Welcome to Cert007 - Your Ultimate IT Certification Partner

- Real Exam Questions
- Instant Access
- Free Updates
- Money-Back Guarantee
- Expert Support

Visit us at **https://www.cert007.com/** for more information

Exam : **DVA-C03**

Title : AWS Certified Developer - Associate

Version : DEMO

1.A ticketing API stores the remaining seats for each performance in a DynamoDB table in one AWS Region. Multiple workers can reserve seats concurrently. Every request has a positive integer quantity and is submitted only once. An existing item has a numeric `remaining` attribute.
The workers use this Boto3 Table resource call, with valid keys and expression values:

```
table.update_item(
    Key={"performanceId": performance_id},
    UpdateExpression="SET #left = #left - :requested",
    ExpressionAttributeNames={"#left": "remaining"},
    ExpressionAttributeValues={":requested": quantity}
)
```

A reservation must fail without changing the item when fewer than `quantity` seats remain.
Which change prevents the count from becoming negative during concurrent requests?
A. Read the item with ConsistentRead=True, verify the count in the worker, and then run the unchanged update.
B. Add ConditionExpression="#left >= :requested" to the same UpdateItem request and handle a failed condition as a rejected reservation.
C. Replace the update expression with ADD using the negative quantity and run the update without a condition.
D. Use if_not_exists to initialize remaining to zero in the update expression and run the subtraction without a condition.
Answer: B
Explanation:
B — Correct. The condition is evaluated as part of the conditional update. Only an item with sufficient remaining seats is decremented; a failed condition leaves that attempted update unapplied.

2.A Lambda function processes batches from a standard SQS queue. The event source mapping already enables ReportBatchItemFailures. Each record can be processed independently, and successful work is idempotent.
The current handler uses the following pseudocode:

```
for record in event.Records:
    process(record)  # may raise an exception
return {"batchItemFailures": []}
```

When the third record fails, Lambda retries records that were processed successfully earlier in the batch.
Which handler change makes the response identify only records that failed processing?
A. Catch the exception around the entire loop and return an empty batchItemFailures list.
B. Leave the loop unchanged and configure a Lambda asynchronous OnFailure destination.
C. Collect failed message IDs and raise a single exception containing those IDs after the loop.
D. Catch errors separately for each record, continue processing other records, and return failed message IDs as itemIdentifier entries in batchItemFailures.
Answer: D
Explanation:
D — Correct. With ReportBatchItemFailures enabled, a successful handler return can identify failed records using their message IDs. Per-record error handling allows other independent records to finish. Idempotency is still needed because partial responses do not guarantee exactly-once processing.

3.An ECS service runs Linux tasks on Fargate. Each task pulls an image from a private ECR repository and then calls DynamoDB PutItem from application code using the default AWS SDK credential provider chain. The task definition specifies separate task and task execution roles with valid trust policies. Network access, DynamoDB capacity, and the deployer's permissions are already correct. There are no explicit denies or other required application permissions.
Which TWO permission assignments are needed for image retrieval and the application's write, while keeping these permissions on their intended roles? (Choose two.)
A. Grant the task execution role ecr:GetAuthorizationToken and the repository-scoped ECR image-pull actions.
B. Grant the task execution role dynamodb:PutItem on the application's target table.
C. Grant the task role dynamodb:PutItem on the application's target table.
D. Grant the task role the ECR image-pull actions instead of granting them to the task execution role.
E. Grant both sets of permissions only to the IAM role that submits the ECS deployment.
Answer: A, C
Explanation:
A — Correct. ECS uses the task execution role for image retrieval. GetAuthorizationToken uses Resource *, while the image-pull actions can be scoped to the required repository.
C — Correct. The default container credential provider supplies task-role credentials to application code. The table-scoped PutItem permission authorizes the required write.

4.A SAM template already deploys a Lambda function through the `live` alias. All production callers invoke that alias.
The function uses this configuration:

```
AutoPublishAlias: live
DeploymentPreference:
  Type: Canary10Percent5Minutes
```

The team has created a CloudWatch alarm named `CheckoutFailureAlarm` that correctly detects failures during the canary interval. The SAM-generated CodeDeploy deployment has the required permissions.
Which change connects that alarm to the deployment so an alarm breach can trigger rollback during traffic shifting?
A. Add the alarm name to the function's EventInvokeConfig OnFailure destination.
B. Add the alarm name to DeploymentPreference.Alarms.
C. Add a PreTraffic hook that checks the alarm once before any traffic is shifted, without configuring deployment alarms.
D. Set ProvisionedConcurrencyConfig on the live alias and leave DeploymentPreference unchanged.
Answer: B
Explanation:
B — Correct. DeploymentPreference.Alarms associates the CloudWatch alarm with the deployment. This provides the deployment with an alarm signal during its traffic-shifting interval.

5.An application invokes a standard Lambda function synchronously. At a sustained request rate, some Invoke API calls return `TooManyRequestsException`. The function has reserved concurrency set to 5.

During the same interval, monitoring shows:

Observation	Value
Function ConcurrentExecutions, Maximum	5
Function Throttles, Sum	Increasing
Function Errors, Sum	0
Successful invocation duration	Stable
Additional concurrency available in the account	More than 200

Load tests confirm that the function and its downstream service can safely support 20 concurrent requests, which is sufficient for the current load. The team wants to retain a dedicated concurrency allocation for this function.

Which change directly addresses the observed bottleneck?

A. Increase the function timeout while leaving reserved concurrency at 5.

B. Increase MaximumRetryAttempts in the function's asynchronous invocation configuration.

C. Configure provisioned concurrency of 5 while leaving reserved concurrency at 5.

D. Increase the function's reserved concurrency from 5 to 20.

Answer: D

Explanation:

D — Correct. The observed concurrency reaches the function's configured cap while account capacity remains available. Raising that cap to the validated level addresses the capacity limit and retains a dedicated allocation.