



Free Questions for DOP-C02
Shared by Sawyer on 16-04-2026

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)



Question 1

Question Type: MultipleChoice

A company is developing a web application and is using AWS CodeBuild for its CI/CD pipeline. The company must generate multiple artifacts from a single build process. The company also needs the ability to determine which build generated each artifact. The artifacts must be stored in an Amazon S3 bucket for further processing and deployment. Builds occur frequently and are based on a large Git repository. The company needs to optimize build times. Which solution will meet these requirements with the MOST operational efficiency?

Options:

- A-** Configure the buildspec.yml file to specify multiple artifacts with different file sets. Enable local caching for the build process by using source cache mode. Use environment variables to dynamically name artifacts based on the build ID.
- B-** Configure the buildspec.yml file to output all files as a single artifact. Enable local caching for the build process by using custom cache mode. Create an AWS Lambda function that is invoked by CodeBuild completion. Program the Lambda function to split the artifact into multiple files and to upload the files to the S3 bucket with dynamic names based on build ID.
- C-** Create separate CodeBuild projects for each artifact type. Enable local caching for the build process by using Docker layer cache mode. Configure each project to output a single artifact to the S3 bucket with a dynamic name based on build ID. Use AWS Step Functions to orchestrate the projects in parallel.
- D-** Set up CodeBuild to generate a single ZIP artifact that contains all files. Enable S3 caching for the build process. Use AWS CodePipeline with a custom action to extract the files and reorganize the files into multiple artifacts in the S3 bucket. Configure the custom action to dynamically name the files based on the time of the build.

Answer:

A

Question 2

Question Type: MultipleChoice

A company is migrating its web application to AWS. The application uses WebSocket connections for real-time updates and requires sticky sessions.

A DevOps engineer must implement a highly available architecture for the application. The application must be accessible to users worldwide with the least possible latency.

Which solution will meet these requirements with the LEAST operational overhead?

Options:

- A- Deploy an Application Load Balancer (ALB). Deploy another ALB in a different AWS Region. Enable cross-zone load balancing and sticky sessions on the ALBs. Integrate the ALBs with Amazon Route 53 latency-based routing.
- B- Deploy a Network Load Balancer (NLB). Deploy another NLB in a different AWS Region. Enable cross-zone load balancing and sticky sessions on the NLBs. Integrate the NLBs with Amazon Route 53 geolocation routing.
- C- Deploy a Network Load Balancer (NLB) with cross-zone load balancing enabled. Configure the NLB with IP-based targets in multiple Availability Zones. Use Amazon CloudFront for global content delivery. Implement sticky sessions by using source IP address preservation on the NLB.
- D- Deploy an Application Load Balancer (ALB) for HTTP traffic. Deploy a Network Load Balancer (NLB) in each of the company's AWS Regions for WebSocket connections. Enable sticky sessions on the ALB. Configure the ALB to forward requests to the NLB.

Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

ALB natively supports WebSocket protocols and sticky sessions via target group session affinity. Deploying ALBs in multiple regions with cross-zone load balancing ensures high availability and fault tolerance.

Using Route 53 latency-based routing allows users worldwide to connect to the lowest latency region, minimizing delay.

NLBs do not support sticky sessions and WebSocket protocol as well as ALBs do. Combining ALBs and NLBs (Option D) increases complexity. CloudFront (Option C) does not natively support WebSocket sticky sessions.

Option A is the simplest, most effective solution meeting all requirements with least operational overhead.

ALB WebSocket Support

Route 53 Latency Based Routing

Question 3

Question Type: MultipleChoice

A company has multiple development teams in different business units that work in a shared single AWS account. All Amazon EC2 resources that are created in the account must include tags that specify who created the resources. The tagging must occur within the first hour of resource creation.

A DevOps engineer needs to add tags to the created resources that include the user ID that created the resource and the cost center ID. The DevOps engineer configures an AWS Lambda Function with the cost center mappings to tag the resources. The DevOps engineer also sets up AWS CloudTrail in the AWS account. An Amazon S3 bucket stores the CloudTrail event logs.

Which solution will meet the tagging requirements?

Options:

- A- Create an S3 event notification on the S3 bucket to invoke the Lambda function for s3. ObjectTagging:Put events. Enable bucket versioning on the S3 bucket.
- B- Enable server access logging on the S3 bucket. Create an S3 event notification on the S3 bucket for s3. ObjectTagging.* events
- C- Create a recurring hourly Amazon EventBridge scheduled rule that invokes the Lambda function. Modify the Lambda function to read the logs from the S3 bucket.
- D- Create an Amazon EventBridge rule that uses Amazon EC2 as the event source. Configure the rule to match events delivered by CloudTrail. Configure the rule to target the Lambda function.

Answer:

B

Question 4

Question Type: MultipleChoice

A company used a lift-and-shift strategy to migrate a workload to AWS. The company has an Auto Scaling group of Amazon EC2 instances. Each EC2 instance runs a web application, a database, and a Redis cache.

Users are experiencing large variations in the web application's response times. Requests to the web application go to a single EC2 instance that is under significant load. The company wants to separate the application components to improve availability and performance.

Which solution will meet these requirements?

Options:

A- Create a Network Load Balancer and an Auto Scaling group for the web application. Migrate the database to an Amazon Aurora Serverless database. Create an Application Load Balancer and an Auto Scaling group for the Redis cache.

B- Create an Application Load Balancer and an Auto Scaling group for the web application. Migrate the database to an Amazon Aurora database that has a Multi-AZ deployment. Create a Network Load Balancer and an Auto Scaling group in a single Availability Zone for the Redis cache.

C- Create a Network Load Balancer and an Auto Scaling group for the web application. Migrate the database to an Amazon Aurora Serverless database. Create an Amazon ElastiCache (Redis OSS) cluster for the cache. Create a target group that has a DNS target type that contains the ElastiCache (Redis OSS) cluster hostname.

D- Create an Application Load Balancer and an Auto Scaling group for the web application. Migrate the database to an Amazon Aurora database that has a Multi-AZ deployment. Create an Amazon ElastiCache (Redis OSS) cluster for the cache.

Answer:

D

Question 5

Question Type: MultipleChoice

A company uses an AWS CodeCommit repository to store its source code and corresponding unit tests. The company has configured an AWS CodePipeline pipeline that includes an AWS CodeBuild project that runs when code is merged to the main branch of the repository.

The company wants the CodeBuild project to run the unit tests. If the unit tests pass, the CodeBuild project must tag the most recent commit.

How should the company configure the CodeBuild project to meet these requirements?

Options:

A- Configure the CodeBuild project to use native Git to clone the CodeCommit repository. Configure the project to run the unit tests. Configure the project to use native Git to create a tag and to push the Git tag to the repository if the code passes the unit tests.

B- Configure the CodeBuild project to use native Git to clone the CodeCommit repository. Configure the project to run the unit tests. Configure the project to use AWS CLI commands to create a new repository in the repository if the code passes the unit tests.

- C- Configure the CodeBuild project to use AWS CLI commands to copy the code from the CodeCommit repository. Configure the project to run the unit tests. Configure the project to use AWS CLI commands to create a new Git tag in the repository if the code passes the unit tests.
- D- Configure the CodeBuild project to use AWS CLI commands to copy the code from the CodeCommit repository. Configure the project to run the unit tests. Configure the project to use AWS CLI commands to create a new repository tag in the repository if the code passes the unit tests.

Answer:

A

Explanation:

Step 1: Using Native Git in CodeBuild To meet the requirement of running unit tests and tagging the most recent commit if the tests pass, the CodeBuild project should be configured to use native Git to clone the CodeCommit repository. Native Git support allows full functionality for managing the repository, including the ability to create and push tags.

Action: Configure the CodeBuild project to use native Git to clone the repository and run the tests.

Why: Using native Git provides flexibility for managing tags and other repository operations after the tests are successfully executed.

Step 2: Tagging the Most Recent Commit Once the unit tests pass, the CodeBuild project can use native Git to create a tag for the most recent commit and push that tag to the repository. This ensures that the tagged commit is linked to the test results.

Action: Configure the project to use native Git to create and push a tag to the repository if the tests pass.

Why: This ensures the correct commit is tagged automatically, streamlining the workflow.

This corresponds to Option A: Configure the CodeBuild project to use native Git to clone the CodeCommit repository. Configure the project to run the unit tests. Configure the project to use native Git to create a tag and to push the Git tag to the repository if the code passes the unit tests.

Question 6

Question Type: MultipleChoice

A company's production environment uses an AWS CodeDeploy blue/green deployment to deploy

an application. The deployment includes Amazon EC2 Auto Scaling groups that launch instances that run Amazon Linux 2.

A working `appspec.yml` file exists in the code repository and contains the following text.

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/application
```

A DevOps engineer needs to ensure that a script downloads and installs a license file onto the instances before the replacement instances start to handle request traffic. The DevOps engineer adds a `hooks` section to the `appspec.yml` file.

Which hook should the DevOps engineer use to run the script that downloads and installs the license file?

Options:

- A- AfterBlockTraffic
- B- BeforeBlockTraffic
- C- BeforeInstall
- D- Down load Bundle

Answer:

C

Explanation:

This hook runs before the new application version is installed on the replacement instances. This is the best place to run the script because it ensures that the license file is downloaded and installed before the replacement instances start to handle request traffic. If you use any other hook, you may encounter errors or inconsistencies in your application.

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>

20%
DISCOUNT

P2P
exams