



# DOP-C01<sup>Q&As</sup>

AWS Certified DevOps Engineer - Professional (DOP-C01)





## Pass Amazon DOP-C01 Exam with 100% Guarantee

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

<https://www.passapply.com/aws-devops-engineer-professional.html>

100% Passing Guarantee  
100% Money Back Assurance

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





### QUESTION 1

A DevOps Engineer is implementing a mechanism for canary testing an application on AWS. The application was recently modified and went through security, unit, and functional testing. The application needs to be deployed on an AutoScaling group and must use a Classic Load Balancer.

Which design meets the requirement for canary testing?

- A. Create a different Classic Load Balancer and Auto Scaling group for blue/green environments. Use Amazon Route 53 and create weighted A records on Classic Load Balancer.
- B. Create a single Classic Load Balancer and an Auto Scaling group for blue/green environments. Use Amazon Route 53 and create A records for Classic Load Balancer IPs. Adjust traffic using A records.
- C. Create a single Classic Load Balancer and an Auto Scaling group for blue/green environments. Create an Amazon CloudFront distribution with the Classic Load Balancer as the origin. Adjust traffic using CloudFront.
- D. Create a different Classic Load Balancer and Auto Scaling group for blue/green environments. Create an Amazon API Gateway with a separate stage for the Classic Load Balancer. Adjust traffic by giving weights to this stage.

Correct Answer: A

---

### QUESTION 2

A DevOps Engineer must track the health of a stateless RESTful service sitting behind a Classic ILoad Balancer. The deployment of new application revisions is through a CI/CD pipeline. If the service's latency increases beyond a defined threshold, deployment should be stopped until the service has recovered.

Which of the following methods allow for the QUICKEST detection time?

- A. Use Amazon CloudWatch metrics provided by Elastic Load Balancing to calculate average latency. Alarm and stop deployment when latency increases beyond the defined threshold.
- B. Use AWS Lambda and Elastic Load Balancing access logs to detect average latency. Alarm and stop deployment when latency increases beyond the defined threshold.
- C. Use AWS CodeDeploy's MinimumHealthyHosts setting to define thresholds for rolling back deployments. If these thresholds are breached, roll back the deployment.
- D. Use Metric Filters to parse application logs in Amazon CloudWatch Logs. Create a filter for latency. Alarm and stop deployment when latency increases beyond the defined threshold.

Correct Answer: C

---

### QUESTION 3

Which statement is true about configuring proxy support for Amazon Inspector agent on a Windows-based system?

- A. Amazon Inspector agent supports proxy usage on Windows-based systems through the use of the WinHTTP proxy.



- B. Amazon Inspector agent supports proxy usage on Linux-based systems but not on Windows.
- C. Amazon Inspector proxy support on Windows-based systems is achieved through installing proxy-enabled version of the agent which comes with preconfigured files that you need to edit to match your environment.
- D. Amazon Inspector agent supports proxy usage on Windows-based systems through awsagent.env configuration file.

Correct Answer: A

Proxy support for AWS agents is achieved through the use of the WinHTTP proxy.

Reference: [https://docs.aws.amazon.com/inspector/latest/userguide/inspector\\_agents-on-win.html#inspectoragent-proxy](https://docs.aws.amazon.com/inspector/latest/userguide/inspector_agents-on-win.html#inspectoragent-proxy)

#### QUESTION 4

A company is running a number of internet-facing APIs that use an AWS Lambda authorizer to control access. A security team wants to be alerted when a large number of requests are failing authorization, as this may indicate API abuse. Given the magnitude of API requests, the team wants to be alerted only if the number of HTTP 403 Forbidden responses goes above 2% of overall API calls.

Which solution will accomplish this?

- A. Use the default Amazon API Gateway 403Error and Count metrics sent to Amazon CloudWatch, and use metric math to create a CloudWatch alarm. Use the  $(403Error/Count)*100$  mathematical expression when defining the alarm. Set the alarm threshold to be greater than 2.
- B. Write a Lambda function that fetches the default Amazon API Gateway 403Error and Count metrics sent to Amazon CloudWatch, calculate the percentage of errors, then push a custom metric to CloudWatch named Custom403Percent. Create a CloudWatch alarm based on this custom metric. Set the alarm threshold to be greater than 2.
- C. Configure Amazon API Gateway to send custom access logs to Amazon CloudWatch Logs. Create a log filter to produce a custom metric for the HTTP 403 response code named Custom403Error. Use this custom metric and the default API Gateway Count metric sent to CloudWatch, and use metric math to create a CloudWatch alarm. Use the  $(Custom403Error/Count)*100$  mathematical expression when defining the alarm. Set the alarm threshold to be greater than 2.
- D. Configure Amazon API Gateway to enable custom Amazon CloudWatch metrics, enable the ALL\_STATUS\_CODE option, and define an APICustom prefix. Use CloudWatch metric math to create a CloudWatch alarm. Use the  $(APICustom403Error/Count)*100$  mathematical expression when defining the alarm. Set the alarm threshold to be greater than 2.

Correct Answer: C

Reference: <https://aws.amazon.com/blogs/compute/analyzing-api-gateway-custom-access-logs-for-custom-domain-names/>

#### QUESTION 5

You are building out a layer in a software stack on AWS that needs to be able to scale out to react to increased demand as fast as possible. You are running the code on EC2 instances in an Auto Scaling Group behind an ELB. Which application code deployment method should you use?



- A. SSH into new instances that come online, and deploy new code onto the system by pulling it from an S3 bucket, which is populated by code that you refresh from source control on new pushes.
- B. Bake an AMI when deploying new versions of code, and use that AMI for the Auto Scaling Launch Configuration.
- C. Create a Dockerfile when preparing to deploy a new version to production and publish it to S3. Use UserData in the Auto Scaling Launch configuration to pull down the Dockerfile from S3 and run it when new instances launch.
- D. Create a new Auto Scaling Launch Configuration with UserData scripts configured to pull the latest code at all times.

Correct Answer: B

... the bootstrapping process can be slower if you have a complex application or multiple applications to install. Managing a fleet of applications with several build tools and dependencies can be a challenging task during rollouts. Furthermore,

your deployment service should be designed to do faster rollouts to take advantage of Auto Scaling.

Reference: <https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf>

[DOP-C01 PDF Dumps](#)

[DOP-C01 VCE Dumps](#)

[DOP-C01 Study Guide](#)