

[2017-Feb-NEW]100% Real Exam Questions-Amazon AWS Certified DevOps Engineer - Professional VCE & PDF 190Q&As Download[Q1-Q10]

2017 Feb NEW Amazon AWS Certified DevOps Engineer - Professional Exam Dumps (PDF & VCE) Updated Today!1.|NEW Amazon AWS Certified DevOps Engineer - Professional PDF and VCE Dumps 190Q&As Download:

<http://www.braindump2go.com/aws-devops-engineer-professional.html>2.|NEW Amazon AWS Certified DevOps Engineer - Professional Exam Questions and Answers Download:<https://1drv.ms/f/s!AvI7wzKf6QBjgh4jQ6bIe9aoaNIC> QUESTION 1 Due to compliance regulations, management has asked you to provide a system that allows for cost-effective long-term storage of your application logs and provides a way for support staff to view the logs more quickly. Currently your log system archives logs automatically to Amazon S3 every hour, and support staff must wait for these logs to appear in Amazon S3, because they do not currently have access to the systems to view live logs. What method should you use to become compliant while also providing a faster way for support staff to have access to logs? A. Update Amazon S3 lifecycle policies to archive old logs to Amazon Glacier, and add a new policy to push all log entries to Amazon SQS for ingestion by the support team. B. Update Amazon S3 lifecycle policies to archive old logs to Amazon Glacier, and use or write a service to also stream your application logs to CloudWatch Logs. C. Update Amazon Glacier lifecycle policies to pull new logs from Amazon S3, and in the Amazon EC2 console, enable the CloudWatch Logs Agent on all of your application servers. D. Update Amazon S3 lifecycle policies to archive old logs to Amazon Glacier. key can be different from the tableEnable Amazon S3 partial uploads on your Amazon S3 bucket, and trigger an Amazon SNS notification when a partial upload occurs. E. Use or write a service to stream your application logs to CloudWatch Logs. Use an Amazon Elastic Map Reduce cluster to live stream your logs from CloudWatch Logs for ingestion by the support team, and create a Hadoop job to push the logs to S3 in five-minute chunks. Answer: E QUESTION 2 You want to securely distribute credentials for your Amazon RDS instance to your fleet of web server instances. The credentials are stored in a file that is controlled by a configuration management system. How do you securely deploy the credentials in an automated manner across the fleet of web server instances, which can number in the hundreds, while retaining the ability to roll back if needed? A. Store your credential files in an Amazon S3 bucket. Use Amazon S3 server-side encryption on the credential files. Have a scheduled job that pulls down the credential files into the instances every 10 minutes. B. Store the credential files in your version-controlled repository with the rest of your code. Have a post-commit action in version control that kicks off a job in your continuous integration system which securely copies the new credential files to all web server instances. C. Insert credential files into user data and use an instance lifecycle policy to periodically refresh the file from the user data. D. Keep credential files as a binary blob in an Amazon RDS MySQL DB instance, and have a script on each Amazon EC2 instance that pulls the files down from the RDS instance. E. Store the credential files in your version-controlled repository with the rest of your code. Use a parallel file copy program to send the credential files from your local machine to the Amazon EC2 instances. Answer: D QUESTION 3 You are using a configuration management system to manage your Amazon EC2 instances. On your Amazon EC2 Instances, you want to store credentials for connecting to an Amazon RDS DB instance. How should you securely store these credentials? A. Give the Amazon EC2 instances an IAM role that allows read access to a private Amazon S3 bucket. Store a file with database credentials in the Amazon S3 bucket. Have your configuration management system pull the file from the bucket when it is needed. B. Launch an Amazon EC2 instance and use the configuration management system to bootstrap the instance with the Amazon RDS DB credentials. Create an AMI from this instance. C. Store the Amazon RDS DB credentials in Amazon EC2 user data. Import the credentials into the Instance on boot. D. Assign an IAM role to your Amazon RDS instance, and use this IAM role to access the Amazon RDS DB from your Amazon EC2 instances. E. Store your credentials in your version control system, in plaintext. Check out a copy of your credentials from the version control system on boot. Use Amazon EBS encryption on the volume storing the Amazon RDS DB credentials. Answer: D QUESTION 4 Your company has developed a web application and is hosting it in an Amazon S3 bucket configured for static website hosting. The application is using the AWS SDK for JavaScript in the browser to access data stored in an Amazon DynamoDB table. How can you ensure that API keys for access to your data in DynamoDB are kept secure? A. Create an Amazon S3 role in IAM with access to the specific DynamoDB tables, and assign it to the bucket hosting your website. B. Configure S3 bucket tags with your AWS access keys for your bucket hosting your website so that the application can query them for access. C. Configure a web identity federation role within IAM to enable access to the correct DynamoDB resources and retrieve temporary credentials. D. Store AWS keys in global variables within your application and configure the application to use these credentials when making requests. Answer: C QUESTION 5 You need to implement A/B deployments for several multi-tier web applications. Each of them has its individual infrastructure: Amazon Elastic Compute Cloud (EC2) front-end servers, Amazon ElastiCache clusters, Amazon Simple Queue Service (SQS) queues, and Amazon Relational Database (RDS) Instances. Which combination of services would give you the

ability to control traffic between different deployed versions of your application? (Choose one.) A. Create one AWS Elastic Beanstalk application and all AWS resources (using configuration files inside the application source bundle) for each web application. New versions would be deployed using Elastic Beanstalk environments and using the Swap URLs feature.B. Using AWS CloudFormation templates, create one Elastic Beanstalk application and all AWS resources (in the same template) for each web application. New versions would be deployed using AWS CloudFormation templates to create new Elastic Beanstalk environments, and traffic would be balanced between them using weighted Round Robin (WRR) records in Amazon Route 53.C. Using AWS CloudFormation templates, create one Elastic Beanstalk application and all AWS resources (in the same template) for each web application. New versions would be deployed updating a parameter on the CloudFormation template and passing it to the cfn-hup helper daemon, and traffic would be balanced between them using Weighted Round Robin (WRR) records in Amazon Route 53.D. Create one Elastic Beanstalk application and all AWS resources (using configuration files inside the application source bundle) for each web application. New versions would be deployed updating the Elastic Beanstalk application version for the current Elastic Beanstalk environment. Answer: B

QUESTION 6 You work for an insurance company and are responsible for the day-to-day operations of your company's online quote system used to provide insurance quotes to members of the public. Your company wants to use the application logs generated by the system to better understand customer behavior. Industry, regulations also require that you retain all application logs for the system indefinitely in order to investigate fraudulent claims in the future. You have been tasked with designing a log management system with the following requirements:- All log entries must be retained by the system, even during unplanned instance failure. - The customer insight team requires immediate access to the logs from the past seven days. - The fraud investigation team requires access to all historic logs, but will wait up to 24 hours before these logs are available. How would you meet these requirements in a cost-effective manner? Choose 3 answers

A. Configure your application to write logs to the instance's ephemeral disk, because this storage is free and has good write performance. Create a script that moves the logs from the instance to Amazon S3 once an hour.

B. Write a script that is configured to be executed when the instance is stopped or terminated and that will upload any remaining logs on the instance to Amazon S3.

C. Create an Amazon S3 lifecycle configuration to move log files from Amazon S3 to Amazon Glacier after seven days.

D. Configure your application to write logs to the instance's default Amazon EBS boot volume, because this storage already exists. Create a script that moves the logs from the instance to Amazon S3 once an hour.

E. Configure your application to write logs to a separate Amazon EBS volume with the "delete on termination" field set to false. Create a script that moves the logs from the instance to Amazon S3 once an hour.

F. Create a housekeeping script that runs on a T2 micro instance managed by an Auto Scaling group for high availability. The script uses the AWS API to identify any unattached Amazon EBS volumes containing log files. Your housekeeping script will mount the Amazon EBS volume, upload all logs to Amazon S3, and then delete the volume.

Answer: C

QUESTION 7 You have an application running on Amazon EC2 in an Auto Scaling group. Instances are being bootstrapped dynamically, and the bootstrapping takes over 15 minutes to complete. You find that instances are reported by Auto Scaling as being In Service before bootstrapping has completed. You are receiving application alarms related to new instances before they have completed bootstrapping, which is causing confusion. You find the cause: your application monitoring tool is polling the Auto Scaling Service API for instances that are In Service, and creating alarms for new previously unknown instances. Which of the following will ensure that new instances are not added to your application monitoring tool before bootstrapping is completed?

A. Create an Auto Scaling group lifecycle hook to hold the instance in a pending: wait state until your bootstrapping is complete. Once bootstrapping is complete, notify Auto Scaling to complete the lifecycle hook and move the instance into a pending: complete state.

B. Use the default Amazon CloudWatch application metrics to monitor your application's health. Configure an Amazon SNS topic to send these CloudWatch alarms to the correct recipients.

C. Tag all instances on launch to identify that they are in a pending state. Change your application monitoring tool to look for this tag before adding new instances, and use the Amazon API to set the instance state to 'pending' until bootstrapping is complete.

D. Increase the desired number of instances in your Auto Scaling group configuration to reduce the time it takes to bootstrap future instances.

Answer: A

QUESTION 8 You have been given a business requirement to retain log files for your application for 10 years. You need to regularly retrieve the most recent logs for troubleshooting. Your logging system must be cost-effective, given the large volume of logs. What technique should you use to meet these requirements?

A. Store your log in Amazon CloudWatch Logs.

B. Store your logs in Amazon Glacier.

C. Store your logs in Amazon S3, and use lifecycle policies to archive to Amazon Glacier.

D. Store your logs in HDFS on an Amazon EMR cluster.

E. Store your logs on Amazon EBS, and use Amazon EBS snapshots to archive them.

Answer: C

QUESTION 9 You work for a startup that has developed a new photo-sharing application for mobile devices. Over recent months your application has increased in popularity; this has resulted in a decrease in the performance of the application due to the increased load. Your application has a two-tier architecture that is composed of an Auto Scaling PHP application tier and a MySQL RDS instance initially deployed with AWS CloudFormation. Your Auto Scaling group

has a min value of 4 and a max value of 8. The desired capacity is now at 8 because of the high CPU utilization of the instances. After some analysis, you are confident that the performance issues stem from a constraint in CPU capacity, although memory utilization remains low. You therefore decide to move from the general-purpose M3 instances to the compute-optimized C3 instances. How would you deploy this change while minimizing any interruption to your end users? A. Sign into the AWS Management Console, copy the old launch configuration, and create a new launch configuration that specifies the C3 instances. Update the Auto Scaling group with the new launch configuration. Auto Scaling will then update the instance type of all running instances. B. Sign into the AWS Management Console, and update the existing launch configuration with the new C3 instance type. Add an UpdatePolicy attribute to your Auto Scaling group that specifies AutoScalingRollingUpdate. C. Update the launch configuration specified in the AWS CloudFormation template with the new C3 instance type. Run a stack update with the new template. Auto Scaling will then update the instances with the new instance type. D. Update the launch configuration specified in the AWS CloudFormation template with the new C3 instance type. Also add an UpdatePolicy attribute to your Auto Scaling group that specifies AutoScalingRollingUpdate. Run a stack update with the new template. Answer: D

QUESTION 10

You've been tasked with implementing an automated data backup solution for your application servers that run on Amazon EC2 with Amazon EBS volumes. You want to use a distributed data store for your backups to avoid single points of failure and to increase the durability of the data. Daily backups should be retained for 30 days so that you can restore data within an hour. How can you implement this through a script that a scheduling daemon runs daily on the application servers?

A. Write the script to call the ec2-create-volume API, tag the Amazon EBS volume with the current date time group, and copy backup data to a second Amazon EBS volume. Use the ec2-describe-volumes API to enumerate existing backup volumes. Call the ec2-delete-volume API to prune backup volumes that are tagged with a date-time group older than 30 days.

B. Write the script to call the Amazon Glacier upload archive API, and tag the backup archive with the current date-time group. Use the list vaults API to enumerate existing backup archives. Call the delete vault API to prune backup archives that are tagged with a date-time group older than 30 days.

C. Write the script to call the ec2-create-snapshot API, and tag the Amazon EBS snapshot with the current date-time group. Use the ec2-describe-snapshot API to enumerate existing Amazon EBS snapshots. Call the ec2-delete-snapShot API to prune Amazon EBS snapshots that are tagged with a date-time group older than 30 days.

D. Write the script to call the ec2-create-volume API, tag the Amazon EBS volume with the current date-time group, and use the ec2-copy-snapshot API to back up data to the new Amazon EBS volume. Use the ec2-describe-snapshot API to enumerate existing backup volumes. Call the ec2-delete-snapshot API to prune backup Amazon EBS volumes that are tagged with a date-time group older than 30 days.

Answer: C !!!RECOMMEND!!!

1.|NEW Amazon AWS Certified DevOps Engineer - Professional PDF and VCE Dumps 190Q&As Download:
<http://www.braindump2go.com/aws-devops-engineer-professional.html>

2.|NEW Amazon AWS Certified DevOps Engineer - Professional Study Guide Video: YouTube Video: [YouTube.com/watch?v=nRhDUfpWQgE](https://www.youtube.com/watch?v=nRhDUfpWQgE)