

- 241.** C. AWS is responsible for the security of virtualization infrastructure. All other items in this list are your responsibility. As a hint on questions like this and related to the AWS shared responsibility model, AWS is typically responsible for anything with the word *infrastructure*, although there are some exclusions (for example, *application infrastructure*).
- 242.** A. An IAM role is assumed by an EC2 instance when it needs to access other AWS services, and that role has permissions associated with it. While these permissions are formally defined in a policy (B), it is the role that is used by the instance for actual service access.
- 243.** A, D. Just as is the case with a compute instance (EC2), a task in a container needs an IAM role with permissions to access S3 (A), which in turn requires a policy specifying a permission that lets ECS tasks access S3 (D). Both of these are required to ensure access. Security groups apply to network traffic and would not affect S3 access, and while a VPC endpoint could be used (C), it is not required.
- 244.** C. By default, newly created S3 buckets are private. They can only be accessed by a user that has been granted explicit access.

## Domain 4: Design Cost-Optimized Architectures

1. A, B. When instance cost is the issue, the answers are almost always to consider some form of lowered instance pricing. AWS provides reserved instances and spot instances and the spot market for this purpose. Further, paying for reserved instances all up front is the most cost-effective means of getting reserved instances. Therefore, A and B are correct. C is problematic, as running a smaller instance for longer is not necessarily any cheaper than running a large instance for shorter amounts of time. Option D has some validity, but AWS is almost certainly going to point you back to either reserved instances or the spot market (A and B).
2. C, D. Reserved instances can be paid for in no up-front, partial up-front, and all up-front models, where all up-front is the least expensive and no up-front is the most expensive.
3. D. Reserved instances are locked to the region in which they are created, so D is correct. You would need to create a new reserved instance in the new region.
4. C. This should be an easy correct answer: Spot instances via the spot market are the potentially least expensive option, given that your compute has flexible timing and needs.

5. B, C. Applications with spiky workloads are reasons to use on-demand, as on-demand can scale up and down quickly. Flexible start and end times is a criterion for choosing spot instances, and steady-state usage is ideal for reserved instances. Anytime you're testing a new application, on-demand is a good choice
6. B, D. Applications with spiky workloads are reasons to use on-demand, as on-demand can scale up and down quickly. Flexible start and end times is a criterion for choosing spot instances, and steady-state usage is ideal for reserved instances. Spot instances also make heavy compute affordable when it would not be on other instance types.
7. C, D. The first option is easy, as it actually has *reserved* in the wording. Steady-state usage is also a use case for reserved instances, to gain cost savings. Large and immediate additional capacity needs are best facilitated by spot instances, and on-demand instances are best for users with no initial payment ability.
8. A, B. S3 shares the durability of all S3 storage classes at 11 9s. It also provides the highest availability throughput of all S3 storage classes. Infrequent access is a use case for S3-IA, while the ability to re-create objects would suggest S3 One Zone-IA.
9. A, D. The problem here is that instances are scaling down too quickly. This results in them then being restarted, which incurs cost. The solutions here should be ones that cause instances to stay around a bit longer, which meets demand. Both A and D do this. Cool-down timers increase the time for the group to ensure that previous scaling actions were completed (A), and the CloudWatch alarm period for scaling down should also be increased (D).
10. D. This is a pretty "by the book" question, and in this case, is the exact use case for which S3-IA (Infrequent Access) was built. Instant access with less frequent requests is ideal for S3-IA.
11. A. S3-IA is less expensive than S3, regardless of use case. It is certainly possible that S3-IA is not *appropriate* for a certain use case, but it is less expensive on a "per byte retrieved" case.
12. C. CloudFront will allow you to cache files that are frequently accessed. In this case, that should actually reduce costs. While CloudFront does incur a new additional cost, it would likely be offset by reduced egress from the EFS as well as the compute of additional EC2 instances to handle requests.
13. D. All the description here suggests using Glacier. The documents are a large archive, and many will never be accessed. However, the requirement for quick retrieval points to a need for expedited retrieval. Glacier with expedited retrieval is still going to cost less than S3-IA for access that isn't that frequent.
14. B. First, EBS is a much better choice than EFS for a single-instance application. While a database would certainly benefit from IOPS, there's no need; peaks are small, and usage overall is low. A General Purpose SSD is sufficient here.

15. C. If you have a larger database workload, provisioned IOPS SSD is ideal.
16. D. A cold HDD is the least expensive EBS volume type.
17. A. This is a tough question. You can eliminate B and C because both involve additional services: CloudWatch, Lambda, and additional EC2 instances. Taking EBS snapshots is good, and by moving those snapshots into S3 (which is the default), you get durability automatically. Mirroring data is also a great option—providing fault tolerance—but this does not provide a durability component, something the question specifically requires. Therefore, A is the best answer.
18. B. There are two components to this question: which storage medium is appropriate, and how should older records be deleted. To get both immediate retrieval and lifecycle management, you'd need S3, as in option B. (Also, EBS does not offer lifecycle management, in option D.)
19. A, C. RDS read replicas would take some of the read load off of the database instance, as would ElastiCache. The first allows reads to go to other instances, and the second caches already accessed data.
20. B, D. Glacier is the easy choice, as it can handle the oldest data and still meet the 10-hour retrieval time. S3 RRS is deprecated and shouldn't be considered. This leaves S3 and S3-IA. S3-IA is always less expensive than S3, so it's the better option here.
21. B. Placement groups are typically in a single availability zone, but now spread placement groups can be placed across availability zones.
22. B. It is typical to think of a spread placement group as a group spread across availability zones, but that is a misnomer. The *spread* in spread placement group means that the instances are spread across distinct underlying hardware, and although they *can* be spread across availability zones, they don't have to be.
23. C. A spread placement group can have a maximum of seven running instances per AZ.
24. A, C. Spread placement groups primarily offer reduced network lag between instances (C). They also allow for cross-VPC spanning of instances (A).
25. C. The only false statement here is C: Spread placement groups cannot be set up across regions, and therefore this entire statement is untrue.
26. B, C. Egress always has a cost associated with it (B), while ingress is always free. Transferring data across regions is treated the same as transfers to the Internet. Only inter-AZ data transfer is guaranteed to be costless (D), making C the other correct answer.
27. C. The least cost is always going to be “free,” so look for anything that might be ingress. In this cost, uploading to S3 is straight ingress and is therefore free and the cheapest option.
28. B. There are no ingress options here, so nothing is guaranteed to be free. In that case, you should then look for something that moves data within the same availability zone. That's always the least expensive (and usually free, depending on IP addresses), and in this case, that's option B: inter-AZ data transfer between instances.

- 29. A, B. First, CloudFront is always a good option. It's free to move data from EC2 to CloudFront, so that could reduce how far data must travel, and associated costs. Then, private IPs allow for communication that doesn't route out to the Internet, and generally AWS charges less for communication from private IP to private IP.
- 30. B. Although there is a free tier, it's a billing option and not an actual support level. That makes B the non-level in this answer set.
- 31. C. AWS reduces the need for large capital expenditures and provides a pay-as-you-go model instead.
- 32. B. AWS uses a pay-as-you-go model for all of its services.
- 33. D. D is incorrect; you actually pay even less as AWS grows, due to economies of scale.
- 34. C. "Migration only" is not a pricing model for instances. The only model not mentioned here is dedicated hosts.
- 35. B. AWS suggests using reserved instance pricing in TCO calculations, as it is closest to on-premises servers in an apples-to-apples comparison.
- 36. A. Standard reserved instances often provide up to a 75% discount as compared to on-demand instances.
- 37. C. There is no "half upfront" payment option. The valid options are no upfront, partial upfront, and all upfront.
- 38. C. Paying all upfront is the cheapest option of these three and provides the greatest savings over on-demand pricing.
- 39. B, D. Reserved instances can be purchased for either one- or three-year terms.
- 40. A, C. A spot instance (A) is a valid model, but spot market (B) is not; spot market is where you purchase spot instances. Dedicated hosts (C) is another valid model. All upfront is a payment option, but not an actual pricing model for instances.
- 41. A. Spot instances are recommended for applications with flexible start and end times, that need to run with low compute prices, or that may have urgent compute needs for large amounts of additional capacity.
- 42. B. On-demand instances are best when usage patterns swing severely and can't be predicted.
- 43. B. On-demand instances are ideal for any usage that swings wildly in unpredictable patterns, particularly if a job cannot be halted. If usage is predictable, a long-running job might benefit from a reserved instance, and if the job can be stopped, then spot instances would be better.
- 44. A, D. This is a little tricky, as dedicated hosts function a bit differently than the other instance types, in both purchasing and payment. In this case, it's important to note that the question is about purchasing, and *not* payment (which would drive you to answer "all upfront," "partial upfront," or "no upfront"). Dedicated hosts can be purchased as an on-demand instance or as a reservation for significant savings.

- 45. B, C. Reserved instances are the best option for steady-state applications and require at least a one-year commitment, which would point to options B and C.
- 46. D. Spot instances are not ideal for spikes in usage, as those instances may be terminated at any time.
- 47. B. In this scenario, you want to ensure that instances stay up (eliminating the spot market) and that there is no long-term commitment (eliminating reserved instances). Dedicated hosts don't make sense, so this leaves on-demand instances.
- 48. A, B. The spot market provides instances that can stop and start at any time. Now, applications on these instances can be stopped and restarted (A). Additionally, costs are significantly lower than on-demand pricing (B). However, the hardware can change often, and spikes in usage are *not* well suited for spot instances.
- 49. D. On a pure "storage per GB" comparison, Amazon Glacier is the least expensive storage class.
- 50. A. S3-SSE is an encryption solution. Standard IA is infrequent access, RRS is reduced redundancy (and is now deprecated), and of course Glacier is a valid S3 storage class.
- 51. C. Uploading data is the textbook definition of ingress, and ingress never has associated fees.
- 52. A. It is always free to move data into CloudFront. There may be a cost associated with egress from CloudFront, but the transfer to CloudFront is cost-free.
- 53. B. The AWS free tier is *just* AWS but without a cost. You can use up to several limits of services (data transfer, compute, storage, etc.) at no cost within the AWS free tier.
- 54. D. The AWS marketplace offers free and paid software products, many of which run on the AWS free tier. You can find AMIs and services as well as many trial offerings from third parties.
- 55. D. AWS free tier offers almost everything that paid AWS does, simply at lesser volumes.
- 56. D. The four AWS support plans are basic, developer, business, and enterprise.
- 57. C, D. This should be pretty intuitive: The higher and more business-oriented levels of support offer 24/7 support; in this case, business and enterprise.
- 58. B. AWS Trusted Advisor is an online resource that helps you reduce cost as well as increase performance and improve security. However, it does not provide logging (C) or affect Auto Scaling limits or configuration.
- 59. C. There are five core Trusted Advisor checks: cost optimization, security, fault tolerance, performance, and service limits. Note: In some places, AWS will say that there are seven checks, but in others, five. The most current documentation indicates the five checks noted here.

- 60. A, B. AWS Trusted Advisor provides advice that typically is useful in all environments, for all use cases. In this set of answers, the two that meet that criteria are turning on MFA for the root account and avoiding global Internet access to an S3 bucket. These recommendations will apply to almost all situations. The other two options—C and D—are use-case specific and therefore would not be suggested by Trusted Advisor.
- 61. D. AWS Trusted Advisor makes recommendations about S3 bucket usage, IAM usage, and snapshots (both EBS and RDS) but does not make recommendations regarding DNS, so D is correct.
- 62. B. AWS Trusted Advisor makes recommendations in five categories: cost optimization, performance, security, fault tolerance, and service limits.
- 63. A, D. AWS Trusted Advisor makes recommendations in five categories: cost optimization, performance, security, fault tolerance, and service limits.
- 64. A, C. First, C is an easy choice: MFA on the root account is one of the most common recommendations. Then, consider the areas in which Trusted Advisor can make absolute recommendations; underuse of DNS records doesn't make a lot of sense (how do you "underuse DNS?") and coming up with the "correct" S3 storage class involves understanding use cases, which Trusted Advisor can't do. This leaves A, idle load balancers.
- 65. A, B. AWS makes five standard recommendations: Right-size your services to meet capacity needs at the lowest cost, save money by reserving, use the spot market, monitor and track service usage, and use Cost Explorer to optimize savings.
- 66. C. AWS makes five standard recommendations: Right-size your services to meet capacity needs at the lowest cost, save money by reserving, use the spot market, monitor and track service usage, and use Cost Explorer to optimize savings. Using the spot market (C) falls into that last category. The other answers are all use-case driven and really don't fit into general cost-saving recommendations.
- 67. C. This should be pretty basic: AWS Cost Explorer provides reports via analysis for evaluating your overall AWS costs over time.
- 68. B. This is largely a matter of recognizing the valid AWS tools—AWS Trusted Advisor and AWS Cost Explorer—and then determining which deals with costs. In this case, that's AWS Cost Explorer.
- 69. B. Cost Explorer gives you reports on EC2 monthly cost and usage that can help analyze monthly spending on instances.
- 70. C. While AWS Cost Explorer can give you information about your monthly storage costs, AWS Budgets allows you to set alerts and then add custom programming to reduce or halt those costs.
- 71. D. This is an important question. None of the tools listed allow for actual "cutoffs" at cost thresholds. AWS Budgets allows you notifications when a threshold is met but does *not* allow you to cut off spending at a certain point on its own.

72. C. Elastic Transcoder allows you to produce media files that are optimized and well suited for various device types.
73. A. This question is as much about recognizing the various AWS service acronyms as anything. Here, RDS—the Relational Database Service—allows you to use Oracle on an AWS managed service.
74. A. Elastic Beanstalk can deploy your code and handle capacity provisioning, load balancing, and setting up Auto Scaling and health checks, all with very little oversight. Note that you'd still need personnel to keep an application like this running, but Elastic Beanstalk can reduce initial resources needed for application deployment.
75. B. CloudFormation allows you to automate provisioning and, in this case, to create standardized JSON scripts that can be lightly modified to stand up entire stacks for multiple applications that share a common structure.
76. D. Snowball is almost always the most cost-effective approach to data transfer when you approach 50 TB, and there are good reasons to consider it even at 10 TB or more.
77. A. Storage Gateway is a hybrid storage service and allows your on-premises data store to interact with S3.
78. C. Large data should always make you think, “Snowball.” Snowball gives you a reliable, scalable, petabyte-scale data transfer solution.
79. D. Redshift is AWS's managed service for OLAP and business intelligence.
80. A. EMR, Elastic MapReduce, is a web service targeted at processing large amounts of data. It is optimized for this task and often provides cost savings over EC2 instances running similar processes.
81. A, D. QuickSight is a business analytics service, and Redshift is ideal for business intelligence and OLAP. While you could build high-performance applications using EC2 instances and provisioned IOPS EBS volumes, managed services like QuickSight and Redshift are almost always going to be more cost effective.
82. D. Both A and B are going to incur significant costs and custom code. C is not a bad option on the analytics side but will still likely require custom code to aggregate the data sources. QuickSight, however, is designed exactly for this task: combining data sources and then performing analytics and extracting insights.
83. C. Glacier is Amazon's long-term data archival solution.
84. A. CloudFormation is ideal for automating deployment without manual intervention, but it's actually Elastic Beanstalk that handles the *provisioning* of resources.
85. C. Kinesis is intended to handle streaming data sources. It collects and processes data from these streaming sources in real time and would be ideal to replace custom code that handles this same process, as the question asks.

- 86. A. Lambda allows you to “ignore” the underlying resources required for running code. You simply give Lambda the code to run, and Lambda will handle provisioning resources in a scalable and cost-effective manner.
- 87. D. CloudWatch provides monitoring of applications and is a low-cost solution for AWS monitoring.
- 88. A. CloudTrail is the AWS service for logging and is particularly helpful for auditing and compliance.
- 89. C. Almost all of these add unnecessary steps and involve multiple instances or either Oracle or PostgreSQL. The easiest, most cost-effective option is to migrate directly from Oracle to PostgreSQL using DMS, the Database Migration Service.
- 90. A. S3 is the AWS choice for durability and flat-file (non-relational data) storage.
- 91. A. IAM is the best option for handling users, groups, and permissions within AWS.
- 92. A, B. IAM is the best option for handling users, groups, and permissions within AWS. You can then add Cognito to offer single sign-on capabilities to your applications.
- 93. B. Trusted Advisor is a great start to find glaring holes or deficiencies in an AWS environment.
- 94. C. OpsWorks is a configuration management tool that actually can use Chef, so many of the existing modules would plug right in and existing expertise would translate directly over.

## Domain 5: Define Operationally Excellent Architectures

- 1. B. AWS does guarantee that all SQS messages will be delivered *at least* once, but the message *may be* delivered more than once (making option A incorrect). This is not related to the number of requests to the queue or the applications using the queue; therefore, both C and D are incorrect. This leaves B, the correct answer.
- 2. B, C. This is a common question AWS often asks to ensure that you understand that managed services like RDS and DynamoDB are indeed completely *managed*: You cannot access the underlying operating system of the service. This leaves EC2 and EMR as the remaining, and correct, answers. While EMR does provide you with a lot of functionality “out of the box,” it still allows root level access, as do EC2 instances.