Analyzing CloudWatch Logs with Kinesis Data Streams and Kinesis Data Analytics

**SPL-TF-300-ARARCH-10-EN - Version 1.0.1**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

As a security administrator at AnyCompany, you want to monitor your Amazon Elastic Compute Cloud (Amazon EC2) instances for abnormal account activity. You have a Linux-based EC2 instance with an application database configured in your environment. If a local user on the database instance fails to log in several times within a set time frame, that could be an indication of a bad actor attempting to brute force attack the account. You want to establish automated alerts to monitor for failed log in attempts and be able to quickly analyze network traffic logs to determine if a breach may have occurred.

In this challenge-based lab, where the step-by-step instructions are originally hidden, you must review the objective of each task and the expected result, and then research and implement the steps required to complete the task.

OBJECTIVES

By the end of this lab, you should be able to do the following:

* Install the Amazon CloudWatch Logs agent on a Linux-based EC2 instance.
* Send system authentication logs to Amazon CloudWatch Logs.
* Configure an Amazon CloudWatch alarm to send notifications after authentication failures.
* Create VPC flow logs to capture network traffic.
* Create Kinesis data streams to store logs from CloudWatch Logs.
* Create a Lambda function to process CloudWatch log data in a Kinesis data stream.
* Analyze logs with Amazon Managed Service for Apache Flink using SQL queries.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should have a basic knowledge of AWS services as defined in the AWS Cloud Practitioner Essentials course. You should be comfortable with navigating the AWS Management Console.

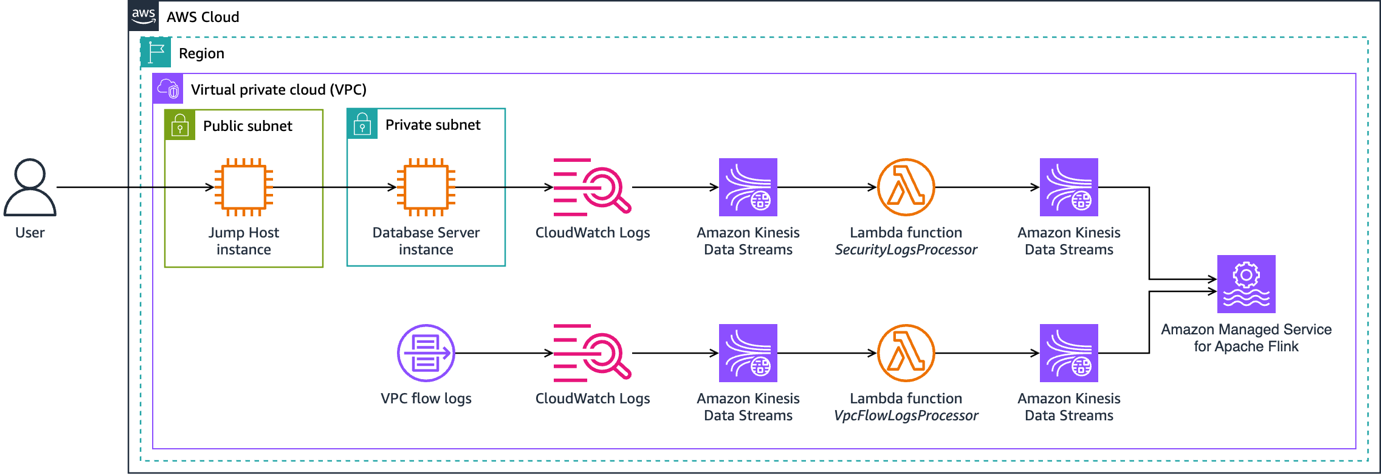
ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Consider:** A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand.
* **File contents:** A code block that displays the contents of a script or file you need to run that has been pre-created for you.
* **Security:** An opportunity to incorporate security best practices.
* **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
* **Task complete:** A conclusion or summary point in the lab.

LAB ENVIRONMENT

The following architecture diagram depicts the AWS services and resources that are used in this lab.



*Image description: The preceding image depicts the architecture diagram of the lab. In this lab, you connect to an EC2 instance (named Jump Host) that is hosted in a public subnet. You then use SSH to connect to another EC2 instance (named Database Server) that is hosted in a private subnet. The CloudWatch agent that is installed on the Database Server instance sends security log messages to a log group in CloudWatch Logs. VPC flow logs are also sent to CloudWatch Logs. The log messages from each log group are then sent to Amazon Kinesis Data streams, where an AWS Lambda function is configured to process the messages from each data stream and send the results to another data stream. Finally, Amazon Managed Service for Apache Flink is used to query the data streams that contain the processed log messages.*

SERVICES USED IN THIS LAB

**Amazon Elastic Compute Cloud (Amazon EC2)**

Amazon EC2 is a web service that provides sizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Just as Amazon Simple Storage Service (Amazon S3) enables storage in the cloud, Amazon EC2 enables *compute* in the cloud. The Amazon EC2 simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

**Amazon CloudWatch Logs**

Amazon CloudWatch Logs lets you monitor and troubleshoot your systems and applications using your existing system, application and custom log files.

With CloudWatch Logs, you can monitor your logs, in near real time, for specific phrases, values or patterns. For example, you could set an alarm on the number of errors that occur in your system logs or view graphs of latency of web requests from your application logs. You can then view the original log data to see the source of the problem. Log data can be stored and accessed indefinitely in highly durable, low-cost storage so you don’t have to worry about filling up hard drives. Amazon CloudWatch Logs lets you monitor and troubleshoot your systems and applications using your existing system, application, and custom log files.

**Amazon Kinesis Data Streams**

Amazon Kinesis Data Streams is a serverless streaming data service that makes it easy to capture, process, and store data streams at any scale. You can use Kinesis Data Streams to collect and process large streams of data records in real time. With Kinesis Data Streams, you can build custom applications that process or analyze streaming data for specialized needs. You can add various types of data such as clickstreams, application logs, and social media to a Kinesis data stream from hundreds of thousands of sources. Within seconds, the data will be available for your applications to read and process from the stream.

**AWS Lambda**

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code, and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

**Amazon Managed Service for Apache Flink**

With Amazon Managed Service for Apache Flink, you can transform and analyze streaming data in real time with Apache Flink. Apache Flink is an open source framework and engine for processing data streams. Amazon Managed Service for Apache Flink reduces the complexity of building, managing, and integrating Apache Flink applications with other AWS services.

Amazon Managed Service for Apache Flink takes care of everything required to continuously run streaming applications and scales automatically to match the volume and throughput of your incoming data. With Amazon Managed Service for Apache Flink, there are no servers to manage, there is no minimum fee or setup cost, and you only pay for the resources your streaming applications consume.

AWS SERVICES NOT USED IN THIS LAB

The lab environment adheres to the [principle of least-privilege permissions](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege). As such, the actions you can perform are limited to only those that are required to complete the lab tasks and learning objectives. Access to AWS services that are not needed to complete the lab tasks is restricted. Expect errors if you attempt to perform actions beyond those provided in the lab guide.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Warning:** Do not change the **Region** unless instructed.

COMMON SIGN-IN ERRORS

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Installing and configuring the Amazon CloudWatch Logs agent**

In this task, you install the Amazon CloudWatch agent on the *Database Server* instance, which uses Amazon Linux 2023. You first connect to the *Jump Host* instance in the public subnet, and then use SSH to connect to the *Database Server* in the private subnet. The *Database Server* instance hosts a database that contains sensitive customer information, so you would like to be notified of any failed attempts to sign in.

At the end of this task, you should be able to validate that the security logs from the *Database Server* instance are in CloudWatch Logs.

**Note:** This lab uses Session Manager, a capability of AWS Systems Manager, as the method of connecting to the *Jump Host* EC2 instance

**Task 1 hints**

**Task 1 full walkthrough**

**Task 2: Creating an Amazon CloudWatch alarm to monitor for failed login attempts**

Now that you are sending the security logs from the database server to CloudWatch, you can create CloudWatch metrics and alarms to monitor the logs for any abnormal behavior.

In this task, you create a CloudWatch metric filter to locate *authentication failure* event messages in the logs from the database server. You then create a CloudWatch alarm and notification to email you any time there are more than two login failures within a 5 minute window.

At the end of this task, you should have a CloudWatch alarm that activates with the given parameters, and send an Amazon Simple Notification Service (Amazon SNS) notification to your email address.

**Task 2 hints**

**Task 2 full walkthrough**

**Task 3: Configuring VPC flow logs to send network traffic data to Amazon CloudWatch Logs**

In this task, you create a CloudWatch log group to store network traffic activity data. You then create a VPC flow log and configure it to send data to the CloudWatch log group. Finally, you generate SSH traffic and observe the results in the VPC flow logs.

At this end of this task, you should be able to validate that the VPC flow logs from the *Lab VPC* VPC are in CloudWatch Logs.

**Note:** An IAM role has been pre-created as part of the lab environment build process for use when configuring the VPC flow logs. It is named **VpcFlowLogsRole**

**Important**

Your account does not have permissions to create IAM roles in this lab. To assist you with task that require a specific IAM role and permissions, a role has been pre-created for you during the lab environment build process.

For this task, you can use the **VpcFlowLogsRole** role.

**Task 3 hints**

**Task 3 full walkthrough**

**Task 4: Sending log messages from CloudWatch Logs to Kinesis Data Streams**

In this task, you create Kinesis data streams to ingest the security and VPC flow log data from CloudWatch Logs that you created in the previous tasks. You then create CloudWatch subscription filters to send log data to the respective data streams.

**Important**

Your account does not have permissions to create IAM roles in this lab. To assist you with task that require a specific IAM role and permissions, a role has been pre-created for you during the lab environment build process.

For this task, you can use the **PutKinesisDataStreamsRole** role.

**Note:** If you would like to use the same resource names as the guided walkthrough steps, use the following:

* Kinesis data streams names:
  + security\_log\_stream
  + vpc\_flow\_log\_stream

**Task 4 hints**

**Task 4 full walkthrough**

**Task 5: Processing the CloudWatch Logs data**

As you discoverd in the previous task, when CloudWatch Logs sends data to other AWS services through a subscription filter, it is automatically gzip compressed and base64 encoded. Before you can analyze the data, you must decompress and decode it.

In this task, you create two Kinesis data streams to store the processed log data. You then create a Python-based AWS Lambda function to decode, decompress, and send the log data to the log streams.

**Important**

Your account does not have permissions to create IAM roles in this lab. To assist you with task that require a specific IAM role and permissions, a role has been pre-created for you during the lab environment build process.

For this task, you can use the **LambdaCWLogsProcessorRole** role.

**File contents:** To assist with building a Python-based function that extracts the keys and values, the following sections provide an example of what the JSON-formatted security and VPC flow log messages looks like. Expand each section to view them.

**Sample JSON-formatted security log**

**Sample JSON-formatted VPC flow log**

**Note:** If you would like to use the same resource names as the guided walkthrough steps, use the following:

* Kinesis data streams names:
  + processed\_security\_logs
  + processed\_vpc\_flow\_logs

**Task 5 hints**

**Task 5 full walkthrough**

**Task 6: Analyzing the log data with Amazon Managed Service for Apache Flink**

Now that you have processed the log data, you can use an Amazon Managed Service for Apache Flink notebook to analyze the log data.

**Important**

Your account does not have permissions to create IAM roles in this lab. To assist you with task that require a specific IAM role and permissions, a role has been pre-created for you during the lab environment build process.

For this task, you can use the **KinesisNotebookRole** role.

**Task 6 hints**

**Task 6 full walkthrough**

**Conclusion**

You have successfully done the following:

* Installed the Amazon CloudWatch Logs agent on a Linux-based EC2 instance.
* Sent system authentication logs to Amazon CloudWatch Logs.
* Configured an Amazon CloudWatch alarm to send notifications after authentication failures.
* Created VPC flow logs to capture network traffic.
* Created Kinesis data streams to store logs from CloudWatch Logs.
* Created a Lambda function to process CloudWatch log data in a Kinesis data stream.
* Analyzed logs with Amazon Managed Service for Apache Flink using SQL queries.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Aggregation interval](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html#flow-logs-aggregration-interval)
* [Logging IP traffic using VPC Flow Logs](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html)
* [Publish flow logs to CloudWatch Logs](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-cwl.html)
* [Log group-level subscription filters](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html)
* [Kinesis Data Streams - Changing the Data Retention Period](https://docs.aws.amazon.com/streams/latest/dev/kinesis-extended-retention.html)