

NetWitness[®] Platform

Universal CloudWatch Logs Event Source Log Configuration Guide

Universal CloudWatch Logs

Last Modified: Tuesday, December 2, 2025

Event Source Product Information:

Vendor: [AWS](#)

Event Source: Amazon CloudWatch Logs

Versions: API v1.0

NetWitness Product Information:

Supported On: NetWitness Platform 12.2 and later

Event Source Log Parser: aws, aws_cloudtrail, aws_route53resolver, aws_windows, kubernetes, postgresql

Collection Method: Plugin Framework

Event Source Class.Subclass: Host.Cloud

Contact Information

NetWitness Community at <https://community.netwitness.com> contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA and other trademarks are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to <https://www.rsa.com/en-us/company/rsa-trademarks>. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates and are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on NetWitness Community. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Miscellaneous

This product, this software, the associated documentations as well as the contents are subject to NetWitness' standard Terms and Conditions in effect as of the issuance date of this documentation and which can be found at <https://www.netwitness.com/standard-form-agreements/>.

© 2024 RSA Security LLC or its affiliates. All Rights Reserved.

November 2024

Contents

Introduction	5
Setting up AWS CloudWatch Logs	6
Supported Event Types	7
Setting Up Amazon Universal CloudWatch Plugin	8
Deploying NetWitness Universal CloudWatch Configuration Files	8
Configure Universal CloudWatch plugin in NetWitness Suite UI	8
Amazon CloudWatch Collection Configuration Parameters	11
Basic Parameters	11
Advanced Parameters	13
Additional Information	15
Getting Help with NetWitness Platform	16
Self-Help Resources	16
Contact NetWitness Support	16
Feedback on Product Documentation	16

Introduction

The Amazon CloudWatch Logs service allows you to collect and store logs from your resources, applications, and services in near real-time. There are three main categories of logs:

- 1) **Vended Logs**- These are natively published by Amazon Web Services (AWS) on behalf of the customer. Currently, AWS supports Amazon Virtual Private Cloud (VPC) Flow and Amazon Route 53 logs.
- 2) **Logs Published by AWS Services**- Currently, over 30 AWS services publish logs to CloudWatch. These services include Amazon API Gateway, AWS Lambda, AWS CloudTrail, and others.
- 3) **Custom Logs**- These are logs from your own application and on-premise resources.

For more information, see

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html>

In NetWitness Universal CloudWatch logs integration, AWS logs or external application logs must be forwarded to your AWS CloudWatch log group before configuring the NetWitness plugin. You can then configure the NetWitness plugin using CloudWatch log group access details. It collects logs from CloudWatch log group through AWS API calls. The route of these logs before being parsed in NetWitness is:

Source → AWS CloudWatch Log group → NetWitness Universal CloudWatch Plugin → NetWitness AWS JSON log parsers/NetWitness user custom log parsers.

Note: For information on supported event types, see [Supported Event Types](#). You can still forward any type of event source logs from your CloudWatch log group to NetWitness Universal CloudWatch plugin and parse it either by creating a custom parser of your own or open a case with to add support for parsing.

Setting up AWS CloudWatch Logs

AWS Regions: Login to your Amazon Web Service (AWS) account and select a CloudWatch supported AWS region before configuring the CloudWatch Logs.

Note: Select the region closest to your country. Refer https://docs.aws.amazon.com/general/latest/gr/cwl_region.html. Note the AWS Region code of your CloudWatch log group as it is required when you configure the NetWitness Universal CloudWatch Plugin.

Programmatic Access: You need the access key and the secret key to collect AWS CloudWatch logs through Application Programming Interface (API) calls in NetWitness Platform.

Note: As the access key provides access to your AWS services, it should be kept secure. To know more about how to get access to your AWS account, refer https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html.

AWS Access Key Permissions: This is a one-time configuration as once you configure the access key to read logs from CloudWatch, it works for all AWS service Logs collected in the CloudWatch log group. You need read-only access to read logs from CloudWatch Logs using user programmatic access.

For information on accessing CloudWatch logs, see '[Allow Read-Only Access to CloudWatch Logs](#)'.

For information on embedding an inline policy see [Amazon AWS Identity and Access Management User Guide](#).

Add the following policy to your user role to allow a read-only access to CloudWatch Logs data.

```
{
"Version":"2012-10-17",
"Statement":[
{
"Action":[
"logs:Describe*",
"logs:Get*"
],
"Effect":"Allow",
"Resource":"*"
}
]
}
```

Supported Event Types

NetWitness sSupporte d Event Source	Steps to route Logs to AWS CloudWatch	Comments
AWS CloudTrail	https://docs.amazonaws.cn/en_us/awscloudtrail/latest/userguide/send-cloudtrail-events-to-cloudwatch-logs.html	
AWS VPC Flow Logs	https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-cwl.html	Select AWS default format in VPC flow logs configuration in AWS.
AWS Route 53	https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/logging-monitoring.html	
AWS AD Logs	https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_enable_log_forwarding.html	
AWS Windows Logs	https://docs.aws.amazon.com/kinesis-agent-windows/latest/userguide/what-is-kinesis-agent-windows.html	In the sink configuration the SinkType should be CloudWatchLogs and the Format should be xml.
Amazon EKS Control Plane logs	https://docs.aws.amazon.com/eks/latest/userguide/control-plane-logs.html	
Amazon RDS for postgresql	https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_LogAccess.Concepts.PostgreSQL.html#USER_LogAccess.Concepts.PostgreSQL.PublishtoCloudWatchLogs	Log Group Name should have /postgresql in the path e.g.: /aws/rds/instance/database-1/ postgresql ,Add to RDS Parameter group log_statement=all,log_destination=jsonlog

Setting Up Amazon Universal CloudWatch Plugin

To set up Amazon Universal CloudWatch plugin in RSA NetWitness Suite, perform the following tasks:

- Deploy NetWitness Universal CloudWatch files from NetWitness Live.
- Configure Universal CloudWatch plugin in NetWitness Suite UI.

Deploying NetWitness Universal CloudWatch Configuration Files

Universal CloudWatch Logs plugin requires resources available in NetWitness Live in order to collect logs and parse it.

To Deploy the amazoncloudwatch Content from Live:

1. In the **NetWitness Suite** menu, select **Live**. Browse Live for Universal CloudWatch Logs plugin by typing *amazoncloudwatch* into the Keywords text box and then click **Search**.
2. Select the item returned from the Search.
3. Click **Deploy** to deploy the **Universal CloudWatch** plugin to the appropriate Log Collectors, using the Deployment Wizard.
4. Log Parsers '**aws, aws_cloudtrail, aws_route53resolver, aws_windows, kubernetes, postgresql**' have been added as required resources of amazoncloudwatch in NetWitness Live. Deploy these parsers to appropriate Log Decoders when you deploy plugin log collection file.

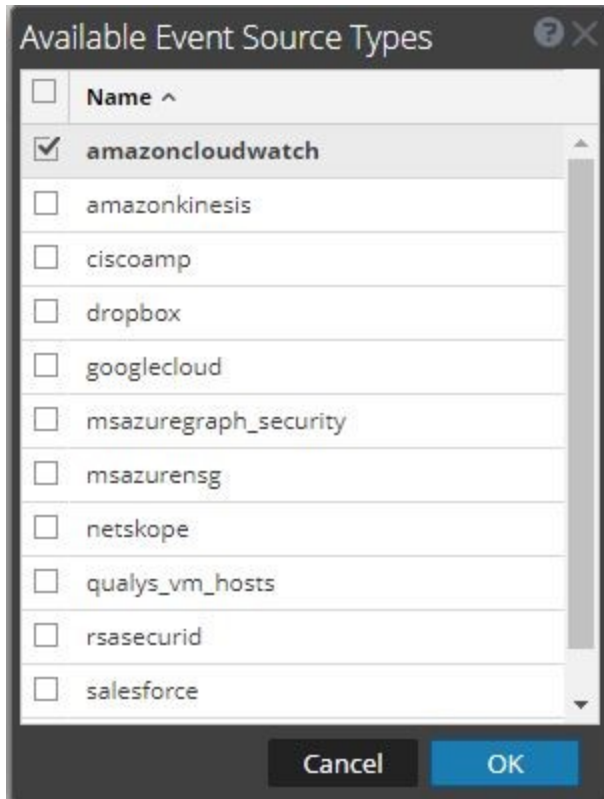
For more details, please refer the [Add or Update Supported Event Source Log Parsers](#) topic, or the *Live Resource Guide* on NetWitness community Link.

Note: On a hybrid installation, you need to deploy the package on both the VLC and the LC. Also, NetWitness aws JSON parsers will be enabled in your LD automatically after deployment.

Configure Universal CloudWatch plugin in NetWitness Suite UI

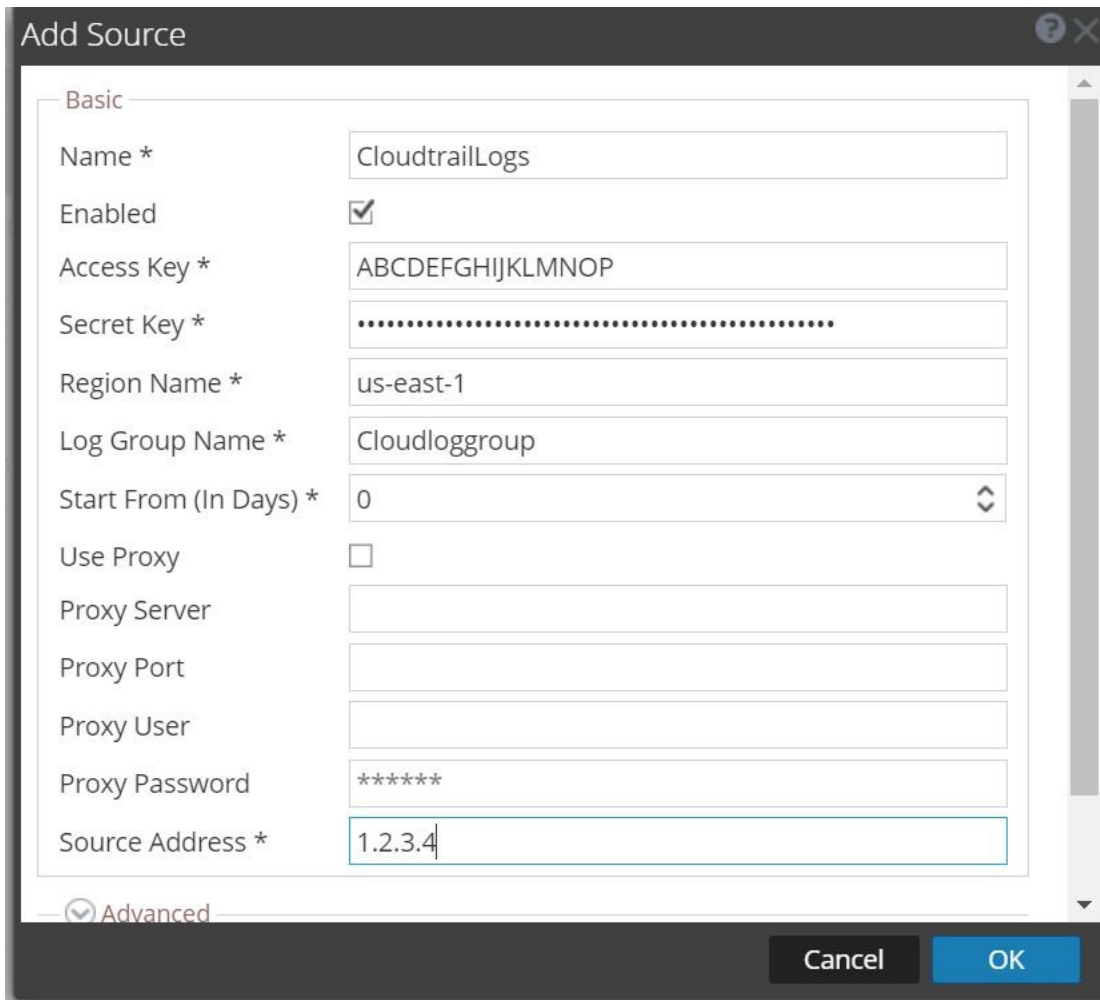
1. In **NetWitness Suite** menu, go to **ADMIN > Services**.
2. In the **Services** grid, select a Log Collector Service, and choose **Config** option from the **System** menu.
3. In the **Event Sources** tab, select **plugins** from the dropdown menu. The **Event Categories** panel displays the file event sources that are configured, if any.
4. In the **Event Categories** panel tool bar, click **+**.

The **Available Event Source Types** dialog is displayed.



Note: Skip this step and go to step 6 directly, if you have already configured an event source using Universal CloudWatch plugin.

5. Select **amazoncloudwatch** from the list and click **OK**. The newly added event source type is displayed in the **Event Categories** panel.
6. Select the **New Type** in the **Event Categories** panel and click **+**, in the **Source** panel tool bar, the **Add Source** dialog is displayed.



7. Define parameter values, as described in Amazon CloudWatch Collection Configuration.
8. Once you click on **Test Connection**, the result of the test will be displayed in the dialog box. If the test is not successful, edit the device or service information based on error messages displayed and retry.

Note: The log collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out and RSA NetWitness Suite displays a **Request Timed Out** Error.

9. If the test is successful, click **OK**. The new event source will be displayed in the **Sources** panel.
10. Repeat steps 6–9 to add another instance of Amazon CloudWatch plugin type.

Amazon CloudWatch Collection Configuration

Parameters

This section describes the Amazon CloudWatch plugin configuration parameters.

Basic Parameters

Name	Description
Name*	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on the screen.
Enabled	Select the checkbox to enable the event source configuration to start collection. The checkbox is selected by default.
Access Key*	AWS IAM Access Key belonging to your AWS user account.
Secret Key*	AWS Secret Key corresponding to the above access key.
Region Name*	AWS Region code where the CloudWatch log group is located. Example: us-east-1.

Name	Description
Log Group Name*	Input the AWS CloudWatch LogGroup Name here.
Start From (In Days)	<p>'N' number of days to backtrack and pull data from the Log Group. Default value is 0 which collects logs of last 30 mins of current day. Max supported value is 180 days.</p> <div data-bbox="1224 831 1419 1520" style="border: 1px solid green; padding: 5px;"> <p>Note: Do not edit the "Start From" value of a running instance. Please create a new event source instance for new "Start From" values because this is used to keep track of logs collection time and bookmark it in plugin to avoid duplicate logs collection.</p> </div>
Use Proxy	Check to enable proxy configuration.
Proxy Server	If you are using a proxy, enter the proxy server address.

Name	Description
Proxy Port	Enter the proxy port.
Proxy User	Username for the proxy (leave empty if using anonymous proxy).
Proxy Password	Password for the proxy (leave empty if using anonymous proxy).
Source Address	IP address that is to be given for the current plugin event source. You can use any valid IP value. This IP will be mapped to device.ip meta in NetWitness aws JSON parser and will be useful in indexing of logs.
Test Connection	Checks the configuration parameters specified in this dialog to make sure that those are correct to connect to the AWS Cloudwatch Logstream.

Advanced Parameters

Click **Advanced** to view and edit the advanced parameters, if necessary.

Name	Description
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is 180. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, the collector waits for that cycle to finish. If you have many event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Duration Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit. The default is set to 600.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum Idle time, in seconds, of a polling cycle. A zero value indicates no limit.
Command Args	Optional arguments to be added to the script invocation.
Debug	<p>Caution: Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem.</p> <p>Caution: Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: Off = (default) disabled On = enabled Verbose = enabled in verbose mode - adds thread information and source context information to the messages.</p> <p>This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). The debug logging is verbose, so limit the number of event sources to minimize performance impact.</p> <div style="border: 1px solid green; padding: 5px;"> <p>Note: If you enable debug, it adds a few more debug info from CloudWatch logs API as extra fields in NetWitness logs. Extra debug fields are <code>nw.eventTimestamp</code>, <code>nw.ingestionTime</code> nw.eventTimestamp – The time the event occurred, expressed as the number of milliseconds after Jan 1, 1970 00:00:00 UTC. nw.ingestionTime- The time the event was ingested to CloudWatch, expressed as the number of milliseconds after Jan 1, 1970 00:00:00 UTC.</p> </div>
Trail By (In Minutes)	<p>Specifies the lag between current time and log collection in RSA NetWitness Plugin. The default value is 60 minutes. Range is set between 5 to 120 minutes. You need to tune this value if you are seeing logs which are near to real time is missing in RSA NetWitness Investigation page.</p> <div style="border: 1px solid green; padding: 5px;"> <p>Note: It was observed that some CloudWatch logs take a while to be available in AWS log group. If you collect too close to the current time, there is a possibility of missing some events. Please refer following link for more details: https://docs.aws.amazon.com/AmazonCloudWatchLogs/latest/APIReference/API_LogStream.html#CWL-Type-LogStream-lastEventTimestamp</p> </div>

Name	Description
AWS Event Type	Optional Parameter. Default value is 'NA'. This field should be filled only when the user wants to route logs belonging to AWS LogGroup to a custom log parser of their own in NetWitness. Filled value will come as one of fields in header of your log in the NetWitness log decoder after collection. If you do not give any value here and event source is supported one, logs will be parsed automatically using NetWitness supported log parsers given in the page 1 of this doc.
AWS Endpoint	Optional Parameter. Default value is 'NA'. An endpoint is the URL of the entry point for an AWS service. An endpoint should be provided when the customer wants to connect privately to the AWS services. By default the AWS uses default endpoint for each service in an AWS Region. Example: protocol://service-code.region-code.amazonaws.com
SSL Enable	Uncheck to disable SSL certificate verification which is not suggested by NetWitness.
HTTPS Proxy	If Proxy server is configured, enable or disable based on the proxy traffic allowed. By default HTTPS Protocol is enabled for the Proxy connection. If the only HTTP traffic allowed via proxy, then uncheck this parameter.
Number of Threads	Tune the number of threads used in logcollector execution if required. This helps to control the throttling exception due to multiple API calls in short time in your AWS environment. Default value is 5 and you can put values between 1 and 5.

Additional Information

Following information is used to parse logs in NetWitness supported log parsers into metas. We expect the default aws format for logs from AWS server to parse logs in NetWitness log parsers.

Logs Formats

vpcflowlogs	https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-records-examples.html
cloudtrail	https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-examples.html
route 53	a) https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/query-logs.html#query-logs-format b) https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver-query-logs-example-json.html
kube-apiserver	https://kubernetes.io/docs/reference/config-api/apiserver-audit.v1/

Getting Help with NetWitness Platform

Self-Help Resources

There are several options that provide you with help as you need it for installing and using NetWitness:

- See the documentation for all aspects of NetWitness here: <https://community.netwitness.com/s/netwitness-platform-documentation>
- Use the **Search** and **Create a Post** fields in NetWitness Community portal to find specific information here: <https://community.netwitness.com/s/netwitness-discussions>
- See the NetWitness Knowledge Base: <https://community.netwitness.com/s/knowledge-base>
- See Troubleshooting section in the guides.
- See also <https://community.netwitness.com/s/netwitness-blog>.
- If you need further assistance, contact NetWitness Support.

Contact NetWitness Support

If you contact NetWitness Support, you should be at your computer. Be prepared to provide the following information:

- The version number of the NetWitness Platform product or application you are using.
- The type of hardware you are using.

Use the following contact information if you have any questions or need assistance.

NetWitness Community Portal	https://community.netwitness.com In the main menu, click Support > View My Cases .
International Contacts (How to Contact NetWitness Support)	https://community.netwitness.com/s/contactsupport
Community	https://community.netwitness.com/s/netwitness-discussions
NW Update	https://update.netwitness.com/
LiveUI	https://live.netwitness.com

Feedback on Product Documentation

You can send an email to feedbacknwdocs@netwitness.com to provide feedback on NetWitness Platform documentation.