



---

**TRUSTED, ACCURATE AND  
RELIABLE!**

---

**The most comprehensive IT certification  
preparation materials in the industry!**

All rights reserved. No part of this document may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. Unauthorized copying, reselling, or distribution of this document is strictly prohibited and may result in legal action.

<https://www.virtulearner.com>  
[support@virtulearner.com](mailto:support@virtulearner.com)

**Amazon**

**DVA-C02**

**AWS Certified Developer -**

**Associate**

**QUESTION: 1**

A company is implementing an application on Amazon EC2 instances. The application needs to process incoming transactions. When the application detects a transaction that is not valid, the application must send a chat message to the company's support team. To send the message, the application needs to retrieve the access token to authenticate by using the chat API.

A developer needs to implement a solution to store the access token. The access token must be encrypted at rest and in transit. The access token must also be accessible from other AWS accounts.

Which solution will meet these requirements with the LEAST management overhead?

- A. Use an AWS Systems Manager Parameter Store SecureString parameter that uses an AWS Key Management Service (AWS KMS) AWS managed key to store the access token. Add a resource-based policy to the parameter to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Parameter Store. Retrieve the token from Parameter Store with the decrypt flag enabled. Use the decrypted access token to send the message to the chat.
- B. Encrypt the access token by using an AWS Key Management Service (AWS KMS) customer managed key. Store the access token in an Amazon DynamoDB table. Update the IAM role of the EC2 instances with permissions to access DynamoDB and AWS KMS. Retrieve the token from DynamoDB and decrypt the token by using AWS KMS on the EC2 instances. Use the decrypted access token to send the message to the chat.
- C. Use AWS Secrets Manager with an AWS Key Management Service (AWS KMS) customer managed key to store the access token. Add a resource-based policy to the secret to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Secrets Manager. Retrieve the token from Secrets Manager. Use the decrypted access token to send the message to the chat.
- D. Encrypt the access token by using an AWS Key Management Service (AWS KMS) AWS managed key. Store the access token in an Amazon S3 bucket. Add a bucket policy to the S3 bucket to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Amazon S3 and AWS KMS. Retrieve the token from the S3 bucket. Decrypt the token by using AWS KMS on the EC2 instances. Use the decrypted access token to send the message to the chat.

**Answer(s):** C

**QUESTION: 2**

A company is running Amazon EC2 instances in multiple AWS accounts. A developer needs to implement an application that collects all the lifecycle events of the EC2 instances. The application needs to store the lifecycle events in a single Amazon Simple Queue Service (Amazon SQS) queue in the company's main AWS account for further processing.

Which solution will meet these requirements?

- A. Configure Amazon EC2 to deliver the EC2 instance lifecycle events from all accounts to the Amazon EventBridge event bus of the main account. Add an EventBridge rule to the event bus of the main account that matches all EC2 instance lifecycle events. Add the SQS queue as a target of the rule.
- B. Use the resource policies of the SQS queue in the main account to give each account permissions to write to that SQS queue. Add to the Amazon EventBridge event bus of each account an EventBridge rule that matches all EC2 instance lifecycle events. Add the SQS queue in the main account as a target of the rule.
- C. Write an AWS Lambda function that scans through all EC2 instances in the company accounts to detect EC2 instance lifecycle changes. Configure the Lambda function to write a notification message to the SQS queue in the main account if the function detects an EC2 instance lifecycle change. Add an Amazon EventBridge scheduled rule that invokes the Lambda function every minute.
- D. Configure the permissions on the main account event bus to receive events from all accounts. Create an Amazon EventBridge rule in each account to send all the EC2 instance lifecycle events to the main account event bus. Add an EventBridge rule to the main account event bus that matches all EC2 instance lifecycle events. Set the SQS queue as a target for the rule.

**Answer(s):** D

**QUESTION: 3**

An application is using Amazon Cognito user pools and identity pools for secure access. A developer wants to integrate the user-specific file upload and download features in the application with Amazon S3. The developer must ensure that the files are saved and retrieved in a secure manner and that users can access only their own files. The file sizes range from 3 KB to 300 MB.

Which option will meet these requirements with the HIGHEST level of security?

- A. Use S3 Event Notifications to validate the file upload and download requests and update the user interface (UI).
- B. Save the details of the uploaded files in a separate Amazon DynamoDB table. Filter the list of files in the user interface (UI) by comparing the current user ID with the user ID associated with the file in the table.
- C. Use Amazon API Gateway and an AWS Lambda function to upload and download files. Validate each request in the Lambda function before performing the requested operation.
- D. Use an IAM policy within the Amazon Cognito identity prefix to restrict users to use their own folders in Amazon S3.

**Answer(s):** D

**QUESTION: 4**