DOP-C02 Q&As

AWS Certified DevOps Engineer - Professional

Pass Amazon DOP-C02 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass2lead.com/dop-c02.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Amazon Official Exam Center

QUESTION 1

Your application stores sensitive information on an EBS volume attached to your EC2 instance. How can you protect your information? (Choose two.)

A. Unmount the EBS volume, take a snapshot and encrypt the snapshot. Re-mount the Amazon EBS volume.

B. It is not possible to encrypt an EBS volume, you must use a lifecycle policy to transfer data to S3 for encryption.

C. Copy the unencrypted snapshot and check the box to encrypt the new snapshot. Volumes restored from this encrypted snapshot will also be encrypted.

D. Create and mount a new, encrypted Amazon EBS volume. Move the data to the new volume. Delete the old Amazon EBS volume.

Correct Answer: CD

These steps are given in the AWS documentation

To migrate data between encrypted and unencrypted volumes

1) Create your destination volume (encrypted or unencrypted, depending on your need).

2) Attach the destination volume to the instance that hosts the data to migrate.

3) Make the destination volume available by following the procedures in Making an Amazon EBS Volume Available for Use. For Linux instances, you can create a mount point at /mnt/destination and mount the destination volume there.

4) Copy the data from your source directory to the destination volume. It may be most convenient to use a bulk-copy utility for this.

To encrypt a volume\\'s data by means of snapshot copying

1) Create a snapshot of your unencrypted CBS volume. This snapshot is also unencrypted.

2) Copy the snapshot while applying encryption parameters. The resulting target snapshot is encrypted.

3) Restore the encrypted snapshot to a new volume, which is also encrypted.

QUESTION 2

A company has chosen AWS to host a new application. The company needs to implement a multi-account strategy. A DevOps engineer creates a new AWS account and an organization in AWS Organizations. The DevOps engineer also creates the OU structure for the organization and sets up a landing zone by using AWS Control Tower.

The DevOps engineer must implement a solution that automatically deploys resources for new accounts that users create through AWS Control Tower Account Factory. When a user creates a new account, the solution must apply AWS CloudFormation templates and SCPs that are customized for the OU or the account to automatically deploy all the resources that are attached to the account. All the OUs are enrolled in AWS Control Tower.

Which solution will meet these requirements in the MOST automated way?

A. Use AWS Service Catalog with AWS Control Tower. Create portfolios and products in AWS Service Catalog. Grant granular permissions to provision these resources. Deploy SCPs by using the AWS CLI and JSON documents.

B. Deploy CloudFormation stack sets by using the required templates. Enable automatic deployment. Deploy stack instances to the required accounts. Deploy a CloudFormation stack set to the organization\\'s management account to deploy SCPs.

C. Create an Amazon EventBridge rule to detect the CreateManagedAccount event. Configure AWS Service Catalog as the target to deploy resources to any new accounts. Deploy SCPs by using the AWS CLI and JSON documents.

D. Deploy the Customizations for AWS Control Tower (CfCT) solution. Use an AWS CodeCommit repository as the source. In the repository, create a custom package that includes the CloudFormation templates and the SCP JSON documents.

Correct Answer: D

The CfCT solution is designed for the exact purpose stated in the question. It extends the capabilities of AWS Control Tower by providing you with a way to automate resource provisioning and apply custom configurations across all AWS accounts created in the Control Tower environment. This enables the company to implement additional account customizations when new accounts are provisioned via the Control Tower Account Factory. The CloudFormation templates and SCPs can be added to a CodeCommit repository and will be automatically deployed to new accounts when they are created. This provides a highly automated solution that does not require manual intervention to deploy resources and SCPs to new accounts.

QUESTION 3

A company manages multiple AWS accounts by using AWS Organizations with OUS for the different business divisions, The company is updating their corporate network to use new IP address ranges. The company has 10 Amazon S3 buckets in different AWS accounts. The S3 buckets store reports for the different divisions. The S3 bucket configurations allow only private corporate network IP addresses to access the S3 buckets.

A DevOps engineer needs to change the range of IP addresses that have permission to access the contents of the S3 buckets The DevOps engineer also needs to revoke the permissions of two OUS in the company

Which solution will meet these requirements?

A. Create a new SCP that has two statements, one that allows access to the new range of IP addresses for all the S3 buckets and one that demes access to the old range of IP addresses for all the S3 buckets. Set a permissions boundary for the OrganzauonAccountAccessRole role In the two OUS to deny access to the S3 buckets.

B. Create a new SCP that has a statement that allows only the new range of IP addresses to access the S3 buckets. Create another SCP that denies access to the S3 buckets. Attach the second SCP to the two OUS

C. On all the S3 buckets, configure resource-based policies that allow only the new range of IP addresses to access the S3 buckets. Create a new SCP that denies access to the S3 buckets. Attach the SCP to the two OUs.

D. On all the S3 buckets, configure resource-based policies that allow only the new range of IP addresses to access the S3 buckets. Set a permissions boundary for the OrganizationAccountAccessRole role in the two OUS to deny access to the S3 buckets.

Correct Answer: C

A comprehensive and detailed explanation is: Option A is incorrect because creating a new SCP that has two statements, one that allows access to the new range of IP addresses for all the S3 buckets and one that denies access to the old range of IP addresses for all the S3 buckets, is not a valid solution. SCPs are not resource-based policies, and they cannot specify the S3 buckets or the IP addresses as resources or conditions. SCPs can only control the actions that can be performed by the principals in the organization, not the access to specific resources. Moreover, setting a permissions boundary for the OrganizationAccountAccessRole role in the two OUs to deny access to the S3 buckets is not sufficient to revoke the permissions of the two OUs, as there might be other roles or users in those OUs that can still access the S3 buckets. Option B is incorrect because creating a new SCP that has a statement that allows only the new range of IP addresses to access the S3 buckets is not a valid solution, for the same reason as option A. SCPs are not resource-based policies, and they cannot specify the S3 buckets or the IP addresses as resources or conditions. Creating another SCP that denies access to the S3 buckets and attaching it to the two OUs is also not a valid solution, as SCPs cannot specify the S3 buckets as resources either.

Option C is correct because it meets both requirements of changing the range of IP addresses that have permission to access the contents of the S3 buckets and revoking the permissions of two OUs in the company. On all the S3 buckets, configuring resource-based policies that allow only the new range of IP addresses to access the S3 buckets is a valid way to update the IP address ranges, as resource-based policies can specify both resources and conditions. Creating a new SCP that denies access to the S3 buckets and attaching it to the two OUs is also a valid way to revoke the permissions of those OUs, as SCPs can deny actions such as s3:PutObject or s3:GetObject on any resource. Option D is incorrect because setting a permissions boundary for the OrganizationAccountAccessRole role in the two OUs to deny access to the S3 buckets is not sufficient to revoke the permissions of the two OUs, as there might be other roles or users in those OUs that can still access the S3 buckets. A permissions boundary is a policy that defines the maximum permissions that an IAM entity can have. However, it does not revoke any existing permissions that are granted by other policies. References: AWS Organizations S3 Bucket Policies Service Control Policies Permissions Boundaries

QUESTION 4

A company wants to use AWS development tools to replace its current bash deployment scripts. The company currently deploys a LAMP application to a group of Amazon EC2 instances behind an Application Load Balancer (ALB). During the deployments, the company unit tests the committed application, stops and starts services, unregisters and re-registers instances with the load balancer, and updates file permissions. The company wants to maintain the same deployment functionality through the shift to using AWS services.

Which solution will meet these requirements?

A. Use AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy\\'s appspec.yml file to restart services, and deregister and register instances with the ALB. Use the appspec.yml file to update file permissions without a custom script.

B. Use AWS CodePipeline to move the application from the AWS CodeCommit repository to AWS CodeDeploy. Use CodeDeploy\\'s deployment group to test the application, unregister and re-register instances with the ALB. and restart services. Use the appspec.yml file to update file permissions without a custom script.

C. Use AWS CodePipeline to move the application source code from the AWS CodeCommit repository to AWS CodeDeploy. Use CodeDeploy to test the application. Use CodeDeploy\\'s appspec.yml file to restart services and update permissions without a custom script. Use AWS CodeBuild to unregister and re-register instances with the ALB.

D. Use AWS CodePipeline to trigger AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy\\'s appspec.yml file to restart services. Unregister and re-register the instances in the AWS CodeDeploy deployment group with the ALB. Update the appspec.yml file to update file permissions without a custom script.

Correct Answer: D

QUESTION 5

A company is using an organization in AWS Organizations to manage multiple AWS accounts. The company\\'s development team wants to use AWS Lambda functions to meet resiliency requirements and is rewriting all applications to work with Lambda functions that are deployed in a VPC. The development team is using Amazon Elastic Pile System (Amazon EFS) as shared storage in Account A in the organization.

The company wants to continue to use Amazon EPS with Lambda Company policy requires all serverless projects to be deployed in Account B.

A DevOps engineer needs to reconfigure an existing EFS file system to allow Lambda functions to access the data through an existing EPS access point.

Which combination of steps should the DevOps engineer take to meet these requirements? (Select THREE.)

A. Update the EFS file system policy to provide Account B with access to mount and write to the EFS file system in Account A.

B. Create SCPs to set permission guardrails with fine-grained control for Amazon EFS.

C. Create a new EFS file system in Account B Use AWS Database Migration Service (AWS DMS) to keep data from Account A and Account B synchronized.

D. Update the Lambda execution roles with permission to access the VPC and the EFS file system.

E. Create a VPC peering connection to connect Account A to Account B.

F. Configure the Lambda functions in Account B to assume an existing IAM role in Account A

Correct Answer: AEF

A Lambda function in one account can mount a file system in a different account. For this scenario, you configure VPC peering between the function VPC and the file system VPC. https://docs.aws.amazon.com/lambda/latest/dg/servicesefs.html https://aws.amazon.com/ru/blogs/storage/mount-amazon-efs-file-systems-cross-account-from-amazon-eks/

1.

Need to update the file system policy on EFS to allow mounting the file system into Account B. ## File System Policy $ cat file-system-policy.json { "Statement": [{ "Effect": "Allow", "Action": ["elasticfilesystem:ClientMount", "elasticfilesystem:ClientWrite"], "Principal": { "AWS": "arn:aws:iam:::root" # Replace with AWS account ID of EKS cluster } }] }

2.

Need VPC peering between Account A and Account B as the pre-requisite

3.

Need to assume cross-account IAM role to describe the mounts so that a specific mount can be chosen.

[Latest DOP-C02 Dumps](https://www.pass2lead.com/dop-c02.html) [DOP-C02 PDF Dumps](https://www.pass2lead.com/dop-c02.html) [DOP-C02 Braindumps](https://www.pass2lead.com/dop-c02.html)