DBCLOUD
EXAMS

QUESTIONS & ANSWERS

verified



Technology Provider: Amazon

Exam: AWS Certified Security - Specialty SCS-C03

Number of questions in the database: 93

Exam Version: NOV.2026

Question #1*Topic 1*

A security engineer is troubleshooting an AWS Lambda function that is named MyLambdaFunction. The function is encountering an error when the function attempts to read the objects in an Amazon S3 bucket that is named DOC-EXAMPLE-BUCKET. The S3 bucket has the following bucket policy:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "lambda.amazonaws.com"
    },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:lambda:::function:MyLambdaFunction"
        }
    }
}
```

Which change should the security engineer make to the policy to ensure that the Lambda function can read the bucket objects?

A. Remove the Condition element. Change the Principal element to the following:

```
{
    "AWS": "arn:aws:lambda:::function:MyLambdaFunction"
}
```

B. Change the Action element to the following:

```
[
    "s3:GetObject*",
    "s3:GetBucket*"
]
```

C. Change the Resource element to "arn:aws:s3:::DOC-EXAMPLE- BUCKET/*''.

D. Change the Resource element to "arn:aws:lambda:::function:MyLambdaFunction". Change the Principal element to the following:

```
{
    "Service": "s3.amazonaws.com"
}
```

Correct Answer: C

https://dbcloud.academy/

Question #2*Topic 1*

HOTSPOT -

A company is building a web application that needs to authenticate external users across multiple microservices that the company hosts on Amazon Elastic Container Service (Amazon ECS). The solution must use temporary credentials and minimize the management overhead required to maintain user databases.

Select and order the correct steps from the following list to implement a secure authentication strategy that meets these requirements. Select each step one time or not at all.

Configure Amazon Cognito user pools for user authentication.

Set up an IAM role for each microservice. Grant each role appropriate permissions.

Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices.

Create an Amazon DynamoDB table to store user credentials for each microservice.

Create an Amazon Cognito application client to interact with the web application.

Set up AWS IAM Identity Center to give users access to the microservices.

Step 1:

Configure Amazon Cognito user pools for user authentication. Set up an IAM role for each microservice. Grant each role appropriate permissions. Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices. Create an Amazon DynamoDB table to store user credentials for each microservice. Create an Amazon Cognito application client to interact with the web application. Set up AWS IAM Identity Center to give users access to the microservices.

Step 2:

Configure Amazon Cognito user pools for user authentication. Set up an IAM role for each microservice. Grant each role appropriate permissions. Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices. Create an Amazon DynamoDB table to store user credentials for each microservice. Create an Amazon Cognito application client to interact with the web application. Set up AWS IAM Identity Center to give users access to the microservices.

Step 3:

Configure Amazon Cognito user pools for user authentication. Set up an IAM role for each microservice. Grant each role appropriate permissions. Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices. Create an Amazon DynamoDB table to store user credentials for each microservice. Create an Amazon Cognito application client to interact with the web application. Set up AWS IAM Identity Center to give users access to the microservices.

Correct Answer:

https://dbcloud.academy/

Step 1:

Configure Amazon Cognito user pools for user authentication.
Set up an IAM role for each microservice. Grant each role appropriate permissions.
Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices.
Create an Amazon DynamoDB table to store user credentials for each microservice.
Create an Amazon Cognito application client to interact with the web application.
Set up AWS IAM Identity Center to give users access to the microservices.

Step 2:

Configure Amazon Cognito user pools for user authentication.
Set up an IAM role for each microservice. Grant each role appropriate permissions.
Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices.
Create an Amazon DynamoDB table to store user credentials for each microservice.
Create an Amazon Cognito application client to interact with the web application.
Set up AWS IAM Identity Center to give users access to the microservices.

Step 3:

Configure Amazon Cognito user pools for user authentication.
Set up an IAM role for each microservice. Grant each role appropriate permissions.
Implement an Amazon API Gateway HTTP API with AWS Lambda authorizers to validate tokens before forwarding requests to microservices.
Create an Amazon DynamoDB table to store user credentials for each microservice.
Create an Amazon Cognito application client to interact with the web application.
Set up AWS IAM Identity Center to give users access to the microservices.

https://dbcloud.academy/

Question #3*Topic 1*

An AWS account administrator created an IAM group and applied the following managed policy to require that each individual user authenticate using multi-factor authentication:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*"
        },
        {
            "Sid": "BlockAnyAccessUnlessSignedInWithMFA",
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": false
                }
            }
        }
    ]
}
```

After implementing the policy, the administrator receives reports that users are unable to perform Amazon EC2 commands using the AWS CLI.
What should the administrator do to resolve this problem while still enforcing multi-factor authentication?

A. Change the value of aws:MultiFactorAuthPresent to true.

B. Instruct users to run the aws sts get-session-token CLI command and pass the multi-factor authentication --serial-number and -token-code parameters. Use these resulting values to make API/CLI calls.

C. Implement federated API/CLI access using SAML 2.0, then configure the identity provider to enforce multi-factor authentication.

D. Create a role and enforce multi-factor authentication in the role trust policy. Instruct users to run the sts assume-role CLI command and pass --serial-number and --token-code parameters. Store the resulting values in environment variables. Add sts:AssumeRole to NotAction in the policy.

Correct Answer: B

Question #4*Topic 1*

A company is using AWS Organizations with the default SCP. The company needs to restrict AWS usage for all AWS accounts that are in a specific OU.
Except for some desired global services, the AWS usage must occur only in the eu-west-1 Region for all accounts in the OU. A security engineer must create an SCP that applies the restriction to existing accounts and any new accounts in the OU.
Which SCP will meet these requirements?

https://dbcloud.academy/

A.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyNonDefaultRegions",
            "Effect": "Deny",
            "NotAction": [
                <Desired Global Services> ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": [
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

B.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyNonDefaultRegions",
            "Effect": "Allow",
            "Action": [
                <Desired Global Services> ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": [
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

https://dbcloud.academy/

C.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyNonDefaultRegions",
            "Effect": "Deny",
            "NotAction": [
                <Desired Global Services> ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

D.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyNonDefaultRegions",
            "Effect": "Allow",
            "NotAction": [
                <Desired Global Services> ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

Correct Answer: C

https://dbcloud.academy/