
Micro Focus Security ArcSight SmartConnectors

SmartConnector for Amazon Web Services CloudWatch Configuration Guide

Document Release Date: December 3, 2020

Software Release Date: December 3, 2020



Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

<https://www.microfocus.com>

Copyright Notice

© Copyright 2020 Micro Focus or one of its affiliates

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, "commercial computer software" is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation ("FAR") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("DOD"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement ("DFARS") and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

[ArcSight Product Documentation on the Micro Focus Security Community](#)

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Contents

Introduction	6
VPC Events	6
Understanding Data Collection	7
Certified Platforms for AWS CloudWatch Deployment	8
AWS Cloudwatch Installation Procedure	8
Prerequisites	10
Generating AWS CloudFormation Stack	16
Installing a Syslog NG Daemon SmartConnector	16
Manually Configuring AWS CloudFormation Stack Network	17
Opening Ports	18
Deploying the Connector	20
AWS Credentials Configuration	20
Deployment	20
Updating the Connector	21
Post-Deployment Configurations	22
Running Lambda Functions in High Availability	22
Removing Lambda's Subnet Warning	22
Upgrading the Connector	24
Undeploying the Connector	24
Configuring the Load Balancer as a Destination	25
Configuring Route 53	25
Query Logging	25
Using Amazon CloudWatch to Access DNS Query Logs	26
To Configure Query Logging	26
To Stop Query Logging:	28
Configuring AWS CloudHSM	28
Monitoring AWS CloudHSM Audit Logs	29
Interpreting HSM Audit Logs	29
Audit Log Reference	30
Configuring AWS CloudHSM Audit Logs	30
Manage Clusters and HSMs	30
Manage Users	30
Preparing to run cloudhsm_mgmt_util	31
Manage Keys	31
Set Up key_mgmt_util	32

Helper Tools	32
Overriding Parser Files	32
Updating Parser Files	34
Apply Monthly Parser Updates from Micro Focus	34
Apply the Parser Updates from a Support Team or Other Sources	35
Send Documentation Feedback	37

Introduction

The Amazon CloudWatch Connector helps to gather all the event logs generated inside a specific VPC, normalizes the events to Common Event Format (CEF), and proceeds to send the data to an ArcSight's destination.

Amazon's Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

VPC Events

A VPC event indicates a change in your AWS environment. AWS resources can generate events when their state changes. For example, Amazon EC2 generates an event when the state of an EC2 changes from pending to running, and Amazon EC2 Auto Scaling generates events when it launches or terminates instances. AWS CloudTrail publishes events when you make API calls. You can set up scheduled events that are generated on a periodic basis.

For a list of services that generate events, and sample events from each service, see the [AWS documentation](#).

Related AWS Services

The following services are used in conjunction with CloudWatch Events:

AWS CloudFormation enables you to model and set up your AWS resources. You create a template that describes the AWS resources you want, and AWS CloudFormation takes care of provisioning and configuring those resources for you. You can use CloudWatch Events rules in your AWS CloudFormation templates. For more information, see the [AWS documentation](#).

AWS Identity and Access Management (IAM) helps you securely control access to AWS resources for your users. Use IAM to control who can use your AWS resources (authentication), what resources they can use, and how they can use them (authorization). For more information, see the [AWS documentation](#).

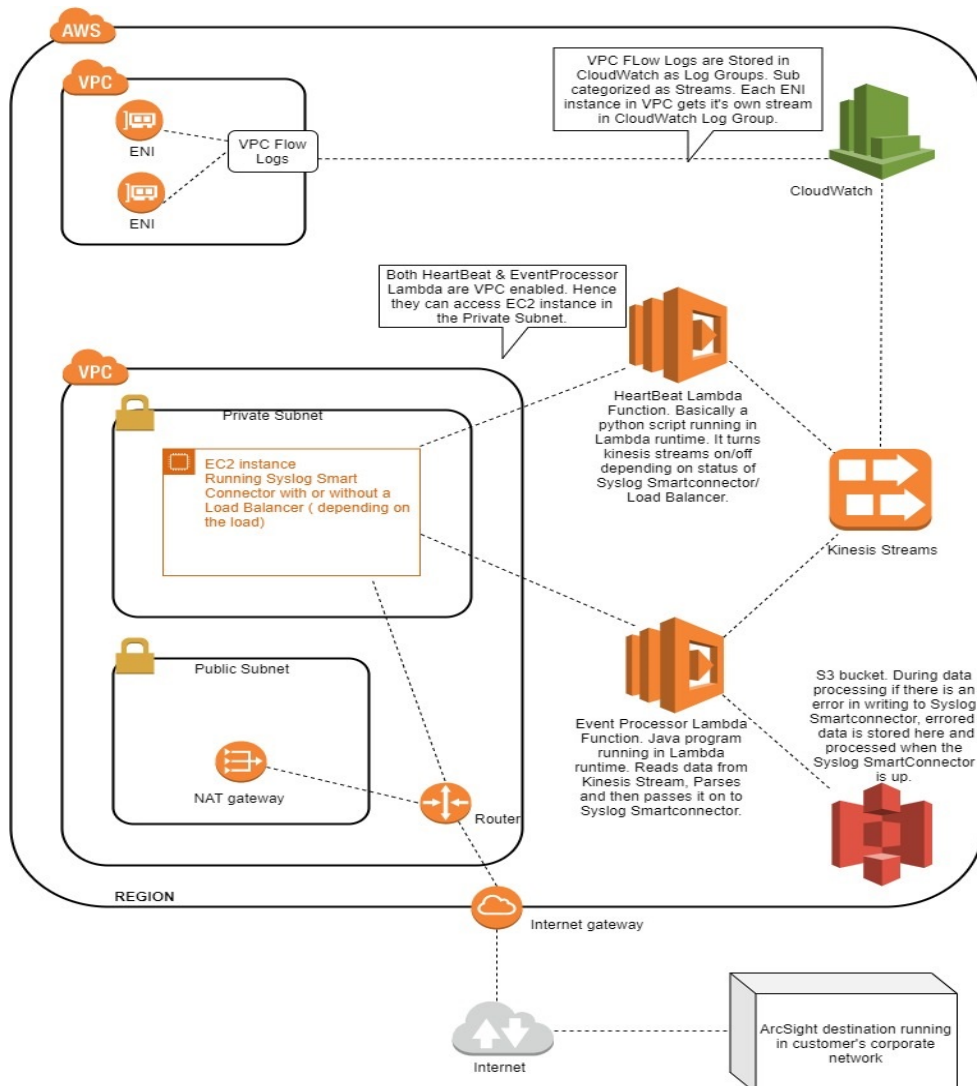
Amazon Kinesis Data Streams enables rapid and nearly continuous data intake and aggregation. The type of data used includes IT infrastructure log data, application logs, social media, market data feeds, and web clickstream data. Because the response time for the data intake and processing is in real time, processing is typically lightweight. For more information, see the [AWS documentation](#).

AWS Lambda enables you to build applications that respond quickly to new information. Upload your application code as Lambda functions and Lambda runs your code on high-availability compute infrastructure. Lambda performs all the administration of the compute resources, including server and operating system maintenance, capacity provisioning, automatic scaling, code and security patch deployment, and code monitoring and logging. For more information, see the [AWS documentation](#).

S3 is cloud storage for the internet. To upload your data (photos, videos, documents etc.), you first create a bucket in one of the AWS Regions. You can then upload any number of objects to the bucket. For more information, see the [AWS Documentation](#).

Understanding Data Collection

The following diagram provides a high-level overview of how the AWS CloudWatch Connector collects and sends VPC Flow Logs to an ArcSight destination.



Note: The image only shows VPC Flow Logs. However, other types of logs are supported, such as Route 53 logs.

The AWS CloudWatch Connector gathers events from a specific VPC.

- **AWS VPC Logs**

- **VPC Flow Logs:** These are network traffic logs (traffic in & out) associated with each ENI (Elastic Network Interface) instance within a VPC. Once created, it appears in the Flow Logs tab of the VPC dashboard.
- **Cloud Watch Logs** The VPC Flow Log created above, is stored as a Log Group in CloudWatch. Each Log Group is created with a delay of approximately 15 minutes after the corresponding VPC Flow Log is created and it can be accessed through a CloudWatch Logs dashboard. Each Log Group contains a separate stream for each ENI (Elastic Network Interface) instance.

The events continue flowing from the Cloud Watch Logs log group to Kinesis Data Stream (subscription).

The events flow from Kinesis to Lambda Event Processor for parsing, in case of communication issues with the ArcSight destination (SmartConnector or Load Balancer), the Lambda Event Processor backs up the events to S3.

Once the events are parsed and communication is established with the ArcSight receiver (Syslog NG Daemon SmartConnector or the ArcSight Load Balancer), the events are sent to either of them and finally to a different destination, for example: ArcSight Logger, ESM, CEF file, etc.

Lambda Functions

Lambda Monitoring continues to monitor the status of the destination (like a cron) and disables/enables the Kinesis subscription to Lambda Event Processing, based on whether the destination is accessible or not.

Lambda Event Processing is triggered automatically along with a batch of events, whenever, new events are available. The function decodes the batch, loads it into the memory in plain text, parses and processes the events contained in that object.

Certified Platforms for AWS CloudWatch Deployment

The installer requires an EC2 instance with Amazon Linux 2.

AWS Cloudwatch Installation Procedure

["Prerequisites" on page 10](#)

["Generating AWS CloudFormation Stack" on page 16](#)

["Installing a Syslog NG Daemon SmartConnector" on page 16](#)

["Manually Configuring AWS CloudFormation Stack Network" on page 17](#)

Prerequisites

Make sure to meet the following pre-requisites prior the deployment.

1. Create a New VPC in AWS.
2. Create a public subnet with an EC2 instance within the VPC created in step 1.
3. Create a private subnet. For more information on how to create a public and private subnet, see ["Configuring an Amazon Virtual Private Cloud \(VPC\) and Subnets " on page 17](#)
4. Ensure, you are able to connect from an EC2 instance in a public subnet to an EC2 instance in private subnet.
5. Generate an IAM user with the required privileges.
This user is used when running the CloudWatch installer in the following step. For more information on how to create an IAM user, see ["AWS Account Setup " on page 12](#)
6. Create or select a S3 bucket.
The location must be the one that the AWS CloudWatch Connector uses to complete the deployment. The external.properties file, the ArcSight SmartConnector or the Load Balancer certificate, and the eventprocessor.jar are stored in your S3 bucket. To create a S3 bucket, see, [AWS Documentation](#).
7. Upload your Syslog NG Daemon Connector or Load Balancer certificate to your S3 bucket.
The certificate is located in the installation folder of the SmartConnector or the Load Balancer. For connectors, the certificate is in \$INSTALLATIONFOLDER/current/user/agent/remote_management.p12. For Load Balancer, the certificate is in \$INSTALLATIONFOLDER/current/user/loadbalancer/loadbalancer.p12.
8. Create or select a folder on the selected bucket to store fault tolerance files.
9. Create a properties file and upload it to the S3 bucket. The location of this file is required for deployment. Follow the template to create the file:

```
##
## External properties file to upload to s3
##
#
# Valid properties
#
#Arcsight connector host name or ip address, required parameter
host.name=0.0.0.0
```

```

#
#Arcsight connector port number, required parameter
port.number=9999
#
#Arcsight connector certificate bucket name in s3, required parameter
certificate.bucket=bucket_name
#
#Arcsight connector certificate key location in s3, required parameter
certificate.key=path/to/file
#
#Arcsight connector certificate password, required parameter
certificate.password=password
#
#Number of retries if the destination does not respond,
#if de destination stills without responding
#the mechanism of transport.cache will be activated
send.retries=3
#
#Arcsight connector transport cache bucket name in s3, required parameter
transport.cache.bucket=bucket_name
#
#Arcsight connector transport cache location in s3, required parameter
transport.cache.directory=path/to/file
#
#Log Level changes the log level to the specified level
#value can be any of: info debug error all warn fatal "trace" "off" or
#case insensitive value
log.level=info debug error all warn fatal trace off

```

The port must be same used for the ArcSight Syslog NG Daemon SmartConnector or the ArcSight Load Balancer. This value is also required during deployment.

The path of the certificate and its password uploaded in step 3 must be added to the [properties.file]. The file should look like this:

```

host.name=18.235.121.137
port.number=1514
certificate.bucket=vlc-s3-bucket
certificate.key=cert/remote_management.p12
certificate.password=changeit
send.retries=3

```

```
transport.cache.bucket=vlc-s3-bucket
transport.cache.directory=transport.cache
log.level=info
```

AWS Account Setup

In order to install the AWS Connector you require an AWS account, for more information, see [AWS Documentation](#). Once given a main AWS account, you may use with your AWS CloudWatch Connector. Or, you may log in to AWS with your main AWS account, create a new user, and give this user privileges to deploy and work with the AWS CloudWatch Connector.

1. Create a new policy.
2. From the AWS Dashboard, select **AWS Cloudwatch Connector**.
3. Select **Policies**.
4. Select **Create Policy**.
5. Go to the **JSON** tab.

From the JSON editor, paste the following JSON document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:UpdateStack",
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:DescribeStackResources"],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/*/*"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStacks"],
```

```

    "Resource": "*"
  }
]
}

```

6. Select **Review Policy**.

7. Save the new Policy.

- a. On the field name, enter CloudFormationBasicExecution.
- b. Select **Create Policy**.
- c. If created correctly, the message "CloudFormation Basic Execution has been created" is displayed.

8. Create a new policy from the AWS Dashboard.

- a. Select **Policies**.
- b. Select **Create Policy**
- c. Go to the **JSON** tab.

From the JSON editor, paste the following JSON document:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:AddPermission",
        "lambda>DeleteFunction",
        "lambda:UpdateFunctionCode",
        "lambda>DeleteEventSourceMapping",
        "lambda:InvokeAsync",
        "lambda:UpdateFunctionConfiguration",
        "lambda:UpdateEventSourceMapping",
        "lambda:RemovePermission",
        "lambda:InvokeFunction",
        "lambda:CreateEventSourceMapping",
        "iam:PutRolePolicy",
        "iam:AddRoleToInstanceProfile",
        "iam:PutGroupPolicy",
        "iam>DeleteRole",

```

```
"iam:PassRole",
"iam:DeleteRolePolicy",
"iam:CreatePolicy",
"iam:AttachGroupPolicy",
"iam:PutUserPolicy",
"iam:DeleteGroup",
"iam:CreateRole",
"iam:AttachRolePolicy",
"iam:CreateGroup",
"iam:AttachUserPolicy",
"iam:AddUserToGroup",
"logs:DeleteSubscriptionFilter",
"logs:PutSubscriptionFilter",
"logs:DeleteLogStream",
"logs:CreateLogStream",
"logs:CreateLogGroup",
"logs:DeleteLogGroup",
"logs:PutLogEvents",
"logs:PutRetentionPolicy",
"kinesis:PutRecords",
"kinesis:RemoveTagsFromStream",
"kinesis:AddTagsToStream",
"kinesis:CreateStream",
"kinesis:PutRecord",
"kinesis:DeleteStream",
"ec2:DeleteRouteTable",
"ec2:CreateNetworkInterfacePermission",
"ec2:CreateRoute",
"ec2:RevokeSecurityGroupEgress",
"ec2:CreateTags",
"ec2:DisassociateRouteTable",
"ec2:CreateNetworkInterface",
"ec2:DescribeSecurityGroups",
"ec2:DeleteFlowLogs",
"ec2:AssociateRouteTable",
"ec2:CreateTrafficMirrorFilterRule",
"ec2:DeleteNetworkInterface",
"ec2:CreateRouteTable",
```

```

"ec2:CreateFlowLogs",
"ec2:CreateTrafficMirrorFilter",
"ec2:DeleteTags",
"ec2:DescribeNetworkInterfaces",
"ec2:CreateSecurityGroup",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:DetachNetworkInterface",
"ec2:DeleteRoute",
"ec2:DeleteSecurityGroup",
"events:RemoveTargets",
"events:PutRule",
"events:DeleteRule",
"events:PutTargets",
"s3:DeleteObject",
"s3:PutObject"
],
"Resource": "*"
}
]
}

```

- d. Select Review Policy.
- e. Save the new Policy.
- f. On the field name, enter **LimitedPermission**.
- g. Select **Create Policy**.
- h. If created correctly, the message "LimitedPermission has been created" is displayed.
- i. Create a new user.
 - i. On the AWS Dashboard, select **IAM**.
 - ii. Select **Users**.
 - iii. Select **Add User**.
- j. On **User Name**, enter the new user name.
- k. Go to **Access Type** and select **Programmatic Access** and **AWS Management Console Access**.
- l. Choose your Console password.
 - i. Select **Auto Generated Password** and AWS creates a random password.
 - ii. Select **Custom password** and choose your custom password.
It must adhere to the account password policy.

- iii. Users can change the password on the first time login. For more information, see, [Require Password Reset](#).
- m. Select **Next**.
You are taken to **Permissions**.
- n. Select **Attach existing policies directly**.
- o. Click the checkboxes of the following policies:
 - i. CloudFormationBasicExecution created above, and the following AWS managed policies
 - ii. LimitedPermission created above, and the following AWS managed policies
 - iii. AmazonEC2ReadOnlyAccess
 - iv. CloudWatchEventsReadOnlyAccess
 - v. AmazonKinesisReadOnlyAccess
 - vi. AmazonS3ReadOnlyAccess
 - vii. CloudWatchLogsReadOnlyAccess
 - viii. IAMReadOnlyAccess
 - ix. AWSLambdaReadOnlyAccess
- p. Go to **Tags < Review < Create User**.
A message is displayed, indicating the user creation was successful.
- q. Write down the **Access Key ID** and the **Secret Access Key**. They are required for deployment.

Generating AWS CloudFormation Stack

To generate AWS CloudFormation Stack:

1. Copy the CloudWatch installer to the EC2 instance in a private subnet (Via SSH through a public subnet).
2. Run the installer to generate the AWS CloudFormation stack. For more information on how to deploy the connector , see ["Deploying the Connector" on page 20](#).

Installing a Syslog NG Daemon SmartConnector

To install a Syslog NG Daemon SmartConnector

1. Copy the Syslog NG Daemon installer to an EC2 instance in private a subnet (via SSH through a public subnet).
2. Install the SyslogNGDaemon.

Manually Configuring AWS CloudFormation Stack Network

To enable a Lambda VPC:

1. From VPC Connection, select Custom VPC.
2. Choose the VPC and subnet created while following the ["Prerequisites" on page 10](#) steps.
3. From Security Group, select the Lambda Security Group created while following the ["Generating AWS CloudFormation Stack" on the previous page](#) steps and click save.

The screenshot shows the AWS Lambda console interface for configuring a VPC connection. The left sidebar shows the navigation menu with 'Functions' selected. The main content area is titled 'VPC' and contains the following sections:

- Information:** A blue box with an information icon stating: "When you connect a function to a VPC in your account, it does not have access to the internet unless your VPC provides access. To give your function access to the internet, route outbound traffic to a NAT gateway in a public subnet. [Learn more](#)".
- VPC connection:** Includes an 'Info' link and a description: "Connect to a virtual private cloud (VPC) to access network resources without exposing them to the internet." There are two radio buttons: 'None' and 'Custom VPC' (which is selected).
- VPC:** A section titled "Choose a VPC for your function to access." with a dropdown menu showing 'vpc-f0155a95 (172.31.0.0/16)' and a refresh button.
- Subnets:** A section titled "Select the VPC subnets for Lambda to use to set up your VPC configuration." with a dropdown menu showing 'subnet-3e0b4404 (172.31.0.0/20) us-east-1e' and a refresh button. Below the dropdown, it says 'Name: ops-private-a'.
- Warning:** A yellow box with a warning icon stating: "We recommend that you choose at least 2 subnets for Lambda to run your functions in high availability mode."
- Security groups:** A section titled "Choose the VPC security groups for Lambda to use to set up your VPC configuration. The table below shows the inbound and outbound rules for the security groups that you choose." with a dropdown menu showing 'sg-0150c578071ddf138 (stack-LambdaSecurityGroup-6RA31SZJMA3)' and a refresh button. Below the dropdown, it says 'Allow connection to Connector host'.

Configuring an Amazon Virtual Private Cloud (VPC) and Subnets

To configure the existing VPC, you must create a private subnet and associate it with the lambda function.

A private subnet is a subnet with a route table pointing to a Nat gateway.

Elements required:

- A VPC
- A public subnet, a public subnet is a subnet associated with an internet gateway.

To create a public subnet:

1. Create an internet gateway if you don't have one.
2. From the VPC console, go to the navigation pane and choose Subnets.
3. To create a new subnet, choose Create Subnet. Otherwise, choose an existing subnet.
4. Choose the Route Table tab, and then choose Edit.
5. From the Change to: drop-down menu, choose an appropriate route table.
For a public subnet, the default route should point to an internet gateway.

To create a NAT gateway:

1. From the VPC console, go to the navigation panel, choose NAT Gateways, and then choose Create NAT Gateway.
2. In the Subnet field, choose the public subnet already created
3. In the Elastic IP Allocation ID field, choose an existing Elastic IP address, or select Create New EIP, and then choose Create a NAT Gateway.

To create a route table

1. In the VPC console, choose Route Tables, and then choose Create Route Table.
2. In the Name tag field, enter a name that is meaningful to you, select the VPC drop-down menu and choose your VPC, and then choose Yes, Create
3. Select the new route table, and then choose the Routes tab.
4. Choose Edit, and then choose Add another route.

Destination: 0.0.0.0/0

Target: private subnet with the NAT gateway created in the previous step

Opening Ports

You must ensure that the ports on the server on which you installed the Syslog NG Daemon SmartConnector is accessible from AWS. The procedure to open ports varies based on whether you have installed Syslog NG Daemon SmartConnector on a virtual machine or not.

Opening Ports on a Non-Virtual Machine

If you installed Syslog NG Daemon SmartConnector on a physical, non-virtual machine, ensure that the ports on which you installed it are accessible to AWS.

Opening Ports on a Virtual Machine

If you have installed the Syslog NG Daemon SmartConnector on a virtual machine in AWS, ensure that the ports on which you installed Syslog NG Daemon SmartConnector are open in both AWS and the virtual machine.

To open inbound ports on AWS:

1. Log in to AWS as a user with administrator privileges.
2. Go to **Services** and select **EC2**.
3. Select **Instances**.
4. Choose the EC2 instance to be edited.
5. Click **Launch-Wizard**.
6. Edit the **Inbound** and **Outbound** fields as required.

To open ports in the virtual server:

1. Log in to the virtual AWS machine.
2. Open the AWS Firewall.
3. Click **Inbound Rules < New Rule < Port < Next < TCP < Specific local ports**
4. Enter the same port or port range on which you installed the Syslog NG Daemon SmartConnector.
5. Click **Next < Allow the connection < Next < Profile < Next**.
6. Name the rule.
7. Click **Finish**.

Note: Linux EC2 instances are not required to follow these steps, only Windows instances.

Deploying the Connector

About

This section provides information about deploying the AWS CloudWatch Connector to collect and forward events from AWS CloudWatch to an ArcSight Syslog NG Daemon SmartConnector or to an ArcSight Load Balancer, then the events can be sent to an ArcSight destination.

Procedure

AWS Credentials Configuration

The first time the `installer.sh` is executed, the AWS credentials must be entered. If the installer was previously executed on the EC2 instance in the main menu pops, so you can update the AWS credentials.

1. Provide execution rights to all the `.sh` files.
2. Execute the "`installer.sh`" and enter your AWS Access Key ID.
3. Enter your AWS Secret Access Key.

If invalid, a message is displayed, indicating to enter the AWS Secret Access Key again.

4. Select the region.

You are taken to the main screen. If required, your credentials and the region can be updated.

Deployment

1. From the main menu, select **Deploy**.
S3 buckets are scanned for analysis.
2. Name your stack. AWS resources created during deployment can be found under this name.

Note: Stack names must be unique on each region and they must have a valid format, if the Stack Name already exists, or if it does not meet the format criteria, an error message is displayed.

3. To create Flow Log Components, select **Yes/No**. If choosing **Yes**, the installer creates the new flow log group under the VPC previously selected, continue to the

next step. If choosing **No**, continue to step 5 and provide an existing Log Group.

4. Enter a retention value in days.

Retention days are applied to newly created flow logs group under the VPC.

The values permitted

are: 1,3,5,7,14,30,60,90,120,150,180,365,400,545,731,1827,3653.

5. Enter the **Log Groups** names. Every log group is monitored and then sent to the event processing Lambda.

Log Groups must exist in the current Region and they cannot be previously subscribed to another service.

6. Select the VPC in which the AWS CloudWatch connector should gather the events

7. Select the S3 bucket in which the event processing jar and properties file are located.

8. Enter the path for the jar key.

The format is: folderName/subFolderName/jarKeyName.jar

Note: If an invalid format is entered, an error message is displayed.

9. Enter the external properties file.

The format is: folderName/subFolderName/jarKeyName.jar

10. Select the private subnet created previously, for more information, see ["Prerequisites" on page 10](#).

11. Enter the port number. It must be the same port as in properties file.

You are taken to the Deployment summary screen

12. Click **Yes** if the values are correct.

You are taken to the Loading screen. When completed, a message is displayed, confirming the deployment was successful.

Updating the Connector

The update functionality allows the user to change some parameters or data entered at the moment of deployment. For example, once the user has selected the Stack, the script allows to update the VPC selected, the S3 bucket, the path for the jar key and jar key name, the path and the properties file name, the Subnet and the port number.

1. From the main menu, select **Update**.
S3 buckets are scanned for analysis.
2. Select the Stack you want to update.
3. To create Flow Log Components, select **Yes/No**.

The installer highlights the option previously selected at the time of deployment. If yes, continue to the next step. If not, continue to step 5.

4. Enter a retention value in days.

Retention days are applied to newly created flow logs group under the VPC.

The values permitted

are: 1,3,5,7,14,30,60,90,120,150,180,365,400,545,731,1827,3653.

5. Enter the **Log Groups** names. Every log group is monitored and then sent to the event processing Lambda.
6. Select a VPC.
7. Select an S3.
8. Update the jar key path/name.
9. Update external properties key path/name.
10. Choose a Subnet.
11. Update the port number.
12. Click **Yes**, if the values are correct.

You are taken to the Update Status screen. When completed, a message is displayed, confirming the update was successful.

Post-Deployment Configurations

Running Lambda Functions in High Availability

When the subnet runs out of IP addresses or if the availability zone goes down, AWS displays a warning.

You can run functions in high availability mode.

1. From **Services < Network < Lambda < Functions**.
2. Select the Lambda Function created.
3. Click **Network**.

For more information, see [AWS Documentation](#) .

Removing Lambda's Subnet Warning

If the subnet is only used for Lambda events, open a browser. It is very unlikely for the subnet to run out of IP addresses since it is only used by two of Lambda Functions in the stack (CloudwatchConnectorEventProcessing and Hearthbeat).

If an availability zone goes down and cannot communicate with the Connector, the events are not sent to the destination, and instead, they are stored in a S3 bucket. The

moment, the availability zones comes back online, the Lambda Event Processing function processes these events first, before processing any new events.

Network

Virtual Private Cloud (VPC) [Info](#)

Choose a VPC for your function to access.

Subnets

Select the VPC subnets for Lambda to use to set up your VPC configuration. Format: "subnet-id (cidr-block) | az name-tag".

subnet-

(14 172.31.0.0/24) | us-east-2b

⚠ We recommend that you choose at least 2 subnets for Lambda to run your functions in high availability mode.

Security groups

Choose the VPC security groups for Lambda to use to set up your VPC configuration. Format: "sg-id (sg-name) | name-tag". The table below shows the inbound and outbound rules for the security groups that you chose.

sg-

(Elastic Load Balancing)-LambdaSecurityGroup-

i When you enable a VPC, your Lambda function loses default internet access. **If you require external internet access for your function, make sure that your security group allows outbound connections and that your VPC has a NAT gateway.**

To add multiple subnets to Lambda functions, manually:

1. Create a new private Subnet in your VPC.
The destination of the new private subnet cannot be located in the same availability zone in which your original subnet was created.
2. Identify the Lambda functions from your stack STACKNAME-CloudwatchConnectorEventProcessing and STACKNAME-Heartbeat.
3. From each Lambda function, go to **Network** and select your recently created subnet from the dropdown.
4. Click **Save**, the warning should not be displayed.

Upgrading the Connector

The upgrade allows to do a binary upgrade of the AWS CloudWatch connector. A binary upgrade, enables you to continue using the components created during deployment. Your custom settings should not be affected.

It's necessary to update the path or the name of the jar key that contains the binary changes.

To upgrade the connector:

1. Run the `installer.sh` .
2. Select the **Stack** you are upgrading.
The script returns the data with the preferences entered during installation or a previous update.
3. A confirmation screen pops, click **Yes**.
The "Enter a Jar Key" screen is displayed.
4. Update the path and/or the name of the jar key and confirm your changes. You are taken to the Update Status Screen. When completed, a message is displayed, confirming the update was successful.

Note: You may update other values. if necessary.

Undeploying the Connector

To undeploy the connector:

1. From the main menu, select **Undeploy**.
2. Select the Stack to be undeployed.
3. Click **Yes**, to confirm undeployment.
You are taken to the Deleting Network Interfaces screen and next, the undeployment progress is shown. When completed, a message is displayed, confirming the undeployment was successful.

Configuring the Load Balancer as a Destination

In environments where the event load is more than what can be handled by a single Syslog NG Daemon SmartConnector, you can configure Load Balancer to handle large event loads. For more information about configuring Load Balancer, see *ArcSight Load Balancer's* documentation.

Configuring Route 53

Prerequisites

- One or more public domains.

About

You can configure **Amazon Route 53** and log information about queries that Route 53 receives, such as :

- Requested main domains or subdomains
- Date and time of a request
- DNS record types (such as A or AAAA)
- Route 53 edge locations responding to a DNS query
- A DNS response code, such as **NoError** or **ServFail**

Query Logging

Query logs contain queries that DNS resolvers forward to Route 53. If a DNS resolver caches a response of a query, for example, the IP address of a Load Balancer, the resolver continues to return the cached response without forwarding the query to Route 53, until the TTL of the record expires.

Query logs might contain the information of one single query, out of every several thousand queries submitted to DNS resolvers. This depends on the DNS queries submitted from one single domain name or a subdomain name, the resolvers being used, and the TTL (time to live) of the record. For more information about how DNS works, see the [AWS Documentation](#).

An example of a Route 53 Query Log looks like this:

```

1.0 2017-12-13T08:15:50.235Z Z123412341234 example.com AAAA NOERROR TCP IAD12
192.168.3.1 192.168.222.0/24

1.0 2017-12-13T08:16:03.983Z Z123412341234 example.com ANY NOERROR UDP FRA6
2001:db8::1234 2001:db8:abcd::/48

1.0 2017-12-13T08:15:50.342Z Z123412341234 bad.example.com A NXDOMAIN UDP
IAD12 192.168.3.1 192.168.111.0/24

1.0 2017-12-13T08:16:05.744Z Z123412341234 txt.example.com TXT NOERROR UDP
JFK5 192.168.1.2 -

```

Using Amazon CloudWatch to Access DNS Query Logs

Amazon Route 53 sends query logs directly to CloudWatch Logs. The logs are not accessible through Route 53. Instead, you use CloudWatch Logs to view logs in near real-time, search and filter data, and export logs to Amazon S3.

Route 53 creates one CloudWatch Logs log stream for each Route 53 edge location that responds to DNS queries of the specified hosted zone and sends query logs to the applicable log stream.

The format of each log stream name is hosted-zone-id/edge-location-ID, for example, Z1D633PJN98FT9/DFW3.

Each edge location is identified by a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations might change in the future.) For a list of edge locations, see [Route 53 Product Details](#).

Procedure

To Configure Query Logging

1. Sign in to the AWS Management Console and open the Route 53 console at <https://console.aws.amazon.com/route53/>
2. From **Navigation**, click **Hosted Zones**.
3. On the hosted zone you are configuring, click the **Radio** button.
4. In **Hosted Zone Details**, select **Configure Query Logging**.
5. From **Send Query Logs to**, select one of the following options:

- a. New Log Group in US East (North Virginia)
- b. Existing Log Group in US East (North Virginia)

If you want to configure this feature for multiple hosted zones, we recommend you to use a consistent prefix for every log group, for example: `/aws/route53/hosted-zone-name`. Resource policies grant permissions to Route 53 and let users publish logs to the Log Group. The maximum number of resource policies that you can create for an AWS account is 10. And, by using a consistent prefix in your Log Group names, you can choose one resource policy for all your Route 53 hosted zones.

Note: Log Groups must be in the US East (North Virginia) Region.

6. If choosing, **Existing Log Group in US East (North Virginia)**, enter the **Existing Log Group Name** and click **Next**. If not, go to step 7.

Route 53 can use the same permissions in any existing Resource Policy, you do not need to choose a specific Resource Policy.

- a. Click **Test** to determine if the existing resource policies grant the permissions required for Route 53 to publish logs to the log group. If the test fails, users may follow one of these options:
 - i. Continue to step 7, to create a New Resource Policy, or:
 - ii. Click **Edit** and continue to change the value **Log groups that the resource policy applies to**. Specify the name of a CloudWatch Logs log group, for example `/aws/route53/example.com`, or a value that includes the current log group, like, `/aws/route53/*`. Next, move on to step 8.

Note: Users can use the wildcard character (*) to replace 0 or to more characters in the name of the log group.

7. To create a **New Log Group in US East (North Virginia)**:

- a. Enter the **Resource Policy Name**.
- b. Enter a valid value in **Log groups that the resource policy applies to**.

The name of the Log Group is at the top of the current page.

- c. Click **Create Policy and test permissions** to determine if the existing resource policies grant the permissions required for Route 53 to publish logs to the log group. If the error "Resource limit exceeded" pops:
 - i. Click **Edit** in one of the existing resource policies.
 - ii. Change the value **Log groups that the resource policy applies to**. Specify the name of a log group, for example `/aws/route53/example.com`, or a value that includes the current log group, like, `/aws/route53/*`.

To view the settings of a Resource Policy, click the arrow on the left side of

the resource policy name.

- iii. Click **Save policy and test permissions**.

Note: Users can use the wildcard character (*) to replace 0 or more characters in the name of the log group.

8. Select **Create Query Logging Config**.

To Stop Query Logging:

Users can delete query logging configurations and Amazon Route 53 will no longer send query logs to CloudWatch Logs.

1. Sign in to the **AWS Management Console**.
2. Open the [Route 53 Console](#).
3. From **Navigation**, click **Hosted zones**.
4. Go to the hosted zone you are deleting the query logging from and click the **Radio** button.
5. From **Hosted Zone Details < Query Logging**, click **Delete**.
6. Click **Confirm** to delete the query logging configuration.

For more information, see [AWS documentation](#).

Configuring AWS CloudHSM

AWS CloudHSM provides hardware security modules in the AWS Cloud. A hardware security module (HSM) is a computing device that processes cryptographic operations and provides secure storage for cryptographic keys.

When you use an HSM from AWS CloudHSM, you can perform a variety of cryptographic tasks:

- Generate, store, import, export, and manage cryptographic keys, including symmetric keys and asymmetric key pairs.
- Use symmetric and asymmetric algorithms to encrypt and decrypt data.
- Use cryptographic hash functions to compute message digests and hash-based message authentication codes (HMACs).
- Cryptographically sign data (including code signing) and verify signatures.
- Generate cryptographically secure random data.

To configure AWS CloudHSM

1. [Create IAM Administrative Groups.](#)
2. [Create a Virtual Private Cloud \(VPC\).](#)
3. [Create a Private Subnet.](#)
4. [Create a Cluster.](#)
5. [Review Cluster Security Group.](#)
6. [Launch an Amazon EC2 Client Instance.](#)
7. [Connect Amazon EC2 Instance to AWS CloudHSM Cluster.](#)
8. [Create an HSM.](#)
9. [Verify the Identity and Authenticity of Your Cluster's HSM \(Optional\).](#)
10. [Initialize the Cluster.](#)
11. [Install and Configure the AWS CloudHSM Client \(Linux\)](#)[Install and Configure the AWS CloudHSM Client \(Windows\).](#)
12. [Activate the Cluster.](#)

For more information, see [Getting Started with AWS CloudHSM](#).

Monitoring AWS CloudHSM Audit Logs

When an HSM in your account receives a command from the AWS CloudHSM command line tools or software libraries, it records its execution of the command in audit log form. The HSM audit logs include all client-initiated management commands, including those that create and delete the HSM, log into and out of the HSM, and manage users and keys. These logs provide a reliable record of actions that have changed the state of the HSM.

AWS CloudHSM collects your HSM audit logs and sends them to Amazon CloudWatch Logs on your behalf.

Interpreting HSM Audit Logs

The events in the HSM audit logs have standard fields. Some event types have additional fields that capture useful information about the event. For example, user login and user management events include the user name and user type of the user. Key management commands include the key handle.

An example of CloudHSM Audit logs looks like this:

Time: 12/19/17 21:01:17.174902, usecs:1513717277174902

Sequence No : 0x2

Reboot counter : 0xe8

Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_CREATE_APPLIANCE_USER (0xfc)
Session Handle : 0x1004001
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : app_user
User Type : CN_APPLIANCE_USER (5)

Audit Log Reference

AWS CloudHSM records HSM management commands in audit log events. Each event has an operation code (Opcode) value that identifies the action that occurred and its response. You can use the Opcode values to search, sort, and filter the logs. For more information, see [Audit Log Reference](#).

Configuring AWS CloudHSM Audit Logs

AWS CloudHSM provides command line tools for managing and using AWS CloudHSM

Manage Clusters and HSMs

These tools get, create, delete, and tag AWS CloudHSM clusters and HSMs:

- [cloudhsmv2](#). To use these commands, you need to [install](#) and [configure](#) AWS CLI.
- HSM2 PowerShell cmdlets in the [AWS Tools for PowerShell](#). These cmdlets are available in a Windows PowerShell module and a cross-platform PowerShell Core module.

Manage Users

This tool creates and deletes HSM users, including implementing quorum authentication of user management tasks:

The **cloudhsm_mgmt_util** command line tool helps crypto officers manage users in the HSMs. It includes tools that create, delete, and list users, and change user passwords. This tool is included in the [Install the AWS CloudHSM Client and Command Line Tools](#).

cloudhsm_mgmt_util also includes commands that allow crypto users (CUs) to share keys and get and set key attributes.

Preparing to run cloudhsm_mgmt_util

Complete the following tasks before you use cloudhsm_mgmt_util. You need to do these steps the first time you use cloudhsm_mgmt_util and after you add or remove HSMs in your cluster.

1. Stop the AWS CloudHSM Client

- Amazon Linux, CentOS 6, RHEL 6: `$ sudo stop cloudhsm-client`
- Amazon Linux 2, CentOS 7, RHEL 7, Ubuntu 16.04 LTS: `$ sudo service cloudhsm-client stop`
- Windows: `C:\Program Files\Amazon\CloudHSM>net.exe stop AWSCloudHSMClient`

2. Update the AWS CloudHSM Configuration Files

- Amazon Linux, CentOS 6, RHEL 6, Amazon Linux 2, CentOS 7, RHEL 7, Ubuntu 16.04 LTS: `$ sudo /opt/cloudhsm/bin/configure -a <HSM ENI IP>`
- Windows: `C:\Program Files\Amazon\CloudHSM>configure.exe -a <HSM ENI IP>`

3. Start the AWS CloudHSM Client

- Amazon Linux, CentOS 6, RHEL 6: `$ sudo start cloudhsm-client`
- Amazon Linux 2, CentOS 7, RHEL 7, Ubuntu 16.04 LTS: `$ sudo service cloudhsm-client start`
- Windows client 1.1.2+: `C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient`
- Windows 1.1.1 and older: `C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe C:\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg`

4. Update the cloudhsm_mgmt_util Configuration File

- Amazon Linux, CentOS 6, RHEL 6, Amazon Linux 2, CentOS 7, RHEL 7, Ubuntu 16.04 LTS: `$ sudo /opt/cloudhsm/bin/configure -m`
- Windows: `C:\Program Files\Amazon\CloudHSM>configure.exe -m`

After completing this step, you can start cloudhsm_mgmt_util and run cloudhsm_mgmt_util commands. For more information on cloudhsm_mgmt_util commands and examples of using the commands, see [cloudhsm_mgmt_util Command Reference](#).

Manage Keys

This tool creates, deletes, imports, and exports symmetric keys and asymmetric key pairs:

The `key_mgmt_util` command line tool helps Crypto Users (CU) manage keys in the HSMs. It includes multiple commands that generate, delete, import, and export keys, get and set attributes, find keys, and perform cryptographic operations.

Set Up `key_mgmt_util`

Complete the following setup before you use `key_mgmt_util`:

1. Start the AWS CloudHSM Client

Before you use `key_mgmt_util`, you must start the AWS CloudHSM client.

2. Start `key_mgmt_util`

- Amazon Linux, CentOS 6, RHEL 6, Amazon Linux 2, CentOS 7, RHEL 7, Ubuntu 16.04 LTS: `$ sudo /opt/cloudhsm/bin/ key_mgmt_util`
- Windows: `C:\Program Files\Amazon\CloudHSM>key_mgmt_util.exe`

The prompt changes to Command: when `key_mgmt_util` is running and we can run commands. For more information about the commands, see [key_mgmt_util Command Reference](#).

Helper Tools

The [Configure Tool](#) updates your CloudHSM client configuration files. This enables the AWS CloudHSM to synchronize the HSMs in a cluster.

These tools help you to use the tools and software libraries.

[Verify the Performance of the HSM](#) measures the performance of your HSM hardware independent of software libraries.

Overriding Parser Files

You can override the internal parsers with the custom parsers. The parser format looks like this:

```
# CEF fields
# any alphanumeric value will be included in the CEF result
# any value that starts with @ and followed by a number 0-13
# will be replaced with the corresponding flow log field value in the CEF result
# HEADER fields, all fields start with Upper case
DeviceVendor=Arcsight
DeviceProduct=Cloudwatch Connector
```



```

DeviceVersion=@0
DeviceEventClassID=CW
Name=AWS Flow Log Event
Severity=0
# EXTENSION fields, all fields in lower case
duid=@1
deviceInboudInterface=@2
src=@3
dst=@4
spt=@5
dpt=@6
proto=@7
cn=@8
in=@9
start=@10
end=@11
act=@12
cs=@13

```

After the parser files are created, save them as VPC Flow Logs, Route 53 Logs, or CloudHSM Audit Logs. Custom parsers and internal parser files are the same; however, the mapping fields change.

Parser Category	File name
VPC Flow logs	vpc.map
Route53 logs	route53.map
CloudHSM Audit Logs	cloudhsm.map

Note: If a parser file does not have the correct name, it cannot be recognized.

To override an internal parser file:

1. Create a custom parser file and upload it to an S3 bucket and a directory that is dedicated to parser overrides.
The S3 bucket and the directory are defined in the **external.properties** file. For more information, see ["Prerequisites" on page 10](#).
2. **Override an existing parser file in the Lambda function.**

It updates the Event Processor Lambda Function.

To override a file and restart the connector, do one of the following:

- Use the AWS command-line interface (CLI):

```
aws lambda update-function-code --function-name <value> --s3-bucket
<value> --s3-key <jar file path>
```

For more information, see [update-function-code](#).

- Use the AWS Console:
 - a. Go to the jar file path that is configured in the S3 bucket.
 - b. Click the **Object URL** icon and copy the URL.
 - c. Go to the Lambda Console.
 - d. In the **Function Code** group box, click **Action**, then select **Upload a file from Amazon S3**.
 - e. In the **Amazon S3 link URL** box, paste the object URL you copied, then click **Save**.

Updating Parser Files

ArcSight SmartConnector for AWS CloudWatch is designed to support parser file updates at run time without any code changes.

The updates extend events of supported services. You can apply parser updates on the basic installation of connectors.

You can apply the following parser updates:

- Updates received from Micro Focus. See ["Updating Parser Files" above](#).
- Updates received from a support team or other sources. See ["Updating Parser Files" above](#).

Apply Monthly Parser Updates from Micro Focus

The monthly updates for ArcSight SmartConnector parser releases are available at the ArcSight Marketplace. To set up your administrative account and download the parser updates, refer to ArcSight Marketplace at: <https://marketplace.microfocus.com/arcsight>

The updates for all parsers are available from Micro Focus as an **ArcSight-8.1.0xxx.0-ConnectorParsers.aup** package.

To apply the parser updates:

1. Download the **ArcSight-8.1.0xxx.0-ConnectorParsers.aup** package from the ArcSight Marketplace.

2. To apply monthly parser updates to Cloud Connectors:

- a. Download the `ArcSight-8.1.0xxxx.0-aup-extractor.jar` utility from the location where you have downloaded the connector.

Note: Your system must have Java 1.8.x or later version installed and Java available in the operating system's path to use the `aup-extractor.jar` utility

- b. Specify the following command to use the utility to extract parser files from the package:

```
java -jar aup-extractor.jar <AUP filename>
```

Examples:

- `java -jar aup-extractor.jar ArcSight-8.1.0xxxx.0-ConnectorParsers.aup` - When the **.aup** package is in the same directory where the JAR file is present.
- `java -jar aup-extractor.jar c:\MyFolder\ArcSight-8.1.0xxxx.0-ConnectorParsers.aup` - When the **.aup** package is present in other directory.

You can either provide one or both the parameters. If you do not provide any parameters, the utility picks up any available .aup file and creates a new folder named **output** in the directory from where the utility is run and uploads the output files.

The following folders will be extracted:

- **aws_cloudwatch**: Contains security parser for AWS Cloudwatch.
 - **aws_securityhub**: Contains security parser for AWS Security Hub.
 - **azure_emitter**: Contains security parser for Azure emitter.
- c. Copy the parser files in the **output/aws_cloudwatch** folder and upload them to the AWS environment. To upload:
 - i. Navigate to your AWS S3 bucket. For more information, see ["Prerequisites" on page 10](#).
 - ii. Search for the **Maps** folder and upload the new parser files.
 - d. Follow the step and sub-steps in [Override an existing parser file in the Lambda function](#).

Apply the Parser Updates from a Support Team or Other Sources

In this case, the updated parser files are already available, which you need to add in the AWS environment.

To add updated parser files provided by a Support Team or other sources to the AWS environment:

- a. Navigate to your AWS S3 bucket. For more information, see ["Prerequisites" on page 10](#).
- b. Search for the **Maps** folder and upload the new parser files.
- c. Follow the step and sub-steps in [Override an existing parser file in the Lambda function](#)

Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Configuration Guide (SmartConnectors 8.1.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to Documentation-Feedback@microfocus.com.

We appreciate your feedback!