



**Free Questions for DOP-C02 by certscare**

**Shared by Crane on 18-01-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

## Question Type: MultipleChoice

---

A company hosts applications in its AWS account. Each application logs to an individual Amazon CloudWatch log group. The company's CloudWatch costs for ingestion are increasing.

A DevOps engineer needs to identify which applications are the source of the increased logging costs.

Which solution will meet these requirements?

### Options:

---

- A) Use CloudWatch metrics to create a custom expression that identifies the CloudWatch log groups that have the most data being written to them.
- B) Use CloudWatch Logs Insights to create a set of queries for the application log groups to identify the number of logs written for a period of time.
- C) Use AWS Cost Explorer to generate a cost report that details the cost for CloudWatch usage.
- D) Use AWS CloudTrail to filter for CreateLogStream events for each application.

### Answer:

---

C

## Explanation:

---

The correct answer is C)

A comprehensive and detailed explanation is:

Option A is incorrect because using CloudWatch metrics to create a custom expression that identifies the CloudWatch log groups that have the most data being written to them is not a valid solution. CloudWatch metrics do not provide information about the size or volume of data being ingested by CloudWatch logs. CloudWatch metrics only provide information about the number of events, bytes, and errors that occur within a log group or stream. Moreover, creating a custom expression with CloudWatch metrics would require using the `search_web` tool, which is not necessary for this use case.

Option B is incorrect because using CloudWatch Logs Insights to create a set of queries for the application log groups to identify the number of logs written for a period of time is not a valid solution. CloudWatch Logs Insights can help analyze and filter log events based on patterns and expressions, but it does not provide information about the cost or billing of CloudWatch logs. CloudWatch Logs Insights also charges based on the amount of data scanned by each query, which could increase the logging costs further.

Option C is correct because using AWS Cost Explorer to generate a cost report that details the cost for CloudWatch usage is a valid solution. AWS Cost Explorer is a tool that helps visualize, understand, and manage AWS costs and usage over time. AWS Cost Explorer can generate custom reports that show the breakdown of costs by service, region, account, tag, or any other dimension. AWS Cost Explorer can also filter and group costs by usage type, which can help identify the specific CloudWatch log groups that are the source of the increased logging costs.

Option D is incorrect because using AWS CloudTrail to filter for `CreateLogStream` events for each application is not a valid solution. AWS CloudTrail is a service that records API calls and account activity for AWS services, including CloudWatch logs. However, AWS

CloudTrail does not provide information about the cost or billing of CloudWatch logs. Filtering for CreateLogStream events would only show when a new log stream was created within a log group, but not how much data was ingested or stored by that log stream.

[CloudWatch Metrics](#)

[CloudWatch Logs Insights](#)

[AWS Cost Explorer](#)

[AWS CloudTrail](#)

## Question 2

---

**Question Type: MultipleChoice**

---

A company sells products through an ecommerce web application. The company wants a dashboard that shows a pie chart of product transaction details. The company wants to integrate the dashboard with the company's existing Amazon CloudWatch dashboards.

Which solution will meet these requirements with the MOST operational efficiency?

**Options:**

---

- A)** Update the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction. Use CloudWatch Logs Insights to query the log group and to visualize the results in a pie chart format Attach the results to the desired CloudWatch dashboard.
- B)** Update the ecommerce application to emit a JSON object to an Amazon S3 bucket for each processed transaction. Use Amazon Athena to query the S3 bucket and to visualize the results In a Pie chart format. Export the results from Athena Attach the results to the desired CloudWatch dashboard
- C)** Update the ecommerce application to use AWS X-Ray for instrumentation. Create a new X-Ray subsegment Add an annotation for each processed transaction. Use X-Ray traces to query the data and to visualize the results in a pie chart format Attach the results to the desired CloudWatch dashboard
- D)** Update the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction\_ Create an AWS Lambda function to aggregate and write the results to Amazon DynamoDB. Create a Lambda subscription filter for the log file. Attach the results to the desired CloudWatch dashboard.

### **Answer:**

---

A

### **Explanation:**

---

The correct answer is A.

A comprehensive and detailed explanation is:

Option A is correct because it meets the requirements with the most operational efficiency. Updating the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction is a simple and cost-effective way to collect the data needed for the dashboard. Using CloudWatch Logs Insights to query the log group and to visualize the results in a pie chart format is also a convenient and integrated solution that leverages the existing CloudWatch dashboards. Attaching the results to the desired CloudWatch dashboard is straightforward and does not require any additional steps or services.

Option B is incorrect because it introduces unnecessary complexity and cost. Updating the ecommerce application to emit a JSON object to an Amazon S3 bucket for each processed transaction is a valid way to store the data, but it requires creating and managing an S3 bucket and its permissions. Using Amazon Athena to query the S3 bucket and to visualize the results in a pie chart format is also a valid way to analyze the data, but it incurs charges based on the amount of data scanned by each query. Exporting the results from Athena and attaching them to the desired CloudWatch dashboard is also an extra step that adds more overhead and latency.

Option C is incorrect because it uses AWS X-Ray for an inappropriate purpose. Updating the ecommerce application to use AWS X-Ray for instrumentation is a good practice for monitoring and tracing distributed applications, but it is not designed for aggregating product transaction details. Creating a new X-Ray subsegment and adding an annotation for each processed transaction is possible, but it would clutter the X-Ray service map and make it harder to debug performance issues. Using X-Ray traces to query the data and to visualize the results in a pie chart format is also possible, but it would require custom code and logic that are not supported by X-Ray natively. Attaching the results to the desired CloudWatch dashboard is also not supported by X-Ray directly, and would require additional steps or services.

Option D is incorrect because it introduces unnecessary complexity and cost. Updating the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction is a simple and cost-effective way to collect the data needed for the dashboard, as in option

A) However, creating an AWS Lambda function to aggregate and write the results to Amazon DynamoDB is redundant, as CloudWatch Logs Insights can already perform aggregation queries on log data. Creating a Lambda subscription filter for the log file is also

redundant, as CloudWatch Logs Insights can already access log data directly. Attaching the results to the desired CloudWatch dashboard would also require additional steps or services, as DynamoDB does not support native integration with CloudWatch dashboards.

[CloudWatch Logs Insights](#)

[Amazon Athena](#)

[AWS X-Ray](#)

[AWS Lambda](#)

[Amazon DynamoDB](#)

## Question 3

---

**Question Type:** MultipleChoice

---

A video-sharing company stores its videos in Amazon S3. The company has observed a sudden increase in video access requests, but the company does not know which videos are most popular. The company needs to identify the general access pattern for the video files. This pattern includes the number of users who access a certain file on a given day, as well as the number of pull requests for certain files.

How can the company meet these requirements with the LEAST amount of effort?

### Options:

---

- A) Activate S3 server access logging. Import the access logs into an Amazon Aurora database. Use an Aurora SQL query to analyze the access patterns.
- B) Activate S3 server access logging. Use Amazon Athena to create an external table with the log files. Use Athena to create a SQL query to analyze the access patterns.
- C) Invoke an AWS Lambda function for every S3 object access event. Configure the Lambda function to write the file access information, such as user, S3 bucket, and file key, to an Amazon Aurora database. Use an Aurora SQL query to analyze the access patterns.
- D) Record an Amazon CloudWatch Logs log message for every S3 object access event. Configure a CloudWatch Logs log stream to write the file access information, such as user, S3 bucket, and file key, to an Amazon Kinesis Data Analytics for SQL application. Perform a sliding window analysis.

### Answer:

---

B

## Question 4

---

Question Type: MultipleChoice

---



A company sells products through an ecommerce web application. The company wants a dashboard that shows a pie chart of product transaction details. The company wants to integrate the dashboard with the company's existing Amazon CloudWatch dashboards.

Which solution will meet these requirements with the MOST operational efficiency?

### Options:

---

- A)** Update the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction. Use CloudWatch Logs Insights to query the log group and to visualize the results in a pie chart format. Attach the results to the desired CloudWatch dashboard.
- B)** Update the ecommerce application to emit a JSON object to an Amazon S3 bucket for each processed transaction. Use Amazon Athena to query the S3 bucket and to visualize the results in a pie chart format. Export the results from Athena. Attach the results to the desired CloudWatch dashboard.
- C)** Update the ecommerce application to use AWS X-Ray for instrumentation. Create a new X-Ray subsegment. Add an annotation for each processed transaction. Use X-Ray traces to query the data and to visualize the results in a pie chart format. Attach the results to the desired CloudWatch dashboard.
- D)** Update the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction. Create an AWS Lambda function to aggregate and write the results to Amazon DynamoDB. Create a Lambda subscription filter for the log file. Attach the results to the desired CloudWatch dashboard.

### Answer:

---

A

## Explanation:

---

The correct answer is A.

A comprehensive and detailed explanation is:

Option A is correct because it meets the requirements with the most operational efficiency. Updating the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction is a simple and cost-effective way to collect the data needed for the dashboard. Using CloudWatch Logs Insights to query the log group and to visualize the results in a pie chart format is also a convenient and integrated solution that leverages the existing CloudWatch dashboards. Attaching the results to the desired CloudWatch dashboard is straightforward and does not require any additional steps or services.

Option B is incorrect because it introduces unnecessary complexity and cost. Updating the ecommerce application to emit a JSON object to an Amazon S3 bucket for each processed transaction is a valid way to store the data, but it requires creating and managing an S3 bucket and its permissions. Using Amazon Athena to query the S3 bucket and to visualize the results in a pie chart format is also a valid way to analyze the data, but it incurs charges based on the amount of data scanned by each query. Exporting the results from Athena and attaching them to the desired CloudWatch dashboard is also an extra step that adds more overhead and latency.

Option C is incorrect because it uses AWS X-Ray for an inappropriate purpose. Updating the ecommerce application to use AWS X-Ray for instrumentation is a good practice for monitoring and tracing distributed applications, but it is not designed for aggregating product transaction details. Creating a new X-Ray subsegment and adding an annotation for each processed transaction is possible, but it would clutter the X-Ray service map and make it harder to debug performance issues. Using X-Ray traces to query the data and to visualize the results in a pie chart format is also possible, but it would require custom code and logic that are not supported by X-Ray natively. Attaching the results to the desired CloudWatch dashboard is also not supported by X-Ray directly, and would require additional steps or services.

Option D is incorrect because it introduces unnecessary complexity and cost. Updating the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction is a simple and cost-effective way to collect the data needed for the dashboard, as in option

A) However, creating an AWS Lambda function to aggregate and write the results to Amazon DynamoDB is redundant, as CloudWatch Logs Insights can already perform aggregation queries on log data. Creating a Lambda subscription filter for the log file is also redundant, as CloudWatch Logs Insights can already access log data directly. Attaching the results to the desired CloudWatch dashboard would also require additional steps or services, as DynamoDB does not support native integration with CloudWatch dashboards.

[CloudWatch Logs Insights](#)

[Amazon Athena](#)

[AWS X-Ray](#)

[AWS Lambda](#)

[Amazon DynamoDB](#)

## Question 5

---

**Question Type:** MultipleChoice

---

A video-sharing company stores its videos in Amazon S3. The company has observed a sudden increase in video access requests, but the company does not know which videos are most popular. The company needs to identify the general access pattern for the video files. This pattern includes the number of users who access a certain file on a given day, as well as the number of pull requests for certain files.

How can the company meet these requirements with the LEAST amount of effort?

### Options:

---

- A)** Activate S3 server access logging. Import the access logs into an Amazon Aurora database. Use an Aurora SQL query to analyze the access patterns.
- B)** Activate S3 server access logging. Use Amazon Athena to create an external table with the log files. Use Athena to create a SQL query to analyze the access patterns.
- C)** Invoke an AWS Lambda function for every S3 object access event. Configure the Lambda function to write the file access information, such as user, S3 bucket, and file key, to an Amazon Aurora database. Use an Aurora SQL query to analyze the access patterns.
- D)** Record an Amazon CloudWatch Logs log message for every S3 object access event. Configure a CloudWatch Logs log stream to write the file access information, such as user, S3 bucket, and file key, to an Amazon Kinesis Data Analytics for SQL application. Perform a sliding window analysis.

### Answer:

---

B

## Question 6

---

### Question Type: MultipleChoice

---

A company hosts applications in its AWS account. Each application logs to an individual Amazon CloudWatch log group. The company's CloudWatch costs for ingestion are increasing.

A DevOps engineer needs to identify which applications are the source of the increased logging costs.

Which solution will meet these requirements?

### Options:

---

- A) Use CloudWatch metrics to create a custom expression that identifies the CloudWatch log groups that have the most data being written to them.
- B) Use CloudWatch Logs Insights to create a set of queries for the application log groups to identify the number of logs written for a period of time.
- C) Use AWS Cost Explorer to generate a cost report that details the cost for CloudWatch usage.
- D) Use AWS CloudTrail to filter for CreateLogStream events for each application.

### Answer:

---

C

## Explanation:

---

The correct answer is C)

A comprehensive and detailed explanation is:

Option A is incorrect because using CloudWatch metrics to create a custom expression that identifies the CloudWatch log groups that have the most data being written to them is not a valid solution. CloudWatch metrics do not provide information about the size or volume of data being ingested by CloudWatch logs. CloudWatch metrics only provide information about the number of events, bytes, and errors that occur within a log group or stream. Moreover, creating a custom expression with CloudWatch metrics would require using the `search_web` tool, which is not necessary for this use case.

Option B is incorrect because using CloudWatch Logs Insights to create a set of queries for the application log groups to identify the number of logs written for a period of time is not a valid solution. CloudWatch Logs Insights can help analyze and filter log events based on patterns and expressions, but it does not provide information about the cost or billing of CloudWatch logs. CloudWatch Logs Insights also charges based on the amount of data scanned by each query, which could increase the logging costs further.

Option C is correct because using AWS Cost Explorer to generate a cost report that details the cost for CloudWatch usage is a valid solution. AWS Cost Explorer is a tool that helps visualize, understand, and manage AWS costs and usage over time. AWS Cost Explorer can generate custom reports that show the breakdown of costs by service, region, account, tag, or any other dimension. AWS Cost Explorer can also filter and group costs by usage type, which can help identify the specific CloudWatch log groups that are the source of the increased logging costs.

Option D is incorrect because using AWS CloudTrail to filter for `CreateLogStream` events for each application is not a valid solution. AWS CloudTrail is a service that records API calls and account activity for AWS services, including CloudWatch logs. However, AWS

CloudTrail does not provide information about the cost or billing of CloudWatch logs. Filtering for CreateLogStream events would only show when a new log stream was created within a log group, but not how much data was ingested or stored by that log stream.

[CloudWatch Metrics](#)

[CloudWatch Logs Insights](#)

[AWS Cost Explorer](#)

[AWS CloudTrail](#)

**To Get Premium Files for DOP-C02 Visit**

<https://www.p2pexams.com/products/dop-c02>

**For More Free Questions Visit**

<https://www.p2pexams.com/amazon/pdf/dop-c02>

