PASSEXAM 問題集

更に上のクオリティ　更に上のサービス

EXAM

1年で無料進級することに提供する

http://www.passexam.jp

Exam : **AWS-DevOps-Engineer-Professional**

Title : AWS-DevOps-Engineer-Professional

Version : Demo

1.What method should I use to author automation if I want to wait for a CloudFormation stack to finish completing in a script?
A. Event subscription using SQS.
B. Event subscription using SNS.
C. Poll using <code>ListStacks</code> / <code>list-stacks</code>.
D. Poll using <code>GetStackStatus</code> / <code>get-stack-status</code>.
Answer: C
Explanation:
Event driven systems are good for IFTTT logic, but only polling will make a script wait to complete.
ListStacks / list-stacks is a real method, GetStackStatus / get-stack-status is not.
Reference: http://docs.aws.amazon.com/cli/latest/reference/cloudformation/list-stacks.html

2.Your application consists of 10% writes and 90% reads. You currently service all requests through a Route53 Alias Record directed towards an AWS ELB, which sits in front of an EC2 Auto Scaling Group. Your system is getting very expensive when there are large traffic spikes during certain news events, during which many more people request to read similar data all at the same time.
What is the simplest and cheapest way to reduce costs and scale with spikes like this?
A. Create an S3 bucket and asynchronously replicate common requests responses into S3 objects. When a request comes in for a precomputed response, redirect to AWS S3.
B. Create another ELB and Auto Scaling Group layer mounted on top of the other system, adding a tier to the system. Serve most read requests out of the top layer.
C. Create a CloudFront Distribution and direct Route53 to the Distribution. Use the ELB as an Origin and specify Cache Behaviours to proxy cache requests which can be served late.
D. Create a Memcached cluster in AWS ElastiCache. Create cache logic to serve requests which can be served late from the in-memory cache for increased performance.
Answer: C
Explanation:
CloudFront is ideal for scenarios in which entire requests can be served out of a cache and usage patterns involve heavy reads and spikiness in demand.
A cache behavior is the set of rules you configure for a given URL pattern based on file extensions, file names, or any portion of a URL path on your website (e.g., *.jpg). You can configure multiple cache behaviors for your web distribution. Amazon CloudFront will match incoming viewer requests with your list of URL patterns, and if there is a match, the service will honor the cache behavior you configure for that URL pattern. Each cache behavior can include the following Amazon CloudFront configuration values: origin server name, viewer connection protocol, minimum expiration period, query string parameters, cookies, and trusted signers for private content.
Reference: https://aws.amazon.com/cloudfront/dynamic-content/

3.You need to perform ad-hoc business analytics queries on well-structured data. Data comes in constantly at a high velocity. Your business intelligence team can understand SQL.
What AWS service(s) should you look to first?
A. Kinesis Firehose + RDS
B. Kinesis Firehose + RedShift
C. EMR using Hive

D. EMR running Apache Spark
Answer: B
Explanation:
Kinesis Firehose provides a managed service for aggregating streaming data and inserting it into RedShift. RedShift also supports ad-hoc queries over well-structured data using a SQL-compliant wire protocol, so the business team should be able to adopt this system easily.
Reference: https://aws.amazon.com/kinesis/firehose/details/

4.You are building a game high score table in DynamoDB. You will store each user's highest score for each game, with many games, all of which have relatively similar usage levels and numbers of players. You need to be able to look up the highest score for any game.
What's the best DynamoDB key structure?
A. HighestScore as the hash / only key.
B. GameID as the hash key, HighestScore as the range key.
C. GameID as the hash / only key.
D. GameID as the range / only key.
Answer: B
Explanation:
Since access and storage for games is uniform, and you need to have ordering within each game for the scores (to access the highest value), your hash (partition) key should be the GameID, and there should be a range key for HighestScore.
Reference:
http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GuidelinesForTables.html#GuidelinesForTables.Partitions

5.What is server immutability?
A. Not updating a server after creation.
B. The ability to change server counts.
C. Updating a server after creation.
D. The inability to change server counts.
Answer: A
Explanation:
... disposable upgrades offer a simpler way to know if your application has unknown dependencies. The underlying EC2 instance usage is considered temporary or ephemeral in nature for the period of deployment until the current release is active. During the new release, a new set of EC2 instances are rolled out by terminating older instances. This type of upgrade technique is more common in an immutable infrastructure.
Reference: https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf