Boost up Your Certification Score

# Amazon
# DOP-C02
## AWS Certified DevOps Engineer - Professional Exam

For More Information – Visit link below:

https://www.examsboost.com/

Product Version

✓ Up to Date products, reliable and verified.
✓ Questions and Answers in PDF Format.

Visit us at: https://www.examsboost.com/test/dop-c02

# Latest Version: 19.4

## Question: 1

A company has a mobile application that makes HTTP API calls to an Application Load Balancer (ALB). The ALB routes requests to an AWS Lambda function. Many different versions of the application are in use at any given time, including versions that are in testing by a subset of users. The version of the application is defined in the user-agent header that is sent with all requests to the API.
After a series of recent changes to the API, the company has observed issues with the application. The company needs to gather a metric for each API operation by response code for each version of the application that is in use. A DevOps engineer has modified the Lambda function to extract the API operation name, version information from the user-agent header and response code.
Which additional set of actions should the DevOps engineer take to gather the required metrics?

A. Modify the Lambda function to write the API operation name, response code, and version number as a log line to an Amazon CloudWatch Logs log group. Configure a CloudWatch Logs metric filter that increments a metric for each API operation name. Specify response code and application version as dimensions for the metric.
B. Modify the Lambda function to write the API operation name, response code, and version number as a log line to an Amazon CloudWatch Logs log group. Configure a CloudWatch Logs Insights query to populate CloudWatch metrics from the log lines. Specify response code and application version as dimensions for the metric.
C. Configure the ALB access logs to write to an Amazon CloudWatch Logs log group. Modify the Lambda function to respond to the ALB with the API operation name, response code, and version number as response metadata. Configure a CloudWatch Logs metric filter that increments a metric for each API operation name. Specify response code and application version as dimensions for the metric.
D. Configure AWS X-Ray integration on the Lambda function. Modify the Lambda function to create an X-Ray subsegment with the API operation name, response code, and version number. Configure X-Ray insights to extract an aggregated metric for each API operation name and to publish the metric to Amazon CloudWatch. Specify response code and application version as dimensions for the metric.

**Answer: A**

Explanation:
"Note that the metric filter is different from a log insights query, where the experience is interactive and provides immediate search results for the user to investigate. No automatic action can be invoked from an insights query. Metric filters, on the other hand, will generate metric data in the form of a time series. This lets you create alarms that integrate into your ITSM processes, execute AWS Lambda functions, or even create anomaly detection models."

https://aws.amazon.com/blogs/mt/quantify-custom-application-metrics-with-amazon-cloudwatch-logs-and-metric-filters/

## Question: 2

A company provides an application to customers. The application has an Amazon API Gateway REST API that invokes an AWS Lambda function. On initialization, the Lambda function loads a large amount of data from an Amazon DynamoDB table. The data load process results in long cold-start times of 8-10 seconds. The DynamoDB table has DynamoDB Accelerator (DAX) configured.
Customers report that the application intermittently takes a long time to respond to requests. The application receives thousands of requests throughout the day. In the middle of the day, the application experiences 10 times more requests than at any other time of the day. Near the end of the day, the application's request volume decreases to 10% of its normal total.
A DevOps engineer needs to reduce the latency of the Lambda function at all times of the day.
Which solution will meet these requirements?

A. Configure provisioned concurrency on the Lambda function with a concurrency value of 1. Delete the DAX cluster for the DynamoDB table.
B. Configure reserved concurrency on the Lambda function with a concurrency value of 0.
C. Configure provisioned concurrency on the Lambda function. Configure AWS Application Auto Scaling on the Lambda function with provisioned concurrency values set to a minimum of 1 and a maximum of 100.
D. Configure reserved concurrency on the Lambda function. Configure AWS Application Auto Scaling on the API Gateway API with a reserved concurrency maximum value of 100.

**Answer: C**

Explanation:
The following are the steps that the DevOps engineer should take to reduce the latency of the Lambda function at all times of the day:
Configure provisioned concurrency on the Lambda function.
Configure AWS Application Auto Scaling on the Lambda function with provisioned concurrency values set to a minimum of 1 and a maximum of 100.
The provisioned concurrency setting ensures that there is always a minimum number of Lambda function instances available to handle requests. The Application Auto Scaling setting will automatically scale the number of Lambda function instances up or down based on the demand for the application.
This solution will ensure that the Lambda function is able to handle the increased load during the middle of the day, while also keeping the cold-start latency low.
The following are the reasons why the other options are not correct:
Option A is incorrect because it will not reduce the cold-start latency of the Lambda function.
Option B is incorrect because it will not scale the number of Lambda function instances up or down based on demand.

Option D is incorrect because it will only configure reserved concurrency on the API Gateway API, which will not affect the Lambda function.

## Question: 3

A company is adopting AWS CodeDeploy to automate its application deployments for a Java-Apache Tomcat application with an Apache Webserver. The development team started with a proof of concept, created a deployment group for a developer environment, and performed functional tests within the application. After completion, the team will create additional deployment groups for staging and production.
The current log level is configured within the Apache settings, but the team wants to change this configuration dynamically when the deployment occurs, so that they can set different log level configurations depending on the deployment group without having a different application revision for each group.
How can these requirements be met with the LEAST management overhead and without requiring different script versions for each deployment group?

A. Tag the Amazon EC2 instances depending on the deployment group. Then place a script into the application revision that calls the metadata service and the EC2 API to identify which deployment group the instance is part of. Use this information to configure the log level settings. Reference the script as part of the AfterInstall lifecycle hook in the appspec.yml file.
B. Create a script that uses the CodeDeploy environment variable DEPLOYMENT_GROUP_ NAME to identify which deployment group the instance is part of. Use this information to configure the log level settings. Reference this script as part of the BeforeInstall lifecycle hook in the appspec.yml file.
C. Create a CodeDeploy custom environment variable for each environment. Then place a script into the application revision that checks this environment variable to identify which deployment group the instance is part of. Use this information to configure the log level settings. Reference this script as part of the ValidateService lifecycle hook in the appspec.yml file.
D. Create a script that uses the CodeDeploy environment variable DEPLOYMENT_GROUP_ID to identify which deployment group the instance is part of to configure the log level settings. Reference this script as part of the Install lifecycle hook in the appspec.yml file.

**Answer: B**

Explanation:
The following are the steps that the company can take to change the log level dynamically when the deployment occurs:
Create a script that uses the CodeDeploy environment variable DEPLOYMENT_GROUP_NAME to identify which deployment group the instance is part of.
Use this information to configure the log level settings.
Reference this script as part of the BeforeInstall lifecycle hook in the appspec.yml file.
The DEPLOYMENT_GROUP_NAME environment variable is automatically set by CodeDeploy when the deployment is triggered. This means that the script does not need to call the metadata service or the EC2 API to identify the deployment group.

This solution is the least complex and requires the least management overhead. It also does not require different script versions for each deployment group.
The following are the reasons why the other options are not correct:
Option A is incorrect because it would require tagging the Amazon EC2 instances, which would be a manual and time-consuming process.
Option C is incorrect because it would require creating a custom environment variable for each environment. This would be a complex and error-prone process.
Option D is incorrect because it would use the DEPLOYMENT_GROUP_ID environment variable. However, this variable is not automatically set by CodeDeploy, so the script would need to call the metadata service or the EC2 API to get the deployment group ID. This would add complexity and overhead to the solution.

## Question: 4

A company requires its developers to tag all Amazon Elastic Block Store (Amazon EBS) volumes in an account to indicate a desired backup frequency. This requirement Includes EBS volumes that do not require backups. The company uses custom tags named Backup_Frequency that have values of none, dally, or weekly that correspond to the desired backup frequency. An audit finds that developers are occasionally not tagging the EBS volumes.
A DevOps engineer needs to ensure that all EBS volumes always have the Backup_Frequency tag so that the company can perform backups at least weekly unless a different value is specified.
Which solution will meet these requirements?

A. Set up AWS Config in the account. Create a custom rule that returns a compliance failure for all Amazon EC2 resources that do not have a Backup Frequency tag applied. Configure a remediation action that uses a custom AWS Systems Manager Automation runbook to apply the Backup_Frequency tag with a value of weekly.
B. Set up AWS Config in the account. Use a managed rule that returns a compliance failure for EC2::Volume resources that do not have a Backup Frequency tag applied. Configure a remediation action that uses a custom AWS Systems Manager Automation runbook to apply the Backup_Frequency tag with a value of weekly.
C. Turn on AWS CloudTrail in the account. Create an Amazon EventBridge rule that reacts to EBS CreateVolume events. Configure a custom AWS Systems Manager Automation runbook to apply the Backup_Frequency tag with a value of weekly. Specify the runbook as the target of the rule.
D. Turn on AWS CloudTrail in the account. Create an Amazon EventBridge rule that reacts to EBS CreateVolume events or EBS ModifyVolume events. Configure a custom AWS Systems Manager Automation runbook to apply the Backup_Frequency tag with a value of weekly. Specify the runbook as the target of the rule.

**Answer: B**

Explanation:
The following are the steps that the DevOps engineer should take to ensure that all EBS volumes always have the Backup_Frequency tag so that the company can perform backups at least weekly unless a different value is specified:

Set up AWS Config in the account.
Use a managed rule that returns a compliance failure for EC2::Volume resources that do not have a Backup Frequency tag applied.
Configure a remediation action that uses a custom AWS Systems Manager Automation runbook to apply the Backup_Frequency tag with a value of weekly.
The managed rule AWS::Config::EBSVolumesWithoutBackupTag will return a compliance failure for any EBS volume that does not have the Backup_Frequency tag applied. The remediation action will then use the Systems Manager Automation runbook to apply the Backup_Frequency tag with a value of weekly to the EBS volume.

## Question: 5

A company is using an Amazon Aurora cluster as the data store for its application. The Aurora cluster is configured with a single DB instance. The application performs read and write operations on the database by using the cluster's instance endpoint.
The company has scheduled an update to be applied to the cluster during an upcoming maintenance window. The cluster must remain available with the least possible interruption during the maintenance window.
What should a DevOps engineer do to meet these requirements?

A. Add a reader instance to the Aurora cluster. Update the application to use the Aurora cluster endpoint for write operations. Update the Aurora cluster's reader endpoint for reads.
B. Add a reader instance to the Aurora cluster. Create a custom ANY endpoint for the cluster. Update the application to use the Aurora cluster's custom ANY endpoint for read and write operations.
C. Turn on the Multi-AZ option on the Aurora cluster. Update the application to use the Aurora cluster endpoint for write operations. Update the Aurora cluster’s reader endpoint for reads.
D. Turn on the Multi-AZ option on the Aurora cluster. Create a custom ANY endpoint for the cluster. Update the application to use the Aurora cluster's custom ANY endpoint for read and write operations.

**Answer: C**

Explanation:
To meet the requirements, the DevOps engineer should do the following:
Turn on the Multi-AZ option on the Aurora cluster.
Update the application to use the Aurora cluster endpoint for write operations.
Update the Aurora cluster's reader endpoint for reads.
Turning on the Multi-AZ option will create a replica of the database in a different Availability Zone. This will ensure that the database remains available even if one of the Availability Zones is unavailable.
Updating the application to use the Aurora cluster endpoint for write operations will ensure that all writes are sent to both the primary and replica databases. This will ensure that the data is always consistent.

Updating the Aurora cluster's reader endpoint for reads will allow the application to read data from the replica database. This will improve the performance of the application during the maintenance window.

# Thank You for Trying Our Product

For More Information – Visit link below:

https://www.examsboost.com/

**15 USD Discount Coupon Code:**

**G74JA8UF**

# FEATURES

- ✓ 90 Days Free Updates
- ✓ Money Back Pass Guarantee
- ✓ Instant Download or Email Attachment
- ✓ 24/7 Live Chat Support
- ✓ PDF file could be used at any Platform
- ✓ 50,000 Happy Customer

