# New DOP-C02 Study Plan - DOP-C02 Online Version


BONUS!!! Download part of Easy4Engine DOP-C02 dumps for free: https://drive.google.com/open?id=1UiYrEsC6QcFTRPhj1UeTOKlwdT351AYw

The AWS Certified DevOps Engineer - Professional (DOP-C02) PDF format, desktop practice test software, and web-based practice test software, all three formats of actual exam questions are ready for quick download. You just need to pay the affordable Amazon DOP-C02 Exam Questions charges and click on the download button. Get them now and start AWS Certified DevOps Engineer - Professional (DOP-C02) exam preparation today.

The DOP-C02 Certification Exam is a challenging test that requires candidates to have a thorough understanding of DevOps principles and best practices, as well as hands-on experience working with AWS tools and services. Candidates must pass a two-hour, multiple-choice exam that consists of 75 questions covering a range of topics, including system automation, monitoring and logging, security and compliance, and infrastructure as code.

**>> New DOP-C02 Study Plan <<**

## Quiz 2026 Amazon DOP-C02: AWS Certified DevOps Engineer - Professional – Trustable New Study Plan

After you pass the test DOP-C02 certification, your working abilities will be recognized by the society and you will find a good job. If you master our DOP-C02 quiz torrent and pass the exam. You will be respected by your colleagues, your boss, your relatives, your friends and the society. All in all, buying our DOP-C02 Test Prep can not only help you pass the exam but also help realize your dream about your career and your future. So don't be hesitated to buy our DOP-C02 exam materials and take action immediately.

Amazon DOP-C02 (AWS Certified DevOps Engineer - Professional) certification exam is a challenging but rewarding certification for professionals who are looking to validate their skills and knowledge in the field of DevOps. It is a valuable credential that can help individuals advance their careers and organizations identify top talent in the field.

## Amazon AWS Certified DevOps Engineer - Professional Sample Questions (Q384-Q389):

**NEW QUESTION # 384**
A company uses a pipeline in AWS CodePipeline to upload AWS CloudFormation templates to an Amazon S3 bucket. The pipeline uses the templates to deploy CloudFormation stacks that match the names of the templates.
The company has experienced issues when it tries to revert templates to a previous version. To prevent these issues, the company must have the ability to review template modifications before the modifications are deployed to production.
Which solution will meet these requirements with the LEAST operational overhead?

- A. Configure a connection in AWS CodeConnections to a Git repository. Store the templates in the Git repository. Configure the pipeline to include a source action that uses the connection. Add a manual review action to the pipeline to review template modifications before the stack deployments.
- B. Add a manual review action in the pipeline to review modifications to the template code before the stack deployments.
- C. Configure a connection in AWS CodeConnections to a Git repository. Store the templates in the Git repository. Configure

a pull request workflow to review template modifications. Configure AWS CloudFormation Git sync for the stacks.

- D. Update the pipeline to invoke an AWS Lambda function to check the template modifications before the stack deployments.

**Answer: B**

Explanation:
The requirement is simply: review changes before production deployment, with the least operational overhead.
B is the lightest change: adding a Manual approval (review) action in CodePipeline creates a controlled gate before the deploy stage. It requires no new repositories, no new services, and no custom code-just pipeline configuration.
Why not the others:
A introduces additional moving parts (Git repo integration, PR workflow management, and CloudFormation Git sync). That's useful, but it's more operational overhead than necessary to satisfy "review before deploy." C requires custom Lambda logic to inspect templates and decide whether to proceed-more code to write, run, secure, and maintain.
D adds both Git integration and a manual approval step-again more overhead than just adding the approval gate.

**NEW QUESTION # 385**
A company's application teams use AWS CodeCommit repositories for their applications. The application teams have repositories in multiple AWS accounts. All accounts are in an organization in AWS Organizations.
Each application team uses AWS IAM Identity Center (AWS Single Sign-On) configured with an external IdP to assume a developer IAM role. The developer role allows the application teams to use Git to work with the code in the repositories.
A security audit reveals that the application teams can modify the main branch in any repository. A DevOps engineer must implement a solution that allows the application teams to modify the main branch of only the repositories that they manage.
Which combination of steps will meet these requirements? (Select THREE.)

- A. For each CodeCommit repository, add an access-team tag that has the value set to the name of the associated team.
- B. Create an approval rule template for each account. Associate the template with all repositories. Add the "aws:ResourceTag/access-team":"$ ;{aws:PrincipalTag/access-team}" condition to the approval rule template.
- C. Create an approval rule template for each team in the Organizations management account. Associate the template with all the repositories. Add the developer role ARN as an approver.
- D. Update the SAML assertion to pass the user's team name. Update the IAM role's trust policy to add an access-team session tag that has the team name.
- E. Attach an SCP to the accounts. Include the following statement: A computer code with text AI-generated content may be incorrect.
- F. Create an IAM permissions boundary in each account. Include the following statement: A computer code with black text AI-generated content may be incorrect.

**Answer: A,D,F**

Explanation:
Short To meet the requirements, the DevOps engineer should update the SAML assertion to pass the user's team name, update the IAM role's trust policy to add an access-team session tag that has the team name, create an IAM permissions boundary in each account, and for each CodeCommit repository, add an access-team tag that has the value set to the name of the associated team.
:
Updating the SAML assertion to pass the user's team name allows the DevOps engineer to use IAM tags to identify which team a user belongs to. This can help enforce fine-grained access control based on the user's team membership1.
Updating the IAM role's trust policy to add an access-team session tag that has the team name allows the DevOps engineer to use IAM condition keys to restrict access based on the session tag value2. For example, the DevOps engineer can use the aws:PrincipalTag condition key to match the access-team tag of the user with the access-team tag of the repository3.
Creating an IAM permissions boundary in each account allows the DevOps engineer to set the maximum permissions that an identity-based policy can grant to an IAM entity. An entity's permissions boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries4. For example, the DevOps engineer can use a permissions boundary policy to limit the actions that a user can perform on CodeCommit repositories based on their access-team tag5.
For each CodeCommit repository, adding an access-team tag that has the value set to the name of the associated team allows the DevOps engineer to use resource tags to identify which team manages a repository. This can help enforce fine-grained access control based on the resource tag value6.
The other options are incorrect because:
Creating an approval rule template for each team in the Organizations management account is not a valid option, as approval rule templates are not supported by AWS Organizations. Approval rule templates are specific to CodeCommit and can only be

associated with one or more repositories in the same AWS Region where they are created7.
Creating an approval rule template for each account is not a valid option, as approval rule templates are not designed to restrict access to modify branches. Approval rule templates are designed to require approvals from specified users or groups before merging pull requests8.
Attaching an SCP to the accounts is not a valid option, as SCPs are not designed to restrict access based on tags. SCPs are designed to restrict access based on service actions and resources across all users and roles in an organization's account9.

**NEW QUESTION # 386**
A company has an application that runs on Amazon EC2 instances in an Auto Scaling group. The application processes a high volume of messages from an Amazon Simple Queue Service (Amazon SQS) queue.
A DevOps engineer noticed that the application took several hours to process a group of messages from the SQS queue. The average CPU utilization of the Auto Scaling group did not cross the threshold of a target tracking scaling policy when processing the messages. The application that processes the SQS queue publishes logs to Amazon CloudWatch Logs.
The DevOps engineer needs to ensure that the queue is processed quickly.
Which solution meets these requirements with the LEAST operational overhead?

- A. Create an AWS Lambda function. Configure the Lambda function to publish a custom metric by using the ApproximateNumberOfMessagesVisible SQS queue attribute and the GroupIn-ServiceInstances Auto Scaling group attribute to publish the queue messages for each instance. Schedule an Amazon EventBridge rule to run the Lambda function every hour. Create a target tracking scaling policy for the Auto Scaling group that uses the custom metric to scale in and out.
- B. Create a target tracking scaling policy for the Auto Scaling group. In the target tracking policy, use the ApproximateNumberOfMessagesVisible SQS queue attribute and the GroupIn-ServiceInstances Auto Scaling group attribute to calculate how many messages are in the queue for each number of instances by using metric math. Use the calculated attribute to scale in and out.
- C. Create an AWS Lambda function. Configure the Lambda function to publish a custom metric by using the ApproximateNumberOfMessagesVisible SQS queue attribute and the GroupIn-ServiceInstances Auto Scaling group attribute to publish the queue messages for each instance. Create a CloudWatch subscription filter for the application logs with the Lambda function as the target. Create a target tracking scaling policy for the Auto Scaling group that uses the custom metric to scale in and out.
- D. Create an AWS Lambda function that logs the ApproximateNumberOfMessagesVisible attribute of the SQS queue to a CloudWatch Logs log group. Schedule an Amazon EventBridge rule to run the Lambda function every 5 minutes. Create a metric filter to count the number of log events from a CloudWatch logs group. Create a target tracking scaling policy for the Auto Scaling group that uses the custom metric to scale in and out.

**Answer: B**

Explanation:
Comprehensive and Detailed Explanation From Exact Extract:
The default CPU utilization metric does not reflect the processing backlog in the SQS queue, so the Auto Scaling group is not scaling properly to handle the workload.
To scale the Auto Scaling group based on queue length, you can create a target tracking scaling policy that uses a custom metric that combines the SQS queue's ApproximateNumberOfMessagesVisible and the number of instances (GroupIn-ServiceInstances) metric using CloudWatch metric math. This allows the scaling policy to calculate the average number of messages per instance and scale accordingly.
This approach requires no additional Lambda functions or log processing, thus minimizing operational overhead.
Option A and B require Lambda functions to publish custom metrics, which increases operational complexity. Option D also adds complexity with logging and metric filters.
Reference:
Scaling based on SQS queue length using metric math:
"You can create CloudWatch metric math expressions combining SQS and Auto Scaling group metrics to enable target tracking scaling policies that respond to queue backlog." (AWS Auto Scaling with SQS) Target Tracking Scaling Policies:
"Target tracking policies can use metric math expressions as a source to make scaling decisions." (AWS Auto Scaling Target Tracking)

**NEW QUESTION # 387**
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 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
- B. 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.
- 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.

**Answer: C**

Explanation:
The correct answer is 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

**NEW QUESTION # 388**
A company runs applications in AWS accounts that are in an organization in AWS Organizations The applications use Amazon EC2 instances and Amazon S3.
The company wants to detect potentially compromised EC2 instances suspicious network activity and unusual API activity in its existing AWS accounts and in any AWS accounts that the company creates in the future When the company detects one to these events the company wants to use an existing Amazon Simple Notification Service (Amazon SNS) topic to send a notification to its operational support team for investigation and remediation.
Which solution will meet these requirements in accordance with AWS best practices?

- A. In the organization's management account configure an AWS account as the AWS CloudTrail administrator account in the CloudTrail administrator account create a CloudTrail organization trail.

Add the company's existing AWS accounts to the organization trail Create an SCP that enables VPC Flow Logs in each account in the organization. Configure AWS Security Hub for the organization.Create an Amazon EventBridge rule with an event pattern to match Security Hub events and to forward matching events to the SNS topic.

- B. In the organization's management account. create an AWS CloudTrail organization trail Activate the organization trail in all AWS accounts in the organization. Create an SCP that enables VPC Flow Logs in each account in the organization. Configure AWS Security Hub for the organization Create an Amazon EventBridge rule with an even pattern to match Security Hub events and to forward matching events to the SNS topic.
- C. In the organization's management account configure an AWS account as the Amazon GuardDuty administrator account. In the GuardDuty administrator account add the company's existing AWS accounts to GuardDuty as members In the GuardDuty administrator account create an Amazon EventBridge rule with an event pattern to match GuardDuty events and to forward matching events to the SNS topic.
- D. In the organization's management account configure Amazon GuardDuty to add newly created AWS accounts by invitation and to send invitations to the existing AWS accounts Create an AWS Cloud Formation stack set that accepts the GuardDuty invitation and creates an Amazon EventBridge rule Configure the rule with an event pattern to match. GuardDuty events and to forward matching events to the SNS topic. Configure the Cloud Formation stack set to deploy into all AWS accounts in the organization.

**Answer: D**

Explanation:
It allows the company to detect potentially compromised EC2 instances, suspicious network activity, and unusual API activity in its existing AWS accounts and in any AWS accounts that the company creates in the future using Amazon GuardDuty. It also provides a solution for automatically adding future AWS accounts to GuardDuty by configuring GuardDuty to add newly created AWS accounts by invitation and to send invitations to the existing AWS accounts.

**NEW QUESTION # 389**
.....

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

BTW, DOWNLOAD part of Easy4Engine DOP-C02 dumps from Cloud Storage: https://drive.google.com/open?id=1UiYrEsC6QcFTRPhj1UeTOKlwdT351AYw