Free Questions for DOP-C02 by certsinside

Shared by Mendez on 09-08-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: **MultipleChoice**

A company wants to use AWS Systems Manager documents to bootstrap physical laptops for developers The bootstrap code Is stored in GitHub A DevOps engineer has already created a Systems Manager activation, installed the Systems Manager agent with the registration code, and installed an activation ID on all the laptops.

Which set of steps should be taken next?

Options:

A- Configure the Systems Manager document to use the AWS-RunShellScnpt command to copy the files from GitHub to Amazon S3, then use the aws-downloadContent plugin with a sourceType of S3

B- Configure the Systems Manager document to use the aws-configurePackage plugin with an install action and point to the Git repository

C- Configure the Systems Manager document to use the aws-downloadContent plugin with a sourceType of GitHub and sourcelnfo with the repository details.

D- Configure the Systems Manager document to use the aws:softwarelnventory plugin and run the script from the Git repository

Answer:

C

Explanation:

Configure the Systems Manager Document to Use the aws-downloadContent Plugin with a sourceType of GitHub and sourceInfo with the Repository Details:

The aws-downloadContent plugin can download content from various sources, including GitHub, which is necessary for bootstrapping the laptops with the code stored in the GitHub repository.

```
schemaVersion: '2.2'

description: 'Download and run bootstrap script from GitHub'

mainSteps:

- action: aws:downloadContent

name: downloadBootstrapScript

inputs:

sourceType: GitHub

sourceInfo: '{'owner':'my-org','repository':'my-repo','path':'scripts/bootstrap.sh','getOptions':'branch:main'}'

destinationPath: /tmp/bootstrap.sh
```

```
- action: aws:runShellScript

name: runBootstrapScript

inputs:

runCommand:

- chmod +x /tmp/bootstrap.sh

- /tmp/bootstrap.sh
```

This setup ensures that the bootstrap code is downloaded from GitHub and executed on the laptops using Systems Manager.

AWS Systems Manager aws-downloadContent Plugin

Running Commands Using Systems Manager

Question 2

Question Type: MultipleChoice

A DevOps engineer needs to implement integration tests into an existing AWS CodePipelme CI/CD workflow for an Amazon Elastic Container Service (Amazon ECS) service. The CI/CD workflow retrieves new application code from an AWS CodeCommit repository and builds a container image. The CI/CD workflow then uploads the container image to Amazon Elastic Container Registry (Amazon

ECR) with a new image tag version.

The integration tests must ensure that new versions of the service endpoint are reachable and that vanous API methods return successful response data The DevOps engineer has already created an ECS cluster to test the service

Which combination of steps will meet these requirements with the LEAST management overhead? (Select THREE.)

Options:

A- Add a deploy stage to the pipeline Configure Amazon ECS as the action provider

B- Add a deploy stage to the pipeline Configure AWS CodeDeploy as the action provider

C- Add an appspec.yml file to the CodeCommit repository

D- Update the image build pipeline stage to output an imagedefinitions json file that references the new image tag.

E- Create an AWS Lambda function that runs connectivity checks and API calls against the service. Integrate the Lambda function with CodePipeline by using aLambda action stage

F- Write a script that runs integration tests against the service. Upload the script to an Amazon S3 bucket. Integrate the script in the S3 bucket with CodePipeline by using an S3 action stage.

Answer:

A, D, E

Explanation:

* Add a Deploy Stage to the Pipeline, Configure Amazon ECS as the Action Provider:

By adding a deploy stage to the pipeline and configuring Amazon ECS as the action provider, the pipeline can automatically deploy the new container image to the ECS cluster.

This ensures that the service is updated with the new image tag, making the new version of the service endpoint reachable.

* Update the Image Build Pipeline Stage to Output an imagedefinitions.json File that Reference the New Image Tag:

The imagedefinitions.json file provides the necessary information about the container images and their tags for the ECS task definitions.

Updating the pipeline to output this file ensures that the correct image version is deployed.

Example imagedefinitions.json

```
[

{

'name': 'container-name',

'imageUri': '123456789012.dkr.ecr.region.amazonaws.com/my-repo:my-tag'

}

]
```

* Reference: CodePipeline ECS Deployment

* Create an AWS Lambda Function that Runs Connectivity Checks and API Calls against the Service. Integrate the Lambda Function with CodePipeline by Using a Lambda Action Stage:

The Lambda function can perform the necessary integration tests by making connectivity checks and API calls to the deployed service endpoint.

Integrating this Lambda function into CodePipeline ensures that these tests are run automatically after deployment, providing near-real-time feedback on the new deployment's health.

Example Lambda function integration:

```
actions:
- name: TestService
actionTypeId:
category: Test
owner: AWS
provider: Lambda
runOrder: 2
configuration:
FunctionName: testServiceFunction
```

These steps ensure that the CI/CD workflow deploys the new container image to ECS, updates the image references, and performs integration tests, meeting the requirements with minimal management overhead.

Question 3

Question Type: MultipleChoice

A company has an application that stores data that includes personally Identifiable Information (Pll) In an Amazon S3 bucket All data Is encrypted with AWS Key Management Service (AWS KMS) customer managed keys. All AWS resources are deployed from an AWS Cloud Formation template.

A DevOps engineer needs to set up a development environment for the application in a different AWS account The data in the development environment's S3 bucket needs to be updated once a week from the production environment's S3 bucket.

The company must not move Pll from the production environment without anonymizmg the Pll first The data in each environment must be encrypted with different KMS customer managed keys.

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

Options:

A- Activate Amazon Macie on the S3 bucket In the production account Create an AWS Step Functions state machine to initiate a discovery job and redact all Pll before copying files to the S3 bucket in the development account. Give the state machine tasks decrypt permissions on the KMS key in the production account. Give the state machine tasks encrypt permissions on the KMS key in the development account

B- Set up S3 replication between the production S3 bucket and the development S3 bucket Activate Amazon Macie on the development S3 bucket Create an AWS Step Functions state machine to initiate a discovery job and redact all Pll as the files are copied to the development S3 bucket. Give the state machine tasks encrypt and decrypt permissions on the KMS key in the development account.

C- Set up an S3 Batch Operations job to copy files from the production S3 bucket to the development S3 bucket. In the development account, configure an
AWS Lambda function to redact all Pll. Configure S3 Object Lambda to use the Lambda function for S3 GET requests Give the Lambda function's 1AM role encrypt and decrypt permissions on the KMS key in the development account.

D- Create a development environment from the CloudFormatlon template in the development account. Schedule an Amazon EventBridge rule to start the AWS Step Functions state machine once a week

E- Create a development environment from the CloudFormation template in the development account. Schedule a cron job on an Amazon EC2 instance to run once a week to start the S3 Batch Operations job.

Answer:

A, D

Explanation:

Activate Amazon Macie on the Production S3 Bucket:

Macie can identify and protect sensitive data such as PII.

Create a Step Functions state machine to automate data discovery and redaction before copying it to the development environment.

Example Step Functions state machine:

```
{
'Comment': 'Anonymize PII and copy data',
'StartAt': 'MacieDiscoveryJob',
'States': {
'MacieDiscoveryJob': {
'Type': 'Task',
'Resource': 'arn:aws:states:::macie:startClassificationJob',
'End': true
}
}
}
```

Create a Development Environment from CloudFormation Template:

Deploy the development environment in a new account using the existing CloudFormation template.

Schedule an EventBridge rule to start the Step Functions state machine on a weekly basis.

EventBridge rule example:

```
{

'ScheduleExpression': 'rate(7 days)',

'StateMachineArn': 'arn:aws:states:<region>::stateMachine:AnonymizeAndCopyData'

}
```

By using Macie for data anonymization and Step Functions for automation, you ensure PII is properly handled before data transfer between environments.

Amazon Macie

AWS Step Functions

AWS CloudFormation Templates

Question 4

Question Type: **MultipleChoice**

A DevOps engineer has created an AWS CloudFormation template that deploys an application on Amazon EC2 instances The EC2 instances run Amazon Linux The application is deployed to the EC2 instances by using shell scripts that contain user dat

a. The EC2 instances have an 1AM instance profile that has an 1AM role with the AmazonSSMManagedlnstanceCore managed policy attached

The DevOps engineer has modified the user data in the CloudFormation template to install a new version of the application. The engineer has also applied the stack update. However, the application was not updated on the running EC2 instances. The engineer needs to ensure that the changes to the application are installed on the running EC2 instances.

Which combination of steps will meet these requirements? (Select TWO.)

Options:

A- Configure the user data content to use the Multipurpose Internet Mail Extensions (MIME) multipart format. Set the scripts-user parameter to always in the text/cloud-config section.

B- Refactor the user data commands to use the cfn-init helper script. Update the user data to install and configure the cfn-hup and cfn-mit helper scripts to monitor and apply the metadata changes

C- Configure an EC2 launch template for the EC2 instances. Create a new EC2 Auto Scaling group. Associate the Auto Scaling group with the EC2 launch template Use the AutoScalingScheduledAction update policy for the Auto Scaling group.

D- Refactor the user data commands to use an AWS Systems Manager document (SSM document). Add an AWS CLI command in the user data to use Systems Manager Run Command to apply the SSM document to the EC2 instances

E- Refactor the user data command to use an AWS Systems Manager document (SSM document) Use Systems Manager State Manager to create an association between the SSM document and the EC2 instances.

Answer:

B, E

Explanation:

Refactor User Data to Use cfn-init and cfn-hup:

cfn-init helps to bootstrap the instance, installing packages and starting services.

cfn-hup is a daemon that can monitor metadata changes and re-apply configurations when necessary.

Example user data script with cfn-init:

```
#!/bin/bash

yum update -y

yum install -y aws-cfn-bootstrap

/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --region ${AWS::Region}

/opt/aws/bin/cfn-hup
```

Use Systems Manager State Manager:

State Manager can automatically apply an AWS Systems Manager document to instances at regular intervals, ensuring configurations are kept up-to-date.

Steps:

Create an SSM document that installs and configures your application.

Use State Manager to associate this document with your EC2 instances.

Example SSM document:

```
{
'schemaVersion': '2.2',
'description': 'Install My Application',
'mainSteps': [
{
'action': 'aws:runShellScript',
'name': 'installApplication',
'inputs': {
'runCommand': [
'yum install -y my-application'

]

}

}

]

}
```

Create State Manager association:

```
aws ssm create-association --name 'InstallMyApplication' --instance-id <instance-id> --document-version '\$LATEST'
```

Using cfn-init and cfn-hup

AWS Systems Manager State Manager

Question 5

Question Type: MultipleChoice

A company gives its employees limited rights to AWS DevOps engineers have the ability to assume an administrator role. For tracking purposes, the security team wants to receive a near-real-time notification when the administrator role is assumed.

How should this be accomplished?

Options:

A- Configure AWS Config to publish logs to an Amazon S3 bucket Use Amazon Athena to query the logs and send a notification to the security team when the administrator role is assumed

B- Configure Amazon GuardDuty to monitor when the administrator role is assumed and send a notification to the security team

C- Create an Amazon EventBridge event rule using an AWS Management Console sign-in events event pattern that publishes a message to an Amazon SNS topic if the administrator role is assumed

D- Create an Amazon EventBridge events rule using an AWS API call that uses an AWS CloudTrail event pattern to invoke an AWS Lambda function that publishes a message to an Amazon SNS topic if the administrator role is assumed.

Answer:

D

Explanation:

* Create an Amazon EventBridge Rule Using an AWS CloudTrail Event Pattern:

AWS CloudTrail logs API calls made in your account, including actions performed by roles.

Create an EventBridge rule that matches CloudTrail events where the AssumeRole API call is made to assume the administrator role.

* Invoke an AWS Lambda Function:

Configure the EventBridge rule to trigger a Lambda function whenever the rule's conditions are met.

The Lambda function will handle the logic to send a notification.

* Publish a Message to an Amazon SNS Topic:

The Lambda function will publish a message to an SNS topic to notify the security team.

Subscribe the security team's email address to this SNS topic to receive real-time notifications.

Example EventBridge rule pattern:

```
{

'source': ['aws.cloudtrail'],

'detail-type': ['AWS API Call via CloudTrail'],

'detail': {

'eventSource': ['sts.amazonaws.com'],

'eventName': ['AssumeRole'],
'requestParameters': {
'roleArn': ['arn:aws:iam:::role/AdministratorRole']
}
}
}
```

Example Lambda function (Node.js) to publish to SNS:

```
const AWS = require('aws-sdk');
const sns = new AWS.SNS();
exports.handler = async (event) => {
const params = {
Message: `Administrator role assumed: ${JSON.stringify(event.detail)}`,
TopicArn: 'arn:aws:sns:<region>::<sns-topic>'
};
await sns.publish(params).promise();
};
```

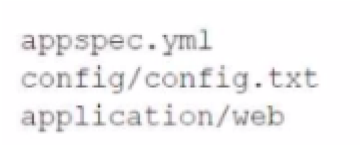
Creating EventBridge Rules

Using AWS Lambda with Amazon SNS

Question 6

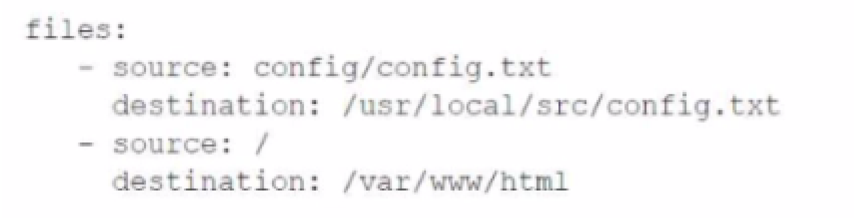
Question Type: MultipleChoice

A company deploys an application to Amazon EC2 instances. The application runs Amazon Linux 2 and uses AWS CodeDeploy. The application has the following file structure for its code repository:



```
appspec.yml
config/config.txt
application/web
```

The appspec.yml file has the following contents in the files section:



```
files:
   - source: config/config.txt
     destination: /usr/local/src/config.txt
   - source: /
     destination: /var/www/html
```

What will the result be for the deployment of the config.txt file?

Options:

A- The config.txt file will be deployed to only /var/www/html/config/config txt

B- The config.txt file will be deployed to /usr/local/src/config.txt and to /var/www/html/config/config txt.

C- The config.txt file will be deployed to only /usr/local/src/config txt

D- The config txt file will be deployed to /usr/local/src/config.txt and to /var/www/html/application/web/config txt

Answer:

C

Explanation:

Deployment of config.txt file based on the appspec.yml:

The appspec.yml file specifies that config/config.txt should be copied to /usr/local/src/config.txt.

The source: / directive in the appspec.yml indicates that the entire directory structure starting from the root of the application source should be copied to the specified destination, which is /var/www/html.

Result of the Deployment:

The config.txt file will be specifically deployed to /usr/local/src/config.txt as per the explicit file mapping.

The entire directory structure including application/web will be copied to /var/www/html, but this does not include config/config.txt since it has a specific destination defined.

Thus, the config.txt file will be deployed only to /usr/local/src/config.txt.

Therefore, the correct answer is:

C . The config.txt file will be deployed to only /usr/local/src/config.txt.

AWS CodeDeploy AppSpec File Reference

AWS CodeDeploy Deployment Process

Question 7

Question Type: **MultipleChoice**

A company has set up AWS CodeArtifact repositories with public upstream repositories The company's development team consumes open source dependencies from the repositories in the company's internal network.

The company's security team recently discovered a critical vulnerability in the most recent version of a package that the development team consumes. The security team has produced a patched version to fix the vulnerability. The company needs to prevent the vulnerable version from being downloaded. The company also needs to allow the security team to publish the patched version.

Which combination of steps will meet these requirements? {Select TWO.)

Options:

A- Update the status of the affected CodeArtifact package version to unlisted

B- Update the status of the affected CodeArtifact package version to deleted

C- Update the status of the affected CodeArtifact package version to archived.

D- Update the CodeArtifact package origin control settings to allow direct publishing and to block upstream operations

E- Update the CodeArtifact package origin control settings to block direct publishing and to allow upstream operations.

Answer:

B, D

Explanation:

Update the status of the affected CodeArtifact package version to deleted:

Deleting the vulnerable package version prevents it from being available for download by any users or systems, ensuring that the compromised version is not consumed.

Update the CodeArtifact package origin control settings to allow direct publishing and to block upstream operations:

By allowing direct publishing, the security team can publish the patched version of the package directly to the CodeArtifact repository.

Blocking upstream operations prevents the repository from automatically fetching and serving the vulnerable package version from upstream public repositories.

By deleting the vulnerable version and configuring the origin control settings to allow direct publishing and block upstream operations, the company ensures that only the patched version is available and the vulnerable version cannot be downloaded.

Managing Package Versions in CodeArtifact

Package Origin Controls in CodeArtifact

Question 8

Question Type: **MultipleChoice**

A company has an AWS Control Tower landing zone. The company's DevOps team creates a workload OU. A development OU and a production OU are nested under the workload OU. The company grants users full access to the company's AWS accounts to deploy applications.

The DevOps team needs to allow only a specific management 1AM role to manage the 1AM roles and policies of any AWS accounts In only the production OU.

Which combination of steps will meet these requirements? {Select TWO.)

Options:

A- Create an SCP that denies full access with a condition to exclude the management 1AM role for the organization root.

B- Ensure that the FullAWSAccess SCP is applied at the organization root

C- Create an SCP that allows IAM related actions Attach the SCP to the development OU

D- Create an SCP that denies IAM related actions with a condition to exclude the management I AM role Attach the SCP to the workload OU

E- Create an SCP that denies IAM related actions with a condition to exclude the management 1AM role Attach the SCP to the production OU

Answer:

B, E

Explanation:

You need to understand how SCP inheritance works in AWS. The way it works for Deny policies is different that allow policies.

Allow polices are passing down to children ONLY if they don't have an allow policy.

Deny policies always pass down to children.

That's why there is always an SCP set to the Root to allow everything by default. If you limit this policy, the whole organization will be limited, not matter what other policies are saying for the other OUs. So it's not A. It's not D because it restricts the wrong OU.

Question 9

Question Type: MultipleChoice

A company uses Amazon RDS for all databases in Its AWS accounts The company uses AWS Control Tower to build a landing zone that has an audit and logging account All databases must be encrypted at rest for compliance reasons. The company's security engineer needs to receive notification about any noncompliant databases that are in the company's accounts

Which solution will meet these requirements with the MOST operational efficiency?

Options:

A- Use AWS Control Tower to activate the optional detective control (guardrail) to determine whether the RDS storage is encrypted Create an Amazon Simple Notification Service (Amazon SNS) topic in the company's audit account. Create an Amazon EventBridge rule to filter noncompliant events from the AWS Control Tower control (guardrail) to notify the SNS topic. Subscribe the security engineer's email address to the SNS topic

B- Use AWS Cloud Formation StackSets to deploy AWS Lambda functions to every account. Write the Lambda function code to determine whether the RDS storage is encrypted in the account the function is deployed to Send the findings as an Amazon CloudWatch metric to the management account Create an Amazon Simple Notification Service (Amazon SNS) topic. Create a CloudWatch alarm that notifies the SNS topic when metric thresholds are met. Subscribe the security engineer's email address to the SNS topic.

C- Create a custom AWS Config rule in every account to determine whether the RDS storage is encrypted Create an Amazon Simple Notification Service (Amazon SNS) topic in the audit account Create an Amazon EventBridge rule to filter noncompliant events from the AWS Control Tower control (guardrail) to notify the SNS topic. Subscribe the security engineer's email address to the SNS topic

D- Launch an Amazon EC2 instance. Run an hourly cron job by using the AWS CLI to determine whether the RDS storage is encrypted in each AWS account Store the results in an RDS database. Notify the security engineer by sending email messages from the EC2 instance when noncompliance is detected

Answer:

A

Explanation:

Activate AWS Control Tower Guardrail:

Use AWS Control Tower to activate a detective guardrail that checks whether RDS storage is encrypted.

Create SNS Topic for Notifications:

Set up an Amazon Simple Notification Service (SNS) topic in the audit account to receive notifications about non-compliant databases.

Create EventBridge Rule to Filter Non-compliant Events:

Create an Amazon EventBridge rule that filters events related to the guardrail's findings on non-compliant RDS instances.

Configure the rule to send notifications to the SNS topic when non-compliant events are detected.

Subscribe Security Engineer's Email to SNS Topic:

Subscribe the security engineer's email address to the SNS topic to receive notifications when non-compliant databases are detected.

By using AWS Control Tower to activate a detective guardrail and setting up SNS notifications for non-compliant events, the company can efficiently monitor and ensure that all RDS databases are encrypted at rest.

AWS Control Tower Guardrails

Amazon SNS

Amazon EventBridge

Question 10

Question Type: MultipleChoice

A company is developing a web application's infrastructure using AWS CloudFormation The database engineering team maintains the database resources in a Cloud Formation template, and the software development team maintains the web application resources in a separate CloudFormation template. As the scope of the application grows, the software development team needs to use resources maintained by the database engineering team However, both teams have their own review and lifecycle management processes that they want to keep. Both teams also require resource-level change-set reviews. The software development team would like to deploy changes to this template using their CI/CD pipeline.

Which solution will meet these requirements?

Options:

A- Create a stack export from the database CloudFormation template and import those references into the web application CloudFormation template

B- Create a CloudFormation nested stack to make cross-stack resource references and parameters available in both stacks.

C- Create a CloudFormation stack set to make cross-stack resource references and parameters available in both stacks.

D- Create input parameters in the web application CloudFormation template and pass resource names and IDs from the database stack.

Answer:

A

Explanation:

* Stack Export and Import:

Use the Export feature in CloudFormation to share outputs from one stack (e.g., database resources) and use them as inputs in another stack (e.g., web application resources).

* Steps to Create Stack Export:

Define the resources in the database CloudFormation template and use the Outputs section to export necessary values.

Outputs:

DBInstanceEndpoint:

Value: !GetAtt DBInstance.Endpoint.Address

Export:

Name: DBInstanceEndpoint

Steps to Import into Web Application Stack:

In the web application CloudFormation template, use the ImportValue function to import these exported values.

Resources:

MyResource:

Type: 'AWS::SomeResourceType'

Properties:

SomeProperty: !ImportValue DBInstanceEndpoint

Resource-Level Change-Set Reviews:

Both teams can continue using their respective review processes, as changes to each stack are managed independently.

Use CloudFormation change sets to preview changes before deploying.

By exporting resources from the database stack and importing them into the web application stack, both teams can maintain their separate review and lifecycle management processes while sharing necessary resources.

AWS CloudFormation Export

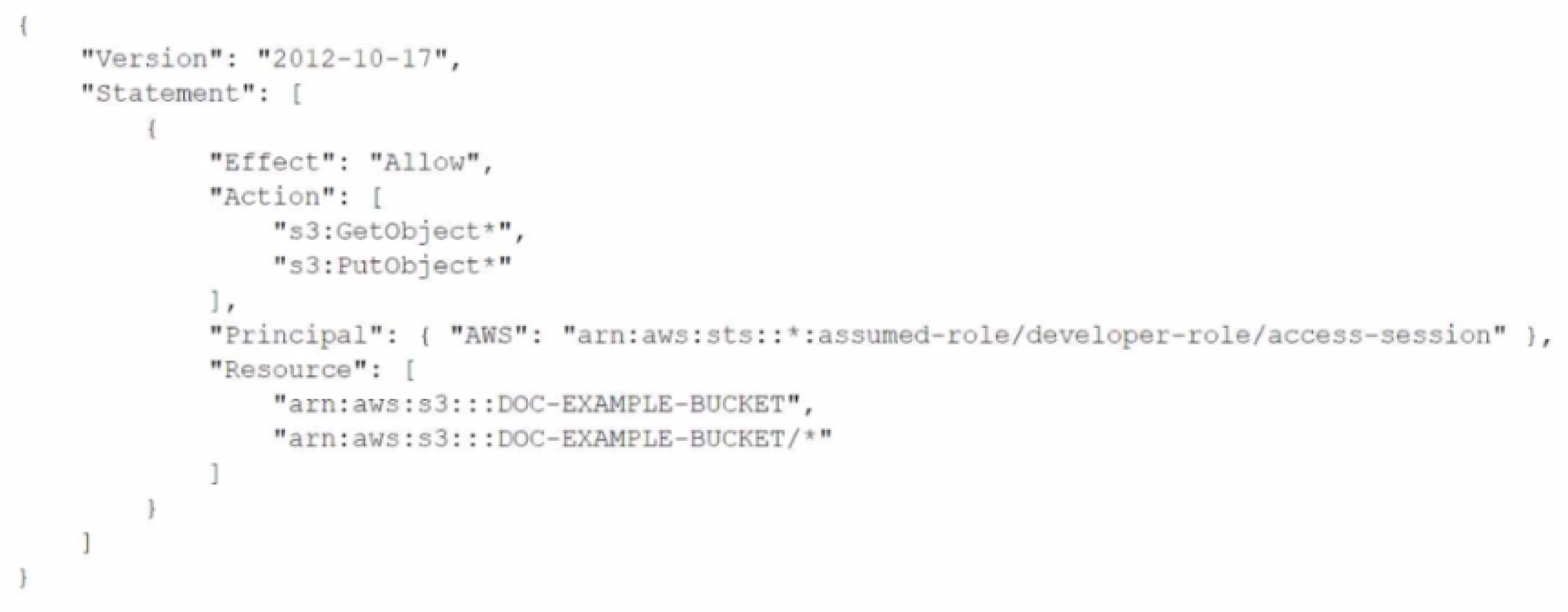
AWS CloudFormation ImportValue

Question 11

Question Type: MultipleChoice

A company has an organization in AWS Organizations. A DevOps engineer needs to maintain multiple AWS accounts that belong to different OUs in the organization. All resources, including 1AM policies and Amazon S3 policies within an account, are deployed through AWS CloudFormation. All templates and code are maintained in an AWS CodeCommit repository Recently, some developers have not been able to access an S3 bucket from some accounts in the organization.

The following policy is attached to the S3 bucket.



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:PutObject*"
            ],
            "Principal": { "AWS": "arn:aws:sts::*:assumed-role/developer-role/access-session" },
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

What should the DevOps engineer do to resolve this access issue?

Options:

A- Modify the S3 bucket policy Turn off the S3 Block Public Access setting on the S3 bucket In the S3 policy, add the awsSourceAccount condition. Add the AWS account IDs of all developers who are experiencing the issue.

B- Verify that no 1AM permissions boundaries are denying developers access to the S3 bucket Make the necessary changes to IAM permissions boundaries. Use an AWS Config recorder in the individual developer accounts that are experiencing the issue to revert any changes that are blocking access. Commit the fix back into the CodeCommit repository. Invoke deployment through Cloud Formation to apply the changes.

C- Configure an SCP that stops anyone from modifying 1AM resources in developer OUs. In the S3 policy, add the awsSourceAccount condition. Add the AWS account IDs of all developers who are experiencing the issue Commit the fix back into the CodeCommit repository Invoke deployment through CloudFormation to apply the changes

D- Ensure that no SCP is blocking access for developers to the S3 bucket Ensure that no 1AM policy permissions boundaries are denying access to developer 1AM users Make the necessary changes to the SCP and 1AM policy permissions boundaries in the CodeCommit repository Invoke deployment through CloudFormation to apply the changes

Answer:

D

Explanation:

Verify No SCP Blocking Access:

Ensure that no Service Control Policy (SCP) is blocking access for developers to the S3 bucket. SCPs are applied at the organization or organizational unit (OU) level in AWS Organizations and can restrict what actions users and roles in the affected accounts can perform.

Verify No IAM Policy Permissions Boundaries Blocking Access:

IAM permissions boundaries can limit the maximum permissions that a user or role can have. Verify that these boundaries are not restricting access to the S3 bucket.

Make Necessary Changes to SCP and IAM Policy Permissions Boundaries:

Adjust the SCPs and IAM permissions boundaries if they are found to be the cause of the access issue. Make sure these changes are reflected in the code maintained in the AWS CodeCommit repository.

Invoke Deployment Through CloudFormation:

Commit the updated policies to the CodeCommit repository.

Use AWS CloudFormation to deploy the changes across the relevant accounts and resources to ensure that the updated permissions are applied consistently.

By ensuring no SCPs or IAM policy permissions boundaries are blocking access and making necessary changes if they are, the DevOps engineer can resolve the access issue for developers trying to access the S3 bucket.

AWS SCPs

IAM Permissions Boundaries

Deploying CloudFormation Templates

To Get Premium Files for DOP-C02 Visit

https://www.p2pexams.com/products/dop-c02

For More Free Questions Visit

https://www.p2pexams.com/amazon/pdf/dop-c02