Effective Way to Prepare for the Amazon DOP-C02 Certification Exam

P.S. Free 2025 Amazon DOP-C02 dumps are available on Google Drive shared by Exams4sures: https://drive.google.com/open?id=1JxijnddTR4ojwf4wwb4Y8YMww1lmmTUZ

Some customers may care about the private information problem while purchasing DOP-C02 Training Materials, if you are concern about this problem, our company will end the anxiety for you if you buy DOP-C02 training material of us . Our company is a professional company, we have lots of experiences in this field, and you email address and other information will be protected well, we respect the privacy of every customers. You give me trust , we give you privacy.

Amazon DOP-C02 (AWS Certified DevOps Engineer - Professional) Certification Exam is designed for professionals who are interested in validating their expertise in DevOps engineering practices and methodologies using AWS technologies. DOP-C02 Exam is intended for individuals who have a strong understanding of DevOps principles, practices, and tools and are experienced in implementing and managing continuous delivery systems and methodologies on AWS.

>> DOP-C02 Printable PDF <<

Amazon DOP-C02 Printable PDF: AWS Certified DevOps Engineer - Professional - Exams4sures Help you Prepare Exam Easily

We take responses from thousands of experts globally while updating the DOP-C02 content of preparation material. Their feedback and reviews of successful applicants enable us to make our Amazon DOP-C02 dumps material comprehensive for exam preparation purposes. This way we bring dependable and latest exam product which is enough to pass the Amazon DOP-C02 certification test on the very first take.

Amazon DOP-C02 exam consists of 75 multiple-choice and multiple-response questions, and the exam duration is 180 minutes. DOP-C02 exam fee is $300, and it can be taken at a testing center or online with a remote proctor. DOP-C02 exam covers various topics, such as designing and managing continuous delivery systems, monitoring and logging systems, implementing and managing security and compliance, and troubleshooting issues. Passing DOP-C02 Exam demonstrates that an individual has the knowledge and skills to design, deploy, and manage highly available, scalable, and fault-tolerant systems on AWS.

Amazon AWS Certified DevOps Engineer - Professional Sample Questions (Q88-Q93):

NEW QUESTION # 88
A company is using AWS CodeDeploy to automate software deployment. The deployment must meet these requirements:
* A number of instances must be available to serve traffic during the deployment Traffic must be balanced across those instances,

and the instances must automatically heal in the event of failure.
* A new fleet of instances must be launched for deploying a new revision automatically, with no manual provisioning.
* Traffic must be rerouted to the new environment to half of the new instances at a time. The deployment should succeed if traffic is rerouted to at least half of the instances; otherwise, it should fail.
* Before routing traffic to the new fleet of instances, the temporary files generated during the deployment process must be deleted.
* At the end of a successful deployment, the original instances in the deployment group must be deleted immediately to reduce costs.
How can a DevOps engineer meet these requirements?

- A. Use an Application Load Balancer and a blue/green deployment. Associate the Auto Scaling group and Application Load Balancer target group with the deployment group. Use the Automatically copy Auto Scaling group option, create a custom deployment configuration with minimum healthy hosts defined as 50%. and assign the configuration to the deployment group. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the BeforeBlockTraffic hook within appspec.yml to delete the temporary files.
- B. Use an Application Load Balancer and a blue/green deployment. Associate the Auto Scaling group and the Application Load Balancer target group with the deployment group. Use the Automatically copy Auto scaling group option, and use CodeDeployDefault.HalfAtAtime as the deployment configuration. Instruct AWSCodeDeploy to terminate the original instances in the deployment group, and use the BeforeAllowTraffic hook within appspec.yml to delete the temporary tiles.
- C. Use an Application Load Balancer and an in-place deployment. Associate the Auto Scaling group with the deployment group. Use the Automatically copy Auto Scaling group option. and use CodeDeployDefault.OneAtAtime as the deployment configuration. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the AllowTraffic hook within appspec.yml to delete the temporary files.
- D. Use an Application Load Balancer and an in-place deployment. Associate the Auto Scaling group and Application Load Balancer target group with the deployment group. Use the Automatically copy Auto Scaling group option, and use CodeDeployDefaulLAIIatOnce as a deployment configuration. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the BlockTraffic hook within appspec.yml to delete the temporary files.

Answer: B

Explanation:
Step 1: Use a Blue/Green Deployment Strategy
A blue/green deployment strategy is necessary to meet the requirement of launching a new fleet of instances for each deployment and ensuring availability. In a blue/green deployment, the new version (green environment) is deployed to a separate set of instances, while the old version (blue environment) remains active. After testing the new version, traffic can be gradually shifted.
Action: Use AWS CodeDeploy's blue/green deployment configuration.
Why: Blue/green deployment minimizes downtime and ensures that traffic is shifted only to healthy instances.
Reference:
Step 2: Use an Application Load Balancer and Auto Scaling Group
The Application Load Balancer (ALB) is essential to balance traffic across multiple instances, and Auto Scaling ensures the deployment scales automatically to meet demand.
Action: Associate the Auto Scaling group and Application Load Balancer target group with the deployment group.
Why: This configuration ensures that traffic is evenly distributed and that instances automatically scale based on traffic load.
Step 3: Use Custom Deployment Configuration
The company requires that traffic be rerouted to at least half of the instances to succeed. AWS CodeDeploy allows you to configure custom deployment settings with specific thresholds for healthy hosts.
Action: Create a custom deployment configuration where 50% of the instances must be healthy.
Why: This ensures that the deployment continues only if at least 50% of the new instances are healthy.
Step 4: Clean Temporary Files Using Hooks
Before routing traffic to the new environment, the temporary files generated during the deployment must be deleted. This can be achieved using the BeforeAllowTraffic hook in the appspec.yml file.
Action: Use the BeforeAllowTraffic lifecycle event hook to clean up temporary files before routing traffic to the new environment.
Why: This ensures that the environment is clean before the new instances start serving traffic.
Step 5: Terminate Original Instances After Deployment
After a successful deployment, AWS CodeDeploy can automatically terminate the original instances (blue environment) to save costs.
Action: Instruct AWS CodeDeploy to terminate the original instances after the new instances are healthy.
Why: This helps in cost reduction by removing unused instances after the deployment.
This corresponds to Option C: Use an Application Load Balancer and a blue/green deployment. Associate the Auto Scaling group and the Application Load Balancer target group with the deployment group. Use the Automatically copy Auto Scaling group option, and use CodeDeployDefault.HalfAtATime as the deployment configuration. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the BeforeAllowTraffic hook within appspec.yml to delete the temporary files.

NEW QUESTION # 89
A DevOps engineer needs to configure an AWS CodePipeline pipeline that publishes container images to an Amazon ECR repository. The pipeline must wait for the previous run to finish and must run when new Git tags are pushed to a Git repository connected to AWS CodeConnections. An existing deployment pipeline must run in response to new container image publications.
Which solution will meet these requirements?

- A. Configure a CodePipeline V1 type pipeline that uses QUEUED mode. Add a trigger filter to the pipeline definition that includes all branches. Add a stage at the end of the pipeline to invoke the existing deployment pipeline.
- B. Configure a CodePipeline V2 type pipeline that uses SUPERSEDED mode. Add a trigger filter to the pipeline definition that includes all branches. Configure an EventBridge rule that matches container image pushes to start the existing deployment pipeline.
- C. Configure a CodePipeline V1 type pipeline that uses SUPERSEDED mode. Add a trigger filter to the pipeline definition that includes all tags. Add a stage at the end of the pipeline to invoke the existing deployment pipeline.
- D. Configure a CodePipeline V2 type pipeline that uses QUEUED mode. Add a trigger filter to the pipeline definition that includes all tags. Configure an EventBridge rule that matches container image pushes to start the existing deployment pipeline.

Answer: D

Explanation:
* CodePipeline V2 with QUEUED modeensures that new executions wait for the previous execution to finish, preventing overlapping runs.
* Adding a trigger filter to listen forGit tagstriggers the pipeline only on new tag pushes.
* An EventBridge rule can detect new image pushes to ECR and start the deployment pipeline, integrating the build and deploy pipelines effectively.
* SUPERSEDED mode cancels the previous run when a new one starts, which is not desired here.
* Using branches instead of tags would trigger on all commits, not just releases.
References:
AWS CodePipeline V2 Triggers
AWS CodePipeline Execution Modes

NEW QUESTION # 90
A company builds an application that uses an Application Load Balancer in front of Amazon EC2 instances that are in an Auto Scaling group. The application is stateless. The Auto Scaling group uses a custom AMI that is fully prebuilt. The EC2 instances do not have a custom bootstrapping process.
The AMI that the Auto Scaling group uses was recently deleted. The Auto Scaling group's scaling activities show failures because the AMI ID does not exist.
Which combination of steps should a DevOps engineer take to meet these requirements? (Select THREE.)

- A. Update the Auto Scaling group to use the new launch template.
- B. Create a new AMI from a running EC2 instance in the Auto Scaling group.
- C. Increase the Auto Scaling group's desired capacity by I.
- D. Reduce the Auto Scaling group's desired capacity to O.
- E. Create a new launch template that uses the new AMI.
- F. Create a new AMI by copying the most recent public AMI of the operating system that the EC2 instances use.

Answer: A,E,F

Explanation:
To restore the functionality of the Auto Scaling group after the AMI was deleted, the DevOps engineer needs to create a new AMI and update the Auto Scaling group to use it. The DevOps engineer can create a new AMI by copying the most recent public AMI of the operating system that the EC2 instances use. This will ensure that the new AMI has the same operating system as the custom AMI that was deleted. The DevOps engineer can then create a new launch template that uses the new AMI and update the Auto Scaling group to use the new launch template. This will allow the Auto Scaling group to launch new instances with the new AMI.

NEW QUESTION # 91
A healthcare services company is concerned about the growing costs of software licensing for an application for monitoring patient wellness. The company wants to create an audit process to ensure that the application is running exclusively on Amazon EC2 Dedicated Hosts. A DevOps engineer must create a workflow to audit the application to ensure compliance.

What steps should the engineer take to meet this requirement with the LEAST administrative overhead?

- A. Use custom Java code running on an EC2 instance. Set up EC2 Auto Scaling for the instance depending on the number of instances to be checked. Send the list of noncompliant EC2 instance IDs to an Amazon SQS queue. Set up another worker instance to process instance IDs from the SQS queue and write them to Amazon DynamoDB. Use an AWS Lambda function to terminate noncompliant instance IDs obtained from the queue, and send them to an Amazon SNS email topic for distribution.
- B. Use AWS CloudTrail. Identify all EC2 instances to be audited by analyzing all calls to the EC2 RunCommand API action. Invoke a AWS Lambda function that analyzes the host placement of the instance. Store the EC2 instance ID of noncompliant resources in an Amazon RDS for MySQL DB instance. Generate a report by querying the RDS instance and exporting the query results to a CSV text file.
- C. Use AWS Config. Identify all EC2 instances to be audited by enabling Config Recording on all Amazon EC2 resources for the region. Create a custom AWS Config rule that triggers an AWS Lambda function by using the "config-rule-change-triggered" blueprint. Modify the Lambda evaluateCompliance () function to verify host placement to return a NON_COMPLIANT result if the instance is not running on an EC2 Dedicated Host. Use the AWS Config report to address noncompliant instances.
- D. Use AWS Systems Manager Configuration Compliance. Use calls to the put-compliance-items API action to scan and build a database of noncompliant EC2 instances based on their host placement configuration. Use an Amazon DynamoDB table to store these instance IDs for fast access. Generate a report through Systems Manager by calling the list-compliance-summaries API action.

Answer: C

Explanation:
Explanation
The correct answer is C. Using AWS Config to identify and audit all EC2 instances based on their host placement configuration is the most efficient and scalable solution to ensure compliance with the software licensing requirement. AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. By creating a custom AWS Config rule that triggers a Lambda function to verify host placement, the DevOps engineer can automate the process of checking whether the instances are running on EC2 Dedicated Hosts or not. The Lambda function can return a NON_COMPLIANT result if the instance is not running on an EC2 Dedicated Host, and the AWS Config report can provide a summary of the compliance status of the instances. This solution requires the least administrative overhead compared to the other options.
Option A is incorrect because using AWS Systems Manager Configuration Compliance to scan and build a database of noncompliant EC2 instances based on their host placement configuration is a more complex and costly solution than using AWS Config. AWS Systems Manager Configuration Compliance is a feature of AWS Systems Manager that enables you to scan your managed instances for patch compliance and configuration inconsistencies. To use this feature, the DevOps engineer would need to install the Systems Manager Agent on each EC2 instance, create a State Manager association to run the put-compliance-items API action periodically, and use a DynamoDB table to store the instance IDs of noncompliant resources. This solution would also require more API calls and storage costs than using AWS Config.
Option B is incorrect because using custom Java code running on an EC2 instance to check and terminate noncompliant EC2 instances is a more cumbersome and error-prone solution than using AWS Config. This solution would require the DevOps engineer to write and maintain the Java code, set up EC2 Auto Scaling for the instance, use an SQS queue and another worker instance to process the instance IDs, use a Lambda function and an SNS topic to terminate and notify the noncompliant instances, and handle any potential failures or exceptions in the workflow. This solution would also incur more compute, storage, and messaging costs than using AWS Config.
Option D is incorrect because using AWS CloudTrail to identify and audit EC2 instances by analyzing the EC2 RunCommand API action is a less reliable and accurate solution than using AWS Config. AWS CloudTrail is a service that enables you to monitor and log the API activity in your AWS account. The EC2 RunCommand API action is used to execute commands on one or more EC2 instances. However, this API action does not necessarily indicate the host placement of the instance, and it may not capture all the instances that are running on EC2 Dedicated Hosts or not. Therefore, option D would not provide a comprehensive and consistent audit of the EC2 instances.

NEW QUESTION # 92
A company has a data ingestion application that runs across multiple AWS accounts. The accounts are in an organization in AWS Organizations. The company needs to monitor the application and consolidate access to the application. Currently the company is running the application on Amazon EC2 instances from several Auto Scaling groups. The EC2 instances have no access to the internet because the data is sensitive Engineers have deployed the necessary VPC endpoints. The EC2 instances run a custom AMI that is built specifically tor the application.
To maintain and troubleshoot the application, system administrators need the ability to log in to the EC2 instances. This access must be automated and controlled centrally. The company's security team must receive a notification whenever the instances are accessed.

Which solution will meet these requirements?

- A. Use EC2 Image Builder to rebuild the custom AMI Include the most recent version of AWS Systems Manager Agent in the Image Configure the Auto Scaling group to attach the AmazonSSMManagedinstanceCore role to all the EC2 instances Use Systems Manager Session Manager to log in to the instances Enable logging of session details to Amazon S3 Create an S3 event notification for new file uploads to send a message to the security team through an Amazon Simple Notification Service (Amazon SNS) topic.
- B. Use AWS Systems Manager Automation to build Systems Manager Agent into the custom AMI Configure AWS Configure to attach an SCP to the root organization account to allow the EC2 instances to connect to Systems Manager Use Systems Manager Session Manager to log in to the instances Enable logging of session details to Amazon S3 Create an S3 event notification for new file uploads to send a message to the security team through an Amazon Simple Notification Service (Amazon SNS) topic.
- C. Deploy a NAT gateway and a bastion host that has internet access Create a security group that allows incoming traffic on all the EC2 instances from the bastion host Install AWS Systems Manager Agent on all the EC2 instances Use Auto Scaling group lifecycle hooks for monitoring and auditing access Use Systems Manager Session Manager to log in to the instances Send logs to a log group m Amazon CloudWatch Logs. Export data to Amazon S3 for auditing Send notifications to the security team by using S3 event notifications.
- D. Create an Amazon EventBridge rule to send notifications to the security team whenever a user logs in to an EC2 instance Use EC2 Instance Connect to log in to the instances. Deploy Auto Scaling groups by using AWS Cloud Formation Use the cfn-init helper script to deploy appropriate VPC routes for external access Rebuild the custom AMI so that the custom AMI includes AWS Systems Manager Agent.

Answer: A

Explanation:
Even if AmazonSSMManagedInstanceCore is a managed policy and not an IAM role I will go with C because this policy is to be attached to an IAM role for EC2 to access System Manager.

NEW QUESTION # 93
.....