

素敵なDOP-C02模擬体験と最高のDOP-C02模擬対策問題



2026年GoShikenの最新DOP-C02 PDFダンプおよびDOP-C02試験エンジンの無料共有: https://drive.google.com/open?id=1KzqXZF5Nsl85yJ3KTY7APbcZQ1cQiv_q

すべての顧客の誠実な要件を考慮して、DOP-C02テスト問題は、高品質の製品、思いやりのあるアフターサービスを備えた候補者に約束します。試験での99%の合格率、購入前の無料トライアル、安全なプライバシー保護など、DOP-C02トレーニング資料の多くの利点がよく認識されています。お客様の視点から、最適なDOP-C02模擬試験へのすべてのお客様の信頼とフィードバックを大切に、最良の選択になります。

Amazon DOP-C02 認定試験は、DevOpsプラクティスとAWSサービスの専門知識を証明するための優れた方法です。この認定資格を取得することで、個人は潜在的な雇用主やクライアントに自分のスキルをアピールすることができ、それによってより良い求人機会や高い給与を得ることができます。

>> DOP-C02模擬体験 <<

完璧なDOP-C02模擬体験 & 合格スムーズDOP-C02模擬対策問題 | 有効的なDOP-C02関連資格試験対応

ユーザーのプライバシー保護は、インターネット時代の永遠の問題です。多くの違法ウェブサイトはユーザーのプライバシーを第三者に販売するため、多くの購入者は奇妙なウェブサイトを信じることを嫌います。ただし、DOP-C02学習エンジンDOP-C02を購入する際に心配する必要はまったくありません。ユーザーの情報が私たちの評判を傷つけているため、ユーザーの情報を決して販売しないことを保証します。

Amazon AWS Certified DevOps Engineer - Professional 認定 DOP-C02 試験問題 (Q340-Q345):

質問 # 340

A company has migrated its container-based applications to Amazon EKS and want to establish automated email notifications. The notifications sent to each email address are for specific activities related to EKS components. The solution will include Amazon SNS topics and an AWS Lambda function to evaluate incoming log events and publish messages to the correct SNS topic. Which logging solution will support these requirements?

- A. Enable Amazon CloudWatch Logs to log the EKS components. Create CloudWatch Logs Insights queries linked to Amazon EventBridge events that invoke Lambda.
- **B. Enable Amazon CloudWatch Logs to log the EKS components. Create a CloudWatch subscription filter for each component with Lambda as the subscription feed destination.**
- C. Enable Amazon S3 logging for the EKS components. Configure S3 PUT Object event notifications with AWS Lambda as the destination.
- D. Enable Amazon S3 logging for the EKS components. Configure an Amazon CloudWatch subscription filter for each component with Lambda as the subscription feed destination.

正解: B

解説:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html#LambdaFunctionExample>
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html>

質問 # 341

A company is using AWS CodePipeline to deploy an application. According to a new guideline, a member of the company's security team must sign off on any application changes before the changes are deployed into production. The approval must be recorded and retained.

Which combination of actions will meet these requirements? (Select TWO.)

- A. Create an AWS CloudTrail trail to deliver logs to Amazon S3.
- B. Configure CodePipeline to write actions to an Amazon S3 bucket at the end of each pipeline stage.
- C. Create a CodePipeline manual approval action before the deployment step. Create a policy that grants the security team access to approve manual approval stages.
- D. Configure CodePipeline to write actions to Amazon CloudWatch Logs.
- E. Create a CodePipeline custom action to invoke an AWS Lambda function for approval. Create a policy that gives the security team access to manage CodePipeline custom actions.

正解: A、C

解説:

To meet the new guideline for application deployment, the company can use a combination of AWS CodePipeline and AWS CloudTrail. A manual approval action in CodePipeline allows the security team to review and approve changes before they are deployed. This action can be configured to pause the pipeline until approval is granted, ensuring that no changes move to production without the necessary sign-off.

Additionally, by creating an AWS CloudTrail trail, all actions taken within CodePipeline, including approvals, are recorded and delivered to an Amazon S3 bucket. This provides an audit trail that can be retained for compliance and review purposes.

References:

* AWS CodePipeline's manual approval action provides a way to ensure that a member of the security team can review and approve changes before they are deployed¹.

* AWS CloudTrail integration with CodePipeline allows for the recording and retention of all pipeline actions, including approvals, which can be stored in Amazon S3 for record-keeping².

質問 # 342

A company has an application and a CI/CD pipeline. The CI/CD pipeline consists of an AWS CodePipeline pipeline and an AWS CodeBuild project. The CodeBuild project runs tests against the application as part of the build process and outputs a test report. The company must keep the test reports for 90 days.

Which solution will meet these requirements?

- A. Add a new stage in the CodePipeline pipeline after the stage that contains the CodeBuild project. Create an Amazon S3 bucket to store the reports. Configure an S3 deploy action type in the new CodePipeline stage with the appropriate path and format for the reports.
- B. Add a new stage in the CodePipeline pipeline. Configure a test action type with the appropriate path and format for the reports. Configure the report expiration time to be 90 days in the CodeBuild project buildspec file.
- C. Add a report group in the CodeBuild project buildspec file with the appropriate path and format for the reports. Create an Amazon S3 bucket to store the reports. Configure an Amazon EventBridge rule that invokes an AWS Lambda function to copy the reports to the S3 bucket when a build is completed. Create an S3 Lifecycle rule to expire the objects after 90 days.
- D. Add a report group in the CodeBuild project buildspec file with the appropriate path and format for the reports. Create an Amazon S3 bucket to store the reports. Configure the report group as an artifact in the CodeBuild project buildspec file. Configure the S3 bucket as the artifact destination. Set the object expiration to 90 days.

正解: C

解説:

The correct solution is to add a report group in the AWS CodeBuild project buildspec file with the appropriate path and format for the reports. Then, create an Amazon S3 bucket to store the reports. You should configure an Amazon EventBridge rule that invokes an AWS Lambda function to copy the reports to the S3 bucket when a build is completed. Finally, create an S3 Lifecycle rule to expire the objects after 90 days. This approach allows for the automated transfer of reports to long-term storage and ensures they

are retained for the required duration without manual intervention¹.

References:

- * AWS CodeBuild User Guide on test reporting¹.
- * AWS CodeBuild User Guide on working with report groups².
- * AWS Documentation on using AWS CodePipeline with AWS CodeBuild³.

質問 # 343

A company uses an Amazon Aurora PostgreSQL global database that has two secondary AWS Regions. A DevOps engineer has configured the database parameter group to guarantee an RPO of 60 seconds. Write operations on the primary cluster are occasionally blocked because of the RPO setting.

The DevOps engineer needs to reduce the frequency of blocked write operations.

Which solution will meet these requirements?

- **A. Remove one of the secondary clusters from the global database.**
- B. Add an additional secondary cluster to the global database.
- C. Configure synchronous replication for the global database.
- D. Enable write forwarding for the global database.

正解: A

解説:

Step 1: Reducing Replication Lag in Aurora Global Databases

In Amazon Aurora global databases, write operations on the primary cluster can be delayed due to the time it takes to replicate to secondary clusters, especially when there are multiple secondary regions involved.

Issue: The write operations are occasionally blocked due to the RPO setting, which guarantees replication within 60 seconds.

Action: Remove one of the secondary clusters from the global database.

Why: Fewer secondary clusters will reduce the overall replication lag, improving write performance and reducing the frequency of blocked writes.

Reference:

This corresponds to Option C: Remove one of the secondary clusters from the global database.

質問 # 344

A DevOps engineer manages a Java-based application that runs in an Amazon Elastic Container Service (Amazon ECS) cluster on AWS Fargate. Auto scaling has not been configured for the application. The DevOps engineer has determined that the Java Virtual Machine (JVM) thread count is a good indicator of when to scale the application. The application serves customer traffic on port 8080 and makes JVM metrics available on port 9404. Application use has recently increased. The DevOps engineer needs to configure auto scaling for the application. Which solution will meet these requirements with the LEAST operational overhead?

- A. Deploy the Amazon CloudWatch agent as a container sidecar. Configure a metric filter for the JVM thread count metric on the CloudWatch log group for the CloudWatch agent. Add a target tracking policy in Fargate. Select the metric from the metric filter as a scale target.
- B. Create an Amazon Managed Service for Prometheus workspace. Deploy AWS Distro for OpenTelemetry as a container sidecar to retrieve JVM metrics from port 9404 to publish the JVM metrics from port 9404 to the Prometheus workspace. Add a target tracking policy in Fargate. Select the Prometheus metric as a scale target.
- C. Create an Amazon Managed Service for Prometheus workspace. Deploy AWS Distro for OpenTelemetry as a container sidecar to publish the JVM metrics from port 9404 to the Prometheus workspace. Configure rules for the workspace to use the JVM thread count metric to scale the application. Add a step scaling policy in Fargate. Select the Prometheus rules to scale up and scaling down.
- **D. Deploy the Amazon CloudWatch agent as a container sidecar. Configure the CloudWatch agent to retrieve JVM metrics from port 9404. Create CloudWatch alarms on the JVM thread count metric to scale the application. Add a step scaling policy in Fargate to scale up and scale down based on the CloudWatch alarms.**

正解: D

質問 # 345

.....

