Free Questions for SCS-C01 by actualtestdumps

Shared by Griffin on 07-06-2022

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company created an AWS account for its developers to use for testing and learning purposes Because MM account will be shared among multiple teams of developers, the company wants to restrict the ability to stop and terminate Amazon EC2 instances so that a team can perform these actions only on the instances it owns.

Developers were Instructed to tag al their instances with a Team tag key and use the team name in the tag value One of the first teams to use this account is Business Intelligence A security engineer needs to develop a highly scalable solution for providing developers with access to the appropriate resources within the account The security engineer has already created individual 1AM roles for each team.

Which additional configuration steps should the security engineer take to complete the task?

Options:

A) For each team, create an AM policy similar to the one that fellows Populate the ec2: ResourceTag/Team condition key with a proper team name Attach resulting policies to the corresponding 1AM roles.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "NotAction": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys/Team": "${aws:PrincipalTag/Team}"
                }
            }
        }
    ]
}
```

B) For each team create an 1AM policy similar to the one that follows Populate the aws TagKeys/Team condition key with a proper team name. Attach the resuming policies to the corresponding 1AM roles.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "NotAction": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys/Team": "${aws:PrincipalTag/Team}"
                }
            }
        }
    ]
}
```

C) Tag each 1AM role with a Team lag key. and use the team name in the tag value. Create an 1AM policy similar to the one that follows, and attach 4 to all the 1AM roles used by developers.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "NotAction": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys/Team": "${aws:PrincipalTag/Team}"
                }
            }
        }
    ]
}
```

D) Tag each IAM role with the Team key, and use the team name in the tag value. Create an IAM policy similar to the one that follows, and it to all the IAM roles used by developers.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "NotAction": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys/Team": "${aws:PrincipalTag/Team}"
                }
            }
        }
    ]
}
```

Answer:

A

Question 2

Question Type: MultipleChoice

Auditors tor a health care company have mandated mat all data volumes be encrypted at rest Infrastructure is deployed mainly via AWS CloudFormation however third-party frameworks and manual deployment are required on some legacy systems

What is the BEST way to monitor, on a recurring basis, whether all EBS volumes are encrypted?

Options:

A) On a recurring basis, update an IAM user policies to require that EC2 instances are created with an encrypted volume

B) Configure an AWS Config rule lo run on a recurring basis 'or volume encryption

C) Set up Amazon Inspector rules tor volume encryption to run on a recurring schedule

D) Use CloudWatch Logs to determine whether instances were created with an encrypted volume

Answer:

A

Question 3

Question Type: **MultipleChoice**

A company has a website with an Amazon CloudFront HTTPS distribution, an Application Load Balancer (ALB) with multiple web instances for dynamic website content, and an Amazon S3 bucket for static website content. The company's security engineer recently updated the website security requirements:

* HTTPS needs to be enforced for all data in transit with specific ciphers.

* The CloudFront distribution needs to be accessible from the internet only.

Which solution will meet these requirements?

Set up an S3 bucket policy with the awssecuretransport key Configure the CloudFront origin access identity (OAI) with the S3 bucket Configure CloudFront to use specific ciphers. Enforce the ALB with an HTTPS listener only and select the appropriate security policy for the ciphers Link the ALB with AWS WAF to allow access from the CloudFront IP ranges.

Set up an S3 bucket policy with the aws:securetransport key. Configure the CloudFront origin access identity (OAI) with the S3 bucket. Enforce the ALB with an HTTPS listener only and select the appropriate security policy for the ciphers.

Modify the CloudFront distribution to use AWS WAF. Force HTTPS on the S3 bucket with specific ciphers in the bucket policy. Configure an HTTPS listener only for the ALB. Set up a security group to limit access to the ALB from the CloudFront IP ranges

Modify the CloudFront distribution to use the ALB as the origin. Enforce an HTTPS listener on the ALB. Create a path-based routing rule on the ALB with proxies that connect lo Amazon S3. Create a bucket policy to allow access from these proxies only.

A company Is trying to replace its on-premises bastion hosts used to access on-premises Linux servers with AWS Systems Manager Session Manager. A security engineer has installed the Systems Manager Agent on all servers. The security engineer verifies that the agent is running on all the servers, but Session Manager cannot connect to them. The security engineer needs to perform verification steps before Session Manager will work on the servers.

Which combination of steps should the security engineer perform? (Select THREE.)

Options:

A) Open inbound port 22 to 0 0.0.0/0 on all Linux servers.

B) Enable the advanced-instances tier in Systems Manager.

C) Create a managed-instance activation for the on-premises servers.

D) Reconfigure the Systems Manager Agent with the activation code and ID.

E) Assign an IAM role to all of the on-premises servers.

F) Initiate an inventory collection with Systems Manager on the on-premises servers

Answer:

C, E, F

Question 4

Question Type: **MultipleChoice**

A company has several workloads running on AWS Employees are required to authenticate using on-premises ADFS and SSO to access the AWS Management Console Developers migrated an existing legacy web application to an Amazon EC2 instance Employees need to access this application from anywhere on the internet but currently, mere is no authentication system but into the application.

How should the Security Engineer implement employee-only access to this system without changing the application?

Options:

A) Place the application behind an Application Load Balancer (ALB) Use Amazon Cognito as authentication (or the ALB Define a SAML-based Amazon Cognito user pool and connect it to ADFS
implement AWS SSO in the master account and link it to ADFS as an identity provide' Define the EC2 instance as a managed resource, then apply an IAM policy on the resource

B) Define an Amazon Cognito identity pool then install the connector on the Active Directory server Use the Amazon Cognito SDK on the application instance to authenticate the employees using their C. Active Directory user names and passwords

D) Create an AWS Lambda custom authorizer as the authenticator for a reverse proxy on Amazon EC2 Ensure the security group on Amazon EC2 only allows access from the Lambda function.

Answer:

B

Question 5

Question Type: MultipleChoice

A recent security audit identified that a company's application team injects database credentials into the environment variables of an AWS Fargate task. The company's security policy mandates that all sensitive data be encrypted at rest and in transit.

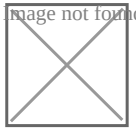
When combination of actions should the security team take to make the application compliant within the security policy? (Select THREE)

Store the credentials securely in a file in an Amazon S3 bucket with restricted access to the application team IAM role Ask the application team to read the credentials from the S3 object instead

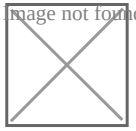
Create an AWS Secrets Manager secret and specify the key/value pairs to be stored in this secret

Modify the application to pull credentials from the AWS Secrets Manager secret instead of the environment variables.

Add the following statement to the container instance IAM role policy



Add the following statement to the execution role policy.



Log in to the AWS Fargate instance, create a script to read the secret value from AWS Secret Manager, and inject the environment variables. Ask the application team to redeploy the application.

Options:

A) Option A

B) Option B

C) Option C

D) Option D

E) Option E

F) Option F

Answer:

B, E, F

Question 6

Question Type: MultipleChoice

A security engineer to ensure their company's uses of AWS meets AWS security best practices. As part of this, the AWS account root user must not be used for daily work. The root user must be monitored for use, and the Security team must be alerted as quickly as possible if the root user is used.

Which solution meets these requirements?

Options:

A) Set up an Amazon CloudWatch Events rule that triggers an Amazon SNS notification.

B) Set up an Amazon CloudWatch Events rule that triggers an Amazon SNS notification logs from S3 and generate notifications using Amazon SNS.

C) Set up a rule in AWS config to trigger root user events. Trigger an AWS Lambda function and generate notifications using Amazon SNS.

D) Use Amazon Inspector to monitor the usage of the root user and generate notifications using Amazon SNS

Answer:

A

Question 7

Question Type: MultipleChoice

A company's Security Engineer has been asked to monitor and report all AWS account root user activities

Which of the following would enable the Security Engineer to monitor and report all root user activities? (Select TWO)

Options:

A) Configuring AWS Organizations to monitor root user API calls on the paying account

B) Creating an Amazon CloudWatch Events rule that will trigger when any API call from the root user is reported

C) Configuring Amazon Inspector to scan the AWS account for any root user activity

D) Configuring AWS Trusted Advisor to send an email to the Security team when the root user logs in to the console

E) Using Amazon SNS to notify the target group

Answer:

B, E

Question 8

Question Type: **MultipleChoice**

A company's Director of information Security wants a daily email report from AWS that contains recommendations for each company account to meet AWS Security best practices

Which solution would meet these requirements?

Options:

A) in every AWS account, configure AWS Lambda to query me AWS Support API tor AWS Trusted Advisor security checks Send the results from Lambda to an Amazon SNS topic to send reports.

B) Configure Amazon GuardDuty in a master account and invite all other accounts to be managed by the master account Use GuardDuty's integration with Amazon SNS to report on findings

C) Use Amazon Athena and Amazon QuickSight to build reports off of AWS CloudTrail Create a daily Amazon CloudWatch trigger to run the report dally and email It using Amazon SNS

C) Use AWS Artifact's prebuilt reports and subscriptions Subscribe the Director of Information Security to the reports by adding the Director as the security alternate contact tor each account

Answer:

A

Question 9

Question Type: **MultipleChoice**

Which of the following bucket policies will ensure that objects being uploaded to a bucket called 'demo' are encrypted.

Please select:

Options:

A) Option

```
"Version":"2012-10-17",
"Id":"PutObj",
"Statement":[{
"Sid":"DenyUploads",
"Effect":"Deny",
"Principal":"*",
"Action":"s3:PutObjectEncrypted",
"Resource":"arn:aws:s3:::demo/*"
}
}
]
}
```

B) Option

```
"Version":"2012-10-17",
"Id":"PutObj",
"Statement":[{
"Sid":"DenyUploads",
"Effect":"Deny",
"Principal":"*",
"Action":"s3:PutObjectEncrypted",
"Resource":"arn:aws:s3:::demo/*"
}
}
]
}
```

C) Option

```
"Version":"2012-10-17",
"Id":"PutObj",
"Statement":[{
"Sid":"DenyUploads",
"Effect":"Deny",
"Principal":"*",
"Action":"s3:PutObjectEncrypted",
"Resource":"arn:aws:s3:::demo/*"
}
}
]
}
```

D) Option

```
"Version":"2012-10-17",
"Id":"PutObj",
"Statement":[{
"Sid":"DenyUploads",
"Effect":"Deny",
"Principal":"*",
"Action":"s3:PutObjectEncrypted",
"Resource":"arn:aws:s3:::demo/*"
}
}
]
}
```

Answer:

A

Explanation:

The condition of 's3:x-amz-server-side-encryption':'aws:kms' ensures that objects uploaded need to be encrypted.

Options B,C and D are invalid because you have to ensure the condition of ns3:x-amz-server-side-encryption':'aws:kms' is present

For more information on AWS KMS best practices, just browse to the below URL:

https://dl.awsstatic.com/whitepapers/aws-kms-best-praaices.pdf

```
The correct answer is: {
 "Version":"2012-10-17",
 "Id":"PutObj",
 "Statement":[{
 "Sid":"DenyUploads",
 "Effect":"Deny",
 "Principal":"*",
 "Action":"s3:PutObject",
 "Resource":"arn:aws:s3:::demo/*",
 "Condition":{
 "StringNotEquals":{
 "s3:x-amz-server-side-encryption":"aws:kms"
 }
 }
 }
 ]
}
```

Submit your Feedback/Queries to our Expert