
Micro Focus Security ArcSight SmartConnectors

SmartConnector for Amazon Web Services CloudWatch Configuration Guide

Document Release Date: January 25, 2019

Software Release Date: January 25, 2019



Legal Notices

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") 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. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2019 Micro Focus or one of its affiliates.

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.

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/ctp/productdocs

Contents

Introduction	4
VPC Events	4
Understanding Data Collection	5
Certified Platforms for AWS Cloudwatch Deployment	6
Prerequisites	7
Deploying the Connector	12
AWS Credentials Configuration	12
Deployment	12
Updating the Connector	13
Post-Deployment Configurations	13
Removing Lambda's Subnet Warning	13
Upgrading the Connector	15
Undeploying the Connector	15
Configuring the Connector Destination	16
Load Balancer	16
Send Documentation Feedback	17

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 instance 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 generate custom application-level events and publish them to CloudWatch Events. You can also 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](#).

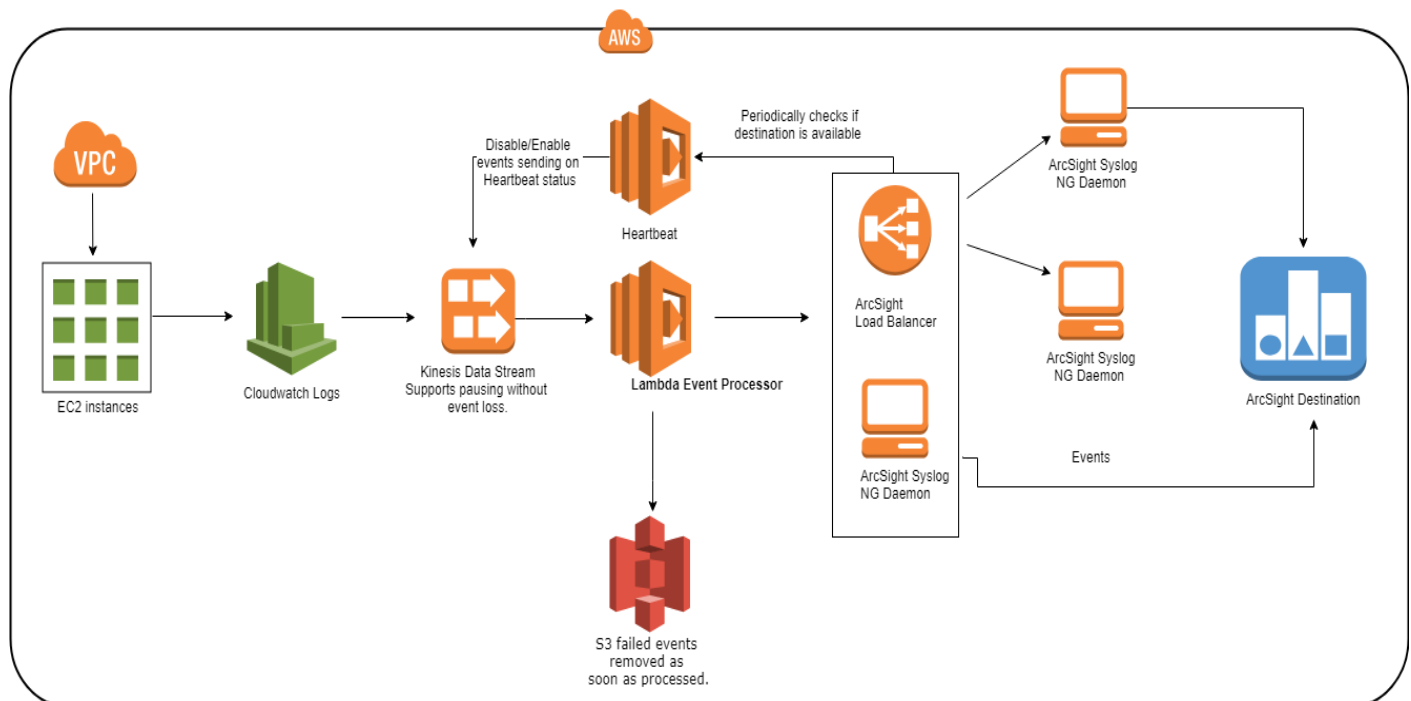
AWS Lambda enables you to build applications that respond quickly to new information. Upload your application code as Lambda functions and Lambda runs your code on high-availability compute infrastructure. Lambda performs all the administration of the compute resources, including server and

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 data to ArcSight's destination.



The AWS VPC Logs Connector gathers events from a specific VPC.

- **AWS VPC Logs**

- **Flow Logs:** Starting point where all the events flow to Cloud Watch Logs (subscription).
- **Cloud Watch Logs** A specific Log Group receives the events from the Flow Logs/.

Note: There is a Log Stream for each EC2 instance that belongs to the VPC.

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.

Prerequisites

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

1. Create a private subnet, see ["Configuring an Amazon Virtual Private Cloud \(VPC\) and Subnets " on page 11](#)
2. Create or select a S3 bucket.

It must be the same location in which the Connector is deployed. The external.properties file, the ArcSight Connector certificate, and the event.processing jar are stored in your S3 bucket. To create a S3 bucket, see, [AWS Documentation](#).

3. Upload your Syslog NG or Load Balancer certificate to your S3 bucket.

The certificate is located in the installation folder of the connector or the Load Balancer. For connectors, the certificate is in \$INSTALLATIONFOLDER/user/agent/remote_management.p12. For Load Balancer, the certificate is in \$INSTALLATIONFOLDER/user/loadbalancer/loadbalancer.p12.

4. Create or select a bucket for fault tolerance files.
5. Create an external 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 location 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 location 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
#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 Connector 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

```

6. [AWS Account Setup](#)

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 use the AWS CloudWatch Connector.

1. Create a new policy.
2. From the AWS Dashboard select an IAM.
3. Select **Policies**.
4. Select **Create Policy**.
5. Go to the **JSON** tab.
6. 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": "*"
    }
  ]
}
```

```
}
]
}
```

7. Select **Review Policy**.
8. Save the new Policy.
 - a. On the field name, enter CloudFormationBasicExecution.
 - b. Select **Create Policy**.
 - c. If created correctly, the message "CloudFormationBasicExecution has been created" is displayed.
9. Create a new user.
 - a. On the AWS Dashboard, select IAM
 - b. Select **Users**.
 - c. Select **Add User**.
 - d. On **User Name**, enter the new user name.
 - e. Go to **Access Type** and select **Programmatic Access** and **AWS Management Console Access**.
 - f. 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.
 - g. Select **Next**.
You are taken to **Permissions**.
 - h. Select **Attach existing policies directly**.
 - i. Click the checkboxes of the following policies:
 - i. CloudFormationBasicExecution created above, and the following AWS managed policies
 - ii. AmazonEC2FullAccess
 - iii. AWSLambdaFullAccess
 - iv. IAMFullAccess
 - v. AmazonS3FullAccess
 - vi. AmazonKinesisFullAccess
 - vii. CloudWatchLogsFullAccess
 - viii. AmazonVPCFullAccess
 - ix. CloudWatchEventsFullAccess
 - j. Go to **Tags < Review < Create User**.

A message is displayed, indicating the user creation was successful.

- k. Write down the **Access Key ID** and the **Secret Access Key**. They are required for deployment.

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 the navigation pane, 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

Deploying the Connector

About

This section provides information about deploying the 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 installer.sh must be executed once, or if the credentials are incorrect. This configuration can be accessed from the main menu.

1. Execute the "installer.sh" and enter your AWS Access Key ID.
2. Enter your AWS Secret Access Key.
If invalid, a message is displayed, indicating to enter the AWS Secret Access Key again.
3. Select the region.
You are taken to the main screen. If required, your credentials and the region can be updated.

Deployment

1. From the first screen, 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. Select the VPC in which the AWS CloudWatch connector should gather the events.
4. Select the S3 bucket in which the event processing jar and properties file are located.
5. 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.

6. Enter the external properties file.

The format is: folderName/subFolderName/jarKeyName.jar

7. Select the private subnet created previously, for more information, see ["Prerequisites" on page 7](#).
8. Enter the port. It must be the same port as in properties file.

You are taken to the Deployment summary screen.

9. 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

1. From the first screen, select **Update**.
S3 buckets are scanned for analysis.
2. Select the Stack, you want to update.
3. Select a VPC.
4. Select an S3.
5. Update the jar key path/name.
6. Update external properties key path/name.
7. Choose a Subnet.
8. Update the port.
9. 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

About

After deployment, if you go to **Network < Lambda Functions**, AWS displays a warning in case, the subnet runs out of IP addresses or if the availability zone goes down, you can run functions in high availability mode. 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

VPC [Info](#)
Choose a VPC for your function to access.

vpc-076b376f (10.0.0.0/16) | Demo_VPC ▼

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

subnet-0bec1e8bf3a6fac47 (10.0.5.0/24) | us-east-2b lambda-private-sn ✕

⚠ 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.

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 update functionality allows the user to update 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.

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

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 updating.

The script returns the data with the preferences entered during installation or a previous update.

3. A confirmation screen pops, click **Yes**.

Undeploying the Connector

To undeploy the connector:

1. From the first screen, select **Undeploy**.
2. Select the Stack to be undeployed.
3. Click **Yes**, to confirm undeployment.

You are taken to the Delete Status screen and next, the undeployment progress is shown. When completed a message is displayed, confirming the undeployment was successful.

Configuring the Connector Destination

Load Balancer

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.

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 7.11)

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 arcsight_doc@microfocus.com.

We appreciate your feedback!