oracle Log Analytics JSONs](#).
2. Edit the sample entity definition files with relevant values for your entities.

In the definition files, you must replace the text within the < > brackets (along with the angular brackets themselves) with the correct values.

For example, in the following replace <db name> with the name of the database instance, such as `macdbl` and <time zone> with the time zone value of the host, such as `PST`.

```
{
  "entities": [
    {
      "name": "<db name>",
      "type": "omc_oracle_db_instance",
      "displayName": "<db name>",
      "timezoneRegion": "<time zone>",
      .....
    }
  ]
}
```

3. Save the JSON file.
4. Run the following `omcli` command to add the entities from the <AGENT_BASE_DIR>/agent_inst/bin location:

```
<AGENT_BASE_DIR>/agent_inst/bin/omcli add_entity agent FILENAME
```

Where *FILENAME* is the name of the file that contains the entity definition to be added.

 **Note:**

- If your entities were already discovered in Oracle Infrastructure Monitoring, those entities will also be available for log collection, if:
 - The target properties for the entity have been defined correctly during discovery so that the log file locations can be resolved.
 - Correct associations (such as `omc_uses`) between the entity and the local host, and the cloud agent and the local host are present.

If the required properties are missing, to enable log collection, you must update the JSON (used at the time of adding these entities in Oracle Infrastructure Monitoring) with the required properties and run the `omcli update_entity` command. See `omcli` Command Options in *Working with Oracle Management Cloud*.
- If you want to add an entity for use with both Oracle Log Analytics and Oracle Infrastructure Monitoring, you should add the entity using credentials. See Add Entities Using JSON Files in *Using Oracle Infrastructure Monitoring*.
- To enable Infrastructure Monitoring for the `omc_oracle_db` entity that is already being used by Oracle Log Analytics, see Monitoring with Oracle Infrastructure Monitoring and Oracle Log Analytics in *Using Oracle Infrastructure Monitoring*.
- For the `omc_oracle_db` entity, if the format is specified as `"displayName: DB_Name"`, it must be changed to `"displayName: DB_Name/sid_name"`. This naming convention will help reduce issues with entity reconciliation if this entity is used later by Oracle Infrastructure Monitoring.