AWS DOP-C02

AWS-DevOps Certification Questions & Answers

Get Instant Access to Vital Exam Acing Materials | Study Guide | Sample Questions | Practice Test

DOP-C02

AWS Certified DevOps Engineer - Professional

75 Questions Exam – 750 / 1000 Cut Score – Duration of 180 minutes

Table of Contents:

Discover More about the DOP-C02 Certification

Are you interested in passing the AWS DOP-C02 exam? First discover, who benefits from the DOP-C02 certification. The DOP-C02 is suitable for a candidate if he wants to learn about Operations. Passing the DOP-C02 exam earns you the AWS Certified DevOps Engineer - Professional title.

While preparing for the DOP-C02 exam, many candidates struggle to get the necessary materials. But do not worry; your struggling days are over. The DOP-C02 PDF contains some of the most valuable preparation tips and the details and instant access to useful **DOP-C02 study materials just at one click**.

DOP-C02 AWS-DevOps Certification Details:

Exam Name	AWS DevOps Engineer Professional (AWS-DevOps)
Exam Code	DOP-C02
Exam Price	$300 USD
Duration	180 minutes
Number of Questions	75
Passing Score	750 / 1000
Recommended Training / Books	**DevOps Engineering on AWS**
Schedule Exam	**AWS Certification**
Sample Questions	**AWS DOP-C02 Sample Questions**
Recommended Practice	**AWS Certified DevOps Engineer - Professional Practice Test**

DOP-C02 Syllabus:

Section	**Objectives**
SDLC Automation - 22%	
Implement CI/CD pipelines.	Knowledge of: • Software development lifecycle (SDLC) concepts, phases, and models • Pipeline deployment patterns for single- and multi-account environments Skills in: • Configuring code, image, and artifact repositories

Section	Objectives
	• Using version control to integrate pipelines with application environments • Setting up build processes (for example, AWS CodeBuild) • Managing build and deployment secrets (for example, AWS Secrets Manager, AWS Systems Manager Parameter Store) • Determining appropriate deployment strategies (for example, AWS CodeDeploy)
Integrate automated testing into CI/CD pipelines.	Knowledge of: • Different types of tests (for example, unit tests, integration tests, acceptance tests, user interface tests, security scans) • Reasonable use of different types of tests at different stages of the CI/CD pipeline Skills in: • Running builds or tests when generating pull requests or code merges (for example, AWS CodeCommit, CodeBuild) • Running load/stress tests, performance benchmarking, and application testing at scale • Measuring application health based on application exit codes • Automating unit tests and code coverage • Invoking AWS services in a pipeline for testing
Build and manage artifacts.	Knowledge of: • Artifact use cases and secure management • Methods to create and generate artifacts • Artifact lifecycle considerations Skills in: • Creating and configuring artifact repositories (for example, AWS CodeArtifact, Amazon S3, Amazon Elastic Container Registry [Amazon ECR]) • Configuring build tools for generating artifacts (for example, CodeBuild, AWS Lambda) • Automating Amazon EC2 instance and container image build processes (for example, EC2 Image Builder)
Implement deployment strategies for instance, container, and	Knowledge of:

Section	**Objectives**
serverless environments.	• Deployment methodologies for various platforms (for example, Amazon EC2, Amazon Elastic Container Service [Amazon ECS], Amazon Elastic Kubernetes Service [Amazon EKS], Lambda) • Application storage patterns (for example, Amazon Elastic File System [Amazon EFS], Amazon S3, Amazon Elastic Block Store [Amazon EBS]) • Mutable deployment patterns in contrast to immutable deployment patterns • Tools and services available for distributing code (for example, CodeDeploy, EC2 Image Builder) Skills in: • Configuring security permissions to allow access to artifact repositories (for example, AWS Identity and Access Management [IAM], CodeArtifact) • Configuring deployment agents (for example, CodeDeploy agent) • Troubleshooting deployment issues • Using different deployment methods (for example, blue/green, canary)
Configuration Management and IaC - 17%	
Define cloud infrastructure and reusable components to provision and manage systems throughout their lifecycle.	Knowledge of: • Infrastructure as code (IaC) options and tools for AWS • Change management processes for IaC-based platforms • Configurations management services and strategies Skills in: • Composing and deploying IaC templates (for example, AWS Serverless Application Model [AWS SAM], AWS CloudFormation, AWS Cloud Development Kit [AWS CDK]) • Applying AWS CloudFormation StackSets across multiple accounts and AWS Regions • Determining optimal configuration management services (for example, AWS OpsWorks, AWS Systems Manager, AWS Config, AWS AppConfig) • Implementing infrastructure patterns, governance controls, and security standards into reusable IaC

Section	**Objectives**
	templates (for example, AWS Service Catalog, CloudFormation modules, AWS CDK)
Deploy automation to create, onboard, and secure AWS accounts in a multi account/multi-Region environment.	Knowledge of: • AWS account structures, best practices, and related AWS services Skills in: • Standardizing and automating account provisioning and configuration • Creating, consolidating, and centrally managing accounts (for example, AWS Organizations, AWS Control Tower) • Applying IAM solutions for multi-account and complex organization structures (for example, SCPs, assuming roles) • Implementing and developing governance and security controls at scale (AWS Config, AWS Control Tower, AWS Security Hub, Amazon Detective, Amazon GuardDuty, AWS Service Catalog, SCPs)
Design and build automated solutions for complex tasks and large-scale environments.	Knowledge of: • AWS services and solutions to automate tasks and processes • Methods and strategies to interact with the AWS software-defined infrastructure Skills in: • Automating system inventory, configuration, and patch management (for example, Systems Manager, AWS Config) • Developing Lambda function automations for complex scenarios (for example, AWS SDKs, Lambda, AWS Step Functions) • Automating the configuration of software applications to the desired state (for example, OpsWorks, Systems Manager State Manager) • Maintaining software compliance (for example, Systems Manager)
Resilient Cloud Solutions - 15%	
Implement highly available solutions to	Knowledge of:

Section	Objectives
meet resilience and business requirements.	- Multi-AZ and multi-Region deployments (for example, compute layer, data layer) - SLAs - Replication and failover methods for stateful services - Techniques to achieve high availability (for example, Multi-AZ, multi-Region) Skills in: - Translating business requirements into technical resiliency needs - Identifying and remediating single points of failure in existing workloads - Enabling cross-Region solutions where available (for example, Amazon DynamoDB, Amazon RDS, Amazon Route 53, Amazon S3, Amazon CloudFront) - Configuring load balancing to support cross-AZ services - Configuring applications and related services to support multiple Availability Zones and Regions while minimizing downtime
Implement solutions that are scalable to meet business requirements.	Knowledge of: - Appropriate metrics for scaling services - Loosely coupled and distributed architectures - Serverless architectures - Container platforms Skills in: - Identifying and remediating scaling issues - Identifying and implementing appropriate auto scaling, load balancing, and caching solutions - Deploying container-based applications (for example, Amazon ECS, Amazon EKS) - Deploying workloads in multiple AWS Regions for global scalability - Configuring serverless applications (for example, Amazon API Gateway, Lambda, AWS Fargate)
Implement automated recovery processes to meet RTO/RPO requirements.	Knowledge of: - Disaster recovery concepts (for example, RTO, RPO)

Section	**Objectives**
	• Backup and recovery strategies (for example, pilot light, warm standby) • Recovery procedures Skills in: • Testing failover of Multi-AZ/multi-Region workloads (for example, Amazon RDS, Amazon Aurora, Route 53, CloudFront) • Identifying and implementing appropriate cross-Region backup and recovery strategies (for example, AWS Backup, Amazon S3, Systems Manager) • Configuring a load balancer to recover from backend failure
Monitoring and Logging - 15%	
Configure the collection, aggregation, and storage of logs and metrics.	Knowledge of: • How to monitor applications and infrastructure • Amazon CloudWatch metrics (for example, namespaces, metrics, dimensions, and resolution) • Real-time log ingestion • Encryption options for at-rest and in-transit logs and metrics (for example, client-side and server-side, AWS Key Management Service [AWS KMS]) • Security configurations (for example, IAM roles and permissions to allow for log collection) Skills in: • Securely storing and managing logs • Creating CloudWatch metrics from log events by using metric filters • Creating CloudWatch metric streams (for example, Amazon S3 or Amazon Kinesis Data Firehose options) • Collecting custom metrics (for example, using the CloudWatch agent) • Managing log storage lifecycles (for example, S3 lifecycles, CloudWatch log group retention) • Processing log data by using CloudWatch log subscriptions (for example, Kinesis, Lambda, Amazon OpenSearch Service) • Searching log data by using filter and pattern syntax or CloudWatch Logs Insights

Section	**Objectives**
	- Configuring encryption of log data (for example, AWS KMS)
Audit, monitor, and analyze logs and metrics to detect issues.	Knowledge of: - Anomaly detection alarms (for example, CloudWatch anomaly detection) - Common CloudWatch metrics and logs (for example, CPU utilization with Amazon EC2, queue length with Amazon RDS, 5xx errors with an Application Load Balancer) - Amazon Inspector and common assessment templates - AWS Config rules - AWS CloudTrail log events Skills in: - Building CloudWatch dashboards and Amazon QuickSight visualizations - Associating CloudWatch alarms with CloudWatch metrics (standard and custom) - Configuring AWS X-Ray for different services (for example, containers, API Gateway, Lambda) - Analyzing real-time log streams (for example, using Kinesis Data Streams) - Analyzing logs with AWS services (for example, Amazon Athena, CloudWatch Logs Insights)
Automate monitoring and event management of complex environments.	Knowledge of: - Event-driven, asynchronous design patterns (for example, S3 Event Notifications or Amazon EventBridge events to Amazon Simple Notification Service [Amazon SNS] or Lambda) - Capabilities of auto scaling a variety of AWS services (for example, EC2 Auto Scaling groups, RDS storage auto scaling, DynamoDB, ECS capacity provider, EKS autoscalers) - Alert notification and action capabilities (for example, CloudWatch alarms to Amazon SNS, Lambda, EC2 automatic recovery) - Health check capabilities in AWS services (for example, Application Load Balancer target groups, Route 53) Skills in:

Section	**Objectives**
	• Configuring solutions for auto scaling (for example, DynamoDB, EC2 Auto Scaling groups, RDS storage auto scaling, ECS capacity provider) • Creating CloudWatch custom metrics and metric filters, alarms, and notifications (for example, Amazon SNS, Lambda) • Configuring S3 events to process log files (for example, by using Lambda), and deliver log files to another destination (for example, OpenSearch Service, CloudWatch Logs) • Configuring EventBridge to send notifications based on a particular event pattern • Installing and configuring agents on EC2 instances (for example, AWS Systems Manager Agent [SSM Agent], CloudWatch agent) • Configuring AWS Config rules to remediate issues • Configuring health checks (for example, Route 53, Application Load Balancer)
Incident and Event Response - 14%	
Manage event sources to process, notify, and take action in response to events.	Knowledge of: • AWS services that generate, capture, and process events (for example, AWS Health, EventBridge, CloudTrail, CloudWatch Events) • Event-driven architectures (for example, fan out, event streaming, queuing) Skills in: • Integrating AWS event sources (for example, AWS Health, EventBridge, CloudTrail, CloudWatch Events) • Building event processing workflows (for example, Amazon Simple Queue Service [Amazon SQS], Kinesis, Amazon SNS, Lambda, Step Functions)
Implement configuration changes in response to events.	Knowledge of: • Fleet management services (for example, Systems Manager, AWS Auto Scaling) • Configuration management services (for example, AWS Config) Skills in: • Applying configuration changes to systems

Section	**Objectives**
	- Modifying infrastructure configurations in response to events - Remediating a non-desired system state
Troubleshoot system and application failures.	Knowledge of: - AWS metrics and logging services (for example, CloudWatch, X-Ray) - AWS service health services (for example, AWS Health, CloudWatch, Systems Manager OpsCenter) - Root cause analysis Skills in: - Analyzing failed deployments (for example, AWS CodePipeline, CodeBuild, CodeDeploy, CloudFormation, CloudWatch synthetic monitoring) - Analyzing incidents regarding failed processes (for example, auto scaling, Amazon ECS, Amazon EKS)
Security and Compliance - 17%	
Implement techniques for identity and access management at scale.	Knowledge of: - Appropriate usage of different IAM entities for human and machine access (for example, users, groups, roles, identity providers, identity-based policies, resource-based policies, session policies) - Identity federation techniques (for example, using IAM identity providers and AWS Single Sign-On) - Permission management delegation by using IAM permissions boundaries - Organizational SCPs Skills in: - Designing policies to enforce least privilege access - Implementing role-based and attribute-based access control patterns - Automating credential rotation for machine identities (for example, Secrets Manager) - Managing permissions to control access to human and machine identities (for example, enabling multi-factor authentication [MFA], AWS Security Token Service [AWS STS], IAM profiles)
Apply automation for security controls and data protection.	Knowledge of:

Section	Objectives
	• Network security components (for example, security groups, network ACLs, routing, AWS Network Firewall, AWS WAF, AWS Shield) • Certificates and public key infrastructure (PKI) • Data management (for example, data classification, encryption, key management, access controls) Skills in: • Automating the application of security controls in multi-account and multi-Region environments (for example, Security Hub, Organizations, AWS Control Tower, Systems Manager) • Combining security controls to apply defense in depth (for example, AWS Certificate Manager [ACM], AWS WAF, AWS Config, AWS Config rules, Security Hub, GuardDuty, security groups, network ACLs, Amazon Detective, Network Firewall) • Automating the discovery of sensitive data at scale (for example, Amazon Macie) • Encrypting data in transit and data at rest (for example, AWS KMS, AWS CloudHSM, ACM)
Implement security monitoring and auditing solutions.	Knowledge of: • Security auditing services and features (for example, CloudTrail, AWS Config, VPC Flow Logs, CloudFormation drift detection) • AWS services for identifying security vulnerabilities and events (for example, GuardDuty, Amazon Inspector, IAM Access Analyzer, AWS Config) • Common cloud security threats (for example, insecure web traffic, exposed AWS access keys, S3 buckets with public access enabled or encryption disabled) Skills in: • Implementing robust security auditing • Configuring alerting based on unexpected or anomalous security events • Configuring service and application logging (for example, CloudTrail, CloudWatch Logs) • Analyzing logs, metrics, and security findings

Broaden Your Knowledge with AWS DOP-C02 Sample Questions:

Question: 1

A company controls the source code for an application in AWS CodeCommit. The company is creating a CI/CD pipeline for the application by using AWS CodePipeline.

The pipeline must start automatically when changes occur to the main branch of the CodeCommit repository. Changes occur frequently every day, so the pipeline must be as responsive as possible.

What should a DevOps engineer do to meet these requirements?

a) Configure the pipeline to periodically check the repository’s main branch for changes. Start the pipeline when changes are detected.
b) Configure an Amazon EventBridge (Amazon CloudWatch Events) rule to detect changes to the repository’s main branch. Configure the pipeline to start in response to the changes.
c) Configure the repository to periodically run an AWS Lambda function. Configure the function to check the repository’s main branch and to start the pipeline when the function detects changes.
d) Configure the repository to publish a notification to an Amazon Simple Notification Service (Amazon SNS) topic when changes occur to the repository’s main branch. Subscribe the pipeline to the SNS topic.

Answer: b

Question: 2

A DevOps engineer needs to implement a blue/green deployment process for an application on AWS. The DevOps engineer must gradually shift the traffic between the environments. The application runs on Amazon EC2 instances behind an Application Load Balancer (ALB). The instances run in an Amazon EC2 Auto Scaling group. The application stores data on an Amazon RDS Multi-AZ DB instance. Amazon Route 53 provides external DNS.
Which combination of steps should the DevOps engineer take to meet these requirements? (Select THREE.)

a) Create a second Auto Scaling group behind the same ALB.
b) Create a second Auto Scaling group behind a second ALB.
c) In Route 53, create a second alias record that points to the new environment. Use a failover routing policy to choose between the two records.
d) In Route 53, create a second alias record that points to the new environment. Use a weighted routing policy to choose between the two records.
e) Configure the new EC2 instances to use the primary RDS DB instance.
f) Configure the new EC2 instances to use the standby RDS DB instance.

Answer: b, d, e

Question: 3

A company has a legacy API that runs on a fleet of Amazon EC2 instances behind a public Application Load Balancer (ALB). The ALB has access logging enabled and stores the access logs in Amazon S3. The API is available through the hostname api.example.com. The company uses Amazon Route 53 to manage the hostname.

Developers have rebuilt five of the API endpoints by using a different AWS Lambda function for each endpoint. A DevOps engineer wants to test the new versions of the Lambda functions with a limited number of random customers. To ensure compatibility with an existing log processing service, the test must not affect the ALB access logs.

How should the DevOps engineer perform the test to meet these requirements?

a) Add the five Lambda functions as targets to the existing target group for the EC2 instances. Set the weight in the target group of each Lambda function target to be less than the EC2 instance targets. Amend the default rule on the ALB to enable target group-level stickiness.
b) Create a single target group that includes all the Lambda functions as individual targets. On the ALB, create a new listener rule that includes a host header condition that matches the API endpoint's hostname. Add the target group to the listener rule. Specify a lower weight for the new target group than the weight of the default rule's target group.
c) Create a new ALB and a new target group for each Lambda function. Create a new listener rule that includes a host header condition that matches each of the endpoints and forwards traffic to the target groups. Create a new Route 53 alias record with a weight of 10. Update the existing Route 53 record for the api.example.com hostname with a weight of 90.
d) Create a new target group for each Lambda function. On the ALB, create new listener rules that include a path condition that matches each of the different endpoints. Set the rules to be weighted between the Lambda function target group for that endpoint and the instance-based target group.

Answer: d

Question: 4

A development team is designing an application that has a large customer base spread across three AWS Regions. The application will use an Amazon DynamoDB table that must be available in all three Regions to deliver low-latency data access. When the table is updated in one Region, the changes must seamlessly propagate to the other Regions.

How should a DevOps engineer configure the table to meet these requirements with the LEAST operational overhead?

a) Create a DynamoDB table in each of the three Regions. Give each table the same name.
b) Configure three DynamoDB tables in each of the three Regions. Use the AWS SDK for DynamoDB to synchronize data changes among the tables.
c) Configure a multi-Region, multi-active DynamoDB global table that includes the three Regions.
d) Use DynamoDB global tables to configure a primary table in one Region and a read replica in each of the other Regions.

Answer: c

Question: 5

A company runs an application on Amazon EC2 instances that use the latest version of the Amazon Linux 2 AMI. When server administrators apply new security patches, the server administrators manually remove affected instances from service, patch the instances, and place the instances back into service.

A new security policy requires the company to apply security patches within 7 days after patches are released. The company's security team must verify that all the EC2 instances are compliant with this policy. The patching must occur during a time that has the least impact on users.

Which solution will automate compliance with these requirements?

a) Configure an AWS CodeBuild project to download and apply patches to all the instances over SSH. Use an Amazon EventBridge (Amazon CloudWatch Events) scheduled rule to run the CodeBuild project during a maintenance window.
b) Use AWS Systems Manager Patch Manager to create a patch baseline. Create a script on the EC2 instances to use the AWS CLI to pull the latest patches from Patch Manager. Create a cron job to schedule the script to run during a maintenance window.
c) Create a script to apply any available security patches. Create a cron job to schedule the script to run during a maintenance window. Install the script and cron job on the application AMI. Redeploy the application.
d) Enlist all the EC2 instances in an AWS Systems Manager Patch Manager patch group. Use Patch Manager to create a patch baseline. Configure a maintenance window to apply the patch baseline.

Answer: d

Question: 6

A company uses AWS CloudTrail on all its AWS accounts and sends all trails to a centralized Amazon S3 bucket. The company sends specified events to a third-party logging tool by using S3 event notifications and an AWS Lambda function. The company has hired a security services provider to set up a security operations center.
The security services provider wants to receive the CloudTrail logs through an Amazon Simple Queue Service (Amazon SQS) queue. The company must continue to use S3 event notifications and the Lambda function to send events to the third-party logging tool.

What is the MOST operationally efficient way to meet these requirements?

a) Add an additional notification to the S3 bucket for all CreateObject events to send all objects to the SQS queue.
b) Replace the existing S3 event notification destination with an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the Lambda function and the SQS queue to the topic.
c) Replace the existing S3 event notification destination with an Amazon Kinesis data stream. Create consumers for the Lambda function and the SQS queue.
d) Configure the trail to send logs to Amazon CloudWatch Logs. Subscribe the SQS queue to the CloudWatch Logs log group.

Answer: b

Question: 7

A company is reviewing its AWS account security policies. The company has staff members in different countries and wants to monitor its AWS accounts for unusual behavior that is associated with an IAM identity. The company wants to send a notification to any staff member for whom unusual activity is detected. The company also wants to send a notification to the user's team leader. An external messaging platform will send the notifications.

The platform requires a target user-id for each recipient. The company already has an API on AWS that the company can use to return the user-id of the staff member and the team leader from IAM user names. The company manages its AWS accounts by using AWS Organizations.

Which solution will meet these requirements?

a) Designate an account in the organization as the Amazon GuardDuty administrator. Add the company's AWS accounts as GuardDuty member accounts that are associated with the GuardDuty administrator account. Create an AWS Lambda function to perform the user-id lookup and to send notifications to the external messaging platform. Create an Amazon EventBridge (Amazon CloudWatch Events) rule in the GuardDuty administrator account to match the Impact:IAMUser/AnomalousBehavior notification type and invoke the Lambda function.
b) Designate an account in the organization as the Amazon Detective administrator. Add the company's AWS accounts as Detective member accounts that are associated with the Detective administrator account. Create an AWS Lambda function to perform the user-id lookup and to send notifications to the external messaging platform. Create an Amazon EventBridge (Amazon CloudWatch Events) rule in the Detective administrator account to match the Impact:IAMUser/AnomalousBehavior notification type and invoke the Lambda function.
c) Designate an account in the organization as the Amazon GuardDuty administrator. Add the company's AWS accounts as GuardDuty member accounts that are associated with the GuardDuty administrator account. Create an AWS Lambda function to perform the user-id lookup and to send notifications to the external messaging platform. Create an Amazon Simple Notification Service (Amazon SNS) topic in the GuardDuty administrator account to match the Impact:IAMUser/AnomalousBehavior notification type and invoke the Lambda function.
d) Designate an account in the organization as the Amazon Detective administrator. Add the company's AWS accounts as Detective member accounts that are associated with the Detective administrator account. Create an AWS Lambda function to perform the user-id lookup and to send notifications to the external messaging platform. Create an Amazon Simple Notification Service (Amazon SNS) topic in the Detective administrator account to match the Impact:IAMUser/AnomalousBehavior notification type and invoke the Lambda function.

Answer: a

Question: 8

A company is using AWS CodeBuild to build an application. Company policy requires all build artifacts to be encrypted at rest. The company must limit access to the artifacts to IAM users in an operations IAM group that have permission to assume an operations IAM role.

Which solution will meet these requirements?

a) Add a post-build command to the CodeBuild build specification to push build objects to an Amazon S3 bucket. Set a bucket policy that prevents upload to the bucket unless the request includes the x-amzserver-side-encryption header. Add a Deny statement for all actions with a NotPrincipal element that references the operations IAM group.
b) Add a post-build command to the CodeBuild build specification to push build objects to an Amazon S3 bucket. Configure an S3 event notification to invoke an AWS Lambda function to get the object, encrypt the object, and put the object back into the S3 bucket with a tag key of Encrypted and a tag value of True. Set a bucket policy with a Deny statement for all actions with a NotPrincipal element that references the operations IAM group. Include in the policy a Condition element that references the Encrypted tag.
c) Add a post-build command to the CodeBuild build specification to push build objects to an Amazon S3 bucket that has S3 default encryption enabled. Set a bucket policy that contains a Deny statement for all actions with a NotPrincipal element that references the operations IAM role.
d) Add a post-build command to the CodeBuild build specification to call the AWS Key Management Service (AWS KMS) Encrypt API operation and pass the artifact to AWS KMS for encryption with a specified KMS key. Push the encrypted artifact to an Amazon S3 bucket. Set up the operations IAM group as the only user for the specified KMS key.

Answer: c

Question: 9

A DevOps team has an application that stores critical company assets in an existing Amazon S3 bucket. The team uses a single AWS Region. A new company policy requires the team to deploy the application to multiple Regions. The assets must always be accessible. Users must use the same endpoint to access the assets.

Which combination of steps should the team take to meet these requirements in the MOST operationally efficient way?

(Select THREE.)

a) Use AWS CloudFormation StackSets to create a new S3 bucket that has versioning enabled in each required Region. Copy the assets from the existing S3 bucket to the new S3 buckets. Create an AWS Lambda function to copy files that are added to the new S3 bucket in the primary Region to the additional Regions.

b) Use AWS CloudFormation StackSets to create a new S3 bucket that has versioning enabled in each required Region. Create multiple S3 replication rules on the new S3 bucket in the primary Region to replicate all its contents to the additional Regions. Copy the assets from the existing S3 bucket to the new S3 bucket in the primary Region.

c) Create an Amazon CloudFront distribution. Configure new origins for each S3 bucket. Create an origin group that contains all the newly created origins. Update the default behavior of the distribution to use the new origin group.

d) Create an Amazon CloudFront distribution. Configure new origins for each S3 bucket. Create a Lambda@Edge function to validate the availability of the origin and to route the viewer request to an available nearby origin.

e) Create an Amazon Route 53 alias record. Configure a failover routing policy that uses the newly created S3 buckets as a target.

f) Create an Amazon Route 53 alias record. Configure a simple routing policy that uses the Amazon CloudFront distribution as a target.

Answer: b, c, f

Question: 10

A DevOps engineer is managing a legacy application on AWS. The application is a monolithic Windows program that runs on a single Amazon EC2 instance. The source code for the application is not available, so the application cannot be modified. The application has a memory leak and malfunctions when memory utilization on the EC2 instance increases to more than 90%.

The DevOps engineer has configured the unified Amazon CloudWatch agent on the EC2 instance to collect the operation system’s memory utilization metrics. The DevOps engineer needs to implement a solution to prevent the application from malfunctioning.

Which combination of steps will meet these requirements with the MOST operational efficiency?

(Select TWO.)

a) Create an Amazon EventBridge (Amazon CloudWatch Events) rule that publishes to an Amazon Simple Notification Service (Amazon SNS) topic when memory utilization increases to more than 80%.
b) Create a metric filter on memory utilization in Amazon CloudWatch Logs. Create a CloudWatch alarm on the memory utilization filter. Configure the alarm to publish to an Amazon Simple Notification Service (Amazon SNS) topic when the memory utilization increases to more than 80%.
c) Create a CloudWatch alarm on the memory utilization metric. Configure the alarm to publish to an Amazon Simple Notification Service (Amazon SNS) topic when the memory utilization increases to more than 80%.
d) Configure an AWS Lambda function to restart the application by using AWS Systems Manager Run Command. Subscribe the Lambda function to the Amazon Simple Notification Service (Amazon SNS) topic.
e) Configure the EC2 instance to run a script that restarts the application. Subscribe the EC2 instance to the Amazon Simple Notification Service (Amazon SNS) topic.

Answer: c, d

Avail the Study Guide to Pass DOP-C02 AWS-DevOps Exam:

- Find out about the DOP-C02 syllabus topics. Visiting the official site offers an idea about the exam structure and other important study resources. Going through the syllabus topics help to plan the exam in an organized manner.
- Once you are done exploring the **DOP-C02 syllabus**, it is time to plan for studying and covering the syllabus topics from the core. Chalk out the best plan for yourself to cover each part of the syllabus in a hassle-free manner.
- A study schedule helps you to stay calm throughout your exam preparation. It should contain your materials and thoughts like study hours, number of topics for daily studying mentioned on it. The best bet to clear the exam is to follow your schedule rigorously.
- The candidate should not miss out on the scope to learn from the DOP-C02 training. Joining the AWS provided training for DOP-C02 exam helps a candidate to strengthen his practical knowledge base from the certification.
- Learning about the probable questions and gaining knowledge regarding the exam structure helps a lot. Go through the **DOP-C02 sample questions** and boost your knowledge
- Make yourself a pro through online practicing the syllabus topics. DOP-C02 practice tests would guide you on your strengths and weaknesses regarding the syllabus topics. Through rigorous practicing, you can improve the weaker sections too. Learn well about time management during exam and become confident gradually with practice tests.

Career Benefits:

- Passing the DOP-C02 exam, helps a candidate to prosper highly in his career. Having the certification on the resume adds to the candidate's benefit and helps to get the best opportunities.