# Domain 1: Design Resilient Architectures

1. B. This is a common question on AWS exams, and relates to your understanding of the various S3 classes. S3 and S3-IA have the same durability, but the availability of S3 is one 9 greater than S3-IA. S3 has 99.99 availability, while S3-IA has 99.9 availability. Glacier has much greater first-byte latency than S3, so both C and D are false.

2. B. Anytime the primary consideration is storage with a local data presence—where data must be stored or seen to be stored locally—a storage gateway gives you the best option. This reduces the choices to B and D. B will store the files in S3 and provide local cached copies, while D will store the files locally and push them to S3 as a backup. Since management is concerned about storage in the cloud of primary files, B is the best choice; local files are the primary source of data, while still allowing the company to experiment with cloud storage without "risking" its data being stored primarily in the cloud.

3. B. Many of these answers are nonsensical in terms of what AWS allows. The limits on size related to S3 are for objects; an individual object can be as large as 5 TB. Both A and C, then, are not useful (or possible). D proposes to increase the maximum object size to 50 GB, but the maximum object size is already 5 TB. Option B is correct; AWS recommends using Multipart Upload for all objects larger than 100 MB.

4. C, D. PUTs of new objects have a read after write consistency. DELETEs and overwrite PUTs have eventual consistency across S3.

5. C. First, note that "on standard class S3" is a red herring, and irrelevant to the question. Second, objects on S3 can be 0 bytes. This is equivalent to using `touch` on a file and then uploading that 0-byte file to S3.

6. A. This is a matter of carefully looking at each URL. Bucket names—when not used as a website—always come after the fully qualified domain name (FQDN); in other words, after the forward slash. That eliminates C. Additionally, the region always comes earlier in the FQDN than amazonaws.com, eliminating D. This leaves A and B. Of the two, A correctly has the complete region, us-east-2.

7. C. This is another question that is tricky unless you work through each part of the URL, piece by piece. The first clue is that this is a website hosted on S3, as opposed to directly accessing an S3 bucket. Where website hosting is concerned, the bucket name is *part of* the FQDN; where direct bucket access is concerned, the bucket name comes *after* the FQDN. This is an essential distinction. This means that A and B are invalid. Then, you need to recall that the s3-website portion of the FQDN is always connected to the region; in other words, it is not a subdomain. The only option where this is the case is C.

8. A. This is another case of rote memorization. S3 and S3-IA have the same durability; however, the availability of S3 is higher (99.99 vs. the 99.9 of S3-IA). Both Glacier and S3-IA have the same durability of standard S3, so both C and D are false.

9. B. This is an important distinction when understanding S3 classes. Standard S3, S3-IA, and S3 One Zone-IA all are equally durable, although in One Zone-IA, data will be lost if

the availability zone is destroyed. Each class has different availability, though: S3 is 99.99, S3-IA is 99.9, and S3 One Zone-IA is 99.5. Therefore, it is false that all have the same availability (B).

**10.** A, C. The wording of this question is critical. S3 buckets are created within a region, but the AWS console and your account will show you *all* S3 buckets at all times. While a bucket is created in a specific region, names of buckets are also global. IAM permissions are also global and affect all regions. RDS and EC2 instances are region specific, and only appear in the regions in which they were created in the AWS console.

**11.** A, D. EBS volumes are block-based storage, meaning that A is correct and B is incorrect. That leaves C and D. The default EBS volume is SSD, so C is false. However, EBS volumes can be in a variety of types, including magnetic and SSD options, so D is true.

**12.** D. AMIs are not cross-region, regardless of account or security group. This makes B and C invalid. A is a valid choice but will not preserve any of the permissions or roles that allow the instance to connect to S3. Therefore, D is the correct option: manual configuration of the AMI *after* it has been copied is required for correct operation.

**13.** D. This is a bit of a trick question if you're not careful. While S3 allows for 0-byte objects, and charges as such, S3-IA charges all objects as if they are *at least* 128 KB in size. So while you can store a smaller object in S3-IA, it will be considered 128 KB for pricing and charging purposes.

**14.** A, D. A Multi-AZ setup is the easiest solution, and the most common. Turning on read replicas (option B) is not a guarantee, as read replicas are not automatically installed in different AZs or regions. However, with option D, a cross-region replica configuration will ensure multiple regions are used. A storage gateway (option C) is backed by S3, not RDS.

**15.** A, D. Launch configurations are concerned primarily with creating new instances while staying abstract from the details of what is on those instances. So the AMI and IAM role for an instance is a general configuration, applies to all created instances, and is correct (A and D). The polling time for latency isn't connected to launching new instances (although it might be a trigger configured elsewhere). Each instance is associated with a different EBS volume, so selecting an EBS volume for multiple instances doesn't actually make sense.

**16.** D. Launch configurations are where details are specified for creating (launching) new instances (option D). Security groups have to do more with what traffic is allowed into and out of the launched instances. The remaining two options—A and C—don't make sense in this context.

**17.** D. By default, EBS root volumes are terminated when the associated instance is terminated. However, this is only the default value; therefore A is not correct. Option B is not directly addressing the question; the EBS volume would still be deleted even if you take a snapshot. Option C is not relevant, but option D is: You can use the AWS CLI (or the console) to set the root volume to persist after instance termination.

**18.** B. EBS volumes are backed up to S3 incrementally.

**19.** B. EBS volumes can only attach to a single instance at one time. The other options are all simply to distract.

**20.** A, B. All instances and most services in AWS provide tagging for metadata. Certificates are related to SSL and help define the identity of a site or transmission, policies are related to permissions and roles, and labels are not (currently) an AWS construct.

**21.** A, B. Valid concerns in this list include placing storage close to your users, to reduce network latency, and distance from your operations center. This latter is a little less obvious but is centered around disaster recovery scenarios: If a disaster destroyed your operations center, you would not want your storage on AWS to be geographically in the same area.

**22.** B. Every EC2 instance provides the option to specify an availability zone. While you don't have to specify something other than the default, instances are always provisioned in a specific availability zone, which is user configurable.

**23.** C. Spread placement groups—which are relatively new to AWS—can be placed across multiple availability zones. Cluster placement groups cannot, and *placement groups* generally refers to cluster placement groups. *Cross-region placement groups* is a made-up term.

**24.** C. A customer gateway is the anchor on the customer side of an Amazon VPN connection. A storage gateway is for caching or storing data and connecting to S3. A virtual private gateway is an important part of a VPN connection but exists on the AWS side of the connection. A virtual private network is actually what VPN stands for.

**25.** B. VPN connections between an on-premises site and AWS consist of a customer gateway on the customer side and a virtual private gateway on the AWS side.

**26.** B. A typical VPN connection uses two different tunnels for redundancy. Both tunnels move between the customer gateway and the virtual private gateway.

**27.** D. Traffic begins at the on-premises site, which means starting at a customer gateway. Traffic then flows through the Internet and to the virtual private gateway at AWS. Then, from the gateway, traffic can flow into an Amazon VPC.

**28.** A, C. Traffic across the Internet can only flow between public IP addresses in most cases. For a VPN connection, you will need a customer gateway with a public IP address as well as a virtual private gateway with a public IP address, both of which you may be responsible for configuring. A VPC does not have an IP address of its own (making option B incorrect), and VPN tunnels do not either (option D).

**29.** A. A storage gateway is the correct answer, as it is used for caching or storing data and connecting to S3. A customer gateway is the anchor on the customer side of an Amazon VPN connection. A virtual private gateway is used for connecting into AWS via a VPN, and a virtual private network is actually what VPN stands for.

**30.** A, B. Both file and volume gateways offer solutions for connecting to cloud-based storage. A cached gateway isn't an AWS option, and a virtual private gateway is used in creating VPN connections.

**31.** A. Each of the options is a valid configuration for a storage gateway. Of the options, file gateway provides an NFS-style protocol for transferring data to and from the gateway and therefore is the best option.

**32.** D. This is relatively easy because the word *tape* actually appears in both the question and the answer. A tape gateway backs up data in Amazon Glacier while providing a virtual tape infrastructure that many existing tape backup systems can utilize.

**33.** C. A stored volume gateway stores data at the on-premises data store and backs up to S3 asynchronously to support disaster recovery. Most important, though, is that by storing data locally, network latency is minimal. Of the available options, only a stored volume gateway provides local data with this speed of access across an entire dataset.

**34.** C. Anytime very large data needs to be moved into AWS, consider Snowball. Snowball is a physical device that allows for data to be physically sent to AWS rather than transferred over a network. It is the only solution that will not potentially cause disruptive network outages or slowdowns.

**35.** A, C. A cached volume gateway stores the most commonly accessed data locally (option D) while keeping the entire dataset in S3. This has the effect of reducing the cost of storage on-site, because you need less (option B). Since both of these are true, you need to select the other two options as reasons to *not* use a cached volumes gateway: A and C.

**36.** A. Be careful here. While it might seem at a glance that a tape gateway is best, most backup solutions do not employ tape backups. They use NFS mounts and file-based backups, which is exactly what a file gateway is best used for.

**37.** B. A cached volume gateway is ideal when a *portion* of a dataset is at issue. The most used data will be cached, and therefore stored in the local cache on premises. If the entire dataset is needed, then a stored volume gateway is a better choice.

**38.** C. If the entire dataset is needed, then a stored volume gateway is a better choice than a cached volume gateway. The stored volume stores the entire dataset on premises and therefore is very fast for all data access.

**39.** D. A tape gateway is ideal for replacing off-site tape directories. The gateway is a virtual tape directory and avoids the costs of transporting actual tapes to an expensive off-site location.

**40.** D. This should be automatic: Glacier is the Amazon offering for long-term "on ice" storage.

**41.** B, D. Launch configurations are specific to a region, as are EC2 instances. While S3 buckets are created in a region, their names are global. IAM users also exist across all of your account.

**42.** A. HTTP 200 is the general return for success, and this is the case for S3 uploads as well.

**43.** D. This is easy to miss. All S3 storage classes (S3 standard, S3-IA, and S3 One Zone-IA) share the same durability of 11 9s.

**44.** D. This is easy to miss. All S3 storage classes (S3 standard, S3-IA, and S3 One Zone-IA) share the same durability of 11 9s.

**45.** D. All S3 storage classes (S3 standard, S3-IA, and S3 One Zone-IA) share the same durability of 11 9s.

**46.** A. While all S3 storage classes share the same durability, they have varying availability. S3-IA has 99.9%, while S3 One Zone-IA is less (99.5%), and S3 standard is higher (99.99%).

**47.** B. While all S3 storage classes share the same durability, they have varying availability. S3-IA has 99.9%, while S3 One Zone-IA is less (99.5%), and S3 standard is higher (99.99%).

**48.** C. While all S3 storage classes share the same durability, S3 standard has the highest availability, at 99.99%.

**49.** D. All of the S3 storage classes support both SSL for data in transit and encryption for data at rest.

**50.** D. All of the S3 storage classes support both SSL for data in transit and encryption for data at rest.

**51.** D. All S3 storage classes have buckets that can be created in a specific region. The objects in the buckets are then stored in availability zones within that region, depending upon the storage class.

**52.** D. While S3 does use availability zones to store objects in buckets, you do not choose the availability zone yourself. Even S3 One Zone-IA does not allow you to specify the AZ for use.

**53.** A. S3 storage is key based. Keys can be a string and the value is the uploaded object.

**54.** C, D. S3 does not provide SSH or SFTP access, nor standard FTP access. You can access your data through the AWS console and through a REST interface via HTTP.

**55.** B, C. S3 is built to automatically scale in times of heavy application usage. There is no requirement to enable Auto Scaling (A); rather, this happens automatically (so B is correct). Further, S3 tends to scale evenly across the AWS network (C). Option D is the opposite of what AWS intends.

**56.** B. When evaluating S3 storage, all storage classes have the same durability. For cost, though, S3 One Zone-IA is the clear winner. Only Glacier is potentially less expensive but does not provide the same quick file access that S3 One Zone-IA does.

**57.** D. This is nearly a trick question. S3 in general is built for scalability, and the different storage classes are not substantially different in terms of how they can scale. However, without knowing how quickly data retrieval must be, and the priorities of the data, it is impossible to choose between S3 standard and S3-IA, and in some cases, even Glacier.

**58.** C. By default, all AWS accounts can create up to 100 buckets. However, this limit can easily be raised by AWS if you request an upgrade.

**59.** B. S3 uploads are, by default, done via a single operation, usually via a single PUT operation. AWS suggests that you can upload objects up to 100 MB before changing to Multipart Upload.

**60.** B. Using the Multipart Upload is almost entirely a function of the size of the files being uploaded. AWS recommends using it for any files greater than 100 MB, and 10 GB is certainly large enough to benefit from Multipart Uploads.

**61.** A, C. Multipart Upload is, as should be the easiest answer, ideal for large objects on stable networks (A). But it also helps handle less-reliable networks as smaller parts can fail while others get through, reducing the overall failure rate (C). There is no cost associated with data ingress (B), and D doesn't make much sense at all!

**62.** A, C. Presigned URLs are created to allow users without AWS credentials to access specific resources (option C). And it's the creator of the URL (option A) that assigns these permissions, rather than the user (option B). Finally, these credentials are associated with the URL but are not encrypted into the URL itself.

**63.** D. Presigned URLs are not tied to specific AWS services. They are simply URLs that can point at anything a normal URL can point at, except that the creator can associate permissions and a timeout with the URL.

**64.** D. A presigned URL is always configured at creation for a valid Time to Live (often referred to as TTL). This time can be very short, or quite long.

**65.** B, D. Overwrite PUTs and DELETEs have eventual consistency. PUTs of new objects have write and then read consistency.

**66.** D. These are all consistent with S3 behavior. Option A could occur as the new object is being propagated to additional S3 buckets. B and C could occur as a result of eventual consistency, where a DELETE operation does not immediately appear.

**67.** C. All regions have eventual consistency for overwrite PUTs and DELETEs.

**68.** A, D. All S3 storage classes are object-based, while EBS and EFS are block-based.

**69.** B. EBS stands for Elastic Block Storage.

**70.** B. New objects uploaded via PUT are subject to read after write consistency. Overwrite PUTs use the eventual consistency model.

**71.** C. This is important because it reflects a recent change by AWS. Until 2018, there was a hard limit on S3 of 100 PUTs per second, but that limit has now been raised to 3500 PUTs per second.

**72.** B. S3 buckets have names based upon the S3 identifier (s3), the region (us-west-1 in this case), and the `amazonaws.com` domain. Then, the bucket name appears *after* the domain. That results in B, `https://s3-us-west-1.amazonaws.com/prototypeBucket32`. Option A has an incorrect region, and both C and D have the bucket name in the domain, which is incorrect.

**73.** A. S3 buckets have names based upon the S3 identifier (s3), the region (us-east-1 in this case), and the `amazonaws.com` domain. Then, the bucket name appears *after* the domain. That results in a URL like `https://s3-us-east-1.amazonaws.com/prototypeBucket32`. However, buckets in US East are a special case and should use the special, unique endpoint *s3.amazonaws.com* (option A).

**74.** B, C. Option A is not the correct format; s3 should be separated from the region with a dash (-). Option B is valid, and option C is the correct unique URL for US East (N. Virginia). Option D is the right format, but jp-west-2 is not an AWS region.

**75.** A, D. S3 supports two styles of bucket URLs: virtual-hosted-style and path-style URLs. Virtual-hosted-style URLs are of the form `http://bucket.s3-`*`aws-region`*`.amazonaws.com`, and path-style URLs are the traditional URLs you've seen: `https://s3-`*`aws-region`*`.amazonaws.com/`*`bucket-name`*.

**76.** B, D. Option A is not a valid URL for S3. Option B is, using the path-style URLs that are most common for S3 buckets. Option C uses a nonexistent region (mx-central-1). Option D is valid and uses the virtual-hosted-style URL format.

**77.** D. AWS storage gateway is a virtual appliance that allows on-premises sites to interact with S3 while still caching (in certain configurations) data locally.

**78.** B. AWS storage gateway is a virtual appliance and is not available as a hardware appliance.

**79.** B, D. While S3 buckets are created in a specific region (A), the names of buckets are global and must exist in a global namespace (so B is untrue). Buckets are object-based (so C is true), and while a single object is limited at 5 TB, the buckets are unlimited in total storage capacity (so D is false).

**80.** A, D. S3 supports read after write consistency for PUTs of new objects and eventual consistency for overwrite PUTs and DELETEs.

**81.** C, D. S3 objects have a key, a value, and a version ID, so the correct answers are C and D.

**82.** C. MFA Delete is the absolute best means of ensuring that objects are not accidentally deleted. MFA—Multi-Factor Authentication—ensures that any object deletion requires multiple forms of authentication.

**83.** B. All Amazon-specific request headers begin with x-amz. This is important to remember as it will help eliminate lots of incorrect answers. This leaves only x-amz-mfa.

**84.** B, C. MFA Delete applies to deleting objects, not buckets (so option A is incorrect). It affects changing the versioning state of a bucket or permanently deleting any object (or a version of that object); this makes B and C correct. Deleting an object's metadata while leaving the object intact does not require MFA Delete.

**85.** A. This answer simply has to be memorized. MFA Delete authentication codes are pulled from hardware or virtual MFA devices, like Google Authenticator on an iPhone.

**86.** D. This is tricky and somewhat un-intuitive. Only the root account can enable MFA Delete. Even the console user that created the bucket—if it isn't the root user—cannot enable MFA Delete on a bucket.

**87.** B. The bucket owner, root account, and all authorized IAM users of a bucket are allowed to enable versioning.

**88.** A, B. Each object in S3 has a name, value (data), version ID, and metadata. The version history of an object won't exist unless versioning is turned on, so it's not always a valid answer.

**89.** B. All metadata in AWS is currently entered using tags, name-value pairs available through the console.

**90.** D. All versions of an object are stored, regardless of how that object is deleted.

**91.** D. Once enabled, it is not possible to disable or turn off versioning on an S3 bucket. While you can suspend versioning, this doesn't actually turn versioning off, and old versions are preserved.

**92.** B, C. CloudFront is intended to cache and deliver static files from your origin servers to users or clients. Dynamic content is also servable through CloudFront from EC2 or other web servers. Object-based storage doesn't make sense in this context, as CloudFront is a distribution mechanism, not a storage facility.

**93.** D. CloudFront serves content from origin servers, usually static files and dynamic responses. These origin servers are often S3 buckets for static content and EC2 instances for dynamic content.

**94.** A, D. CloudFront serves content from origin servers, usually static files and dynamic responses. These origin servers are often S3 buckets for static content and EC2 instances for dynamic content (options A and D).

**95.** B, D. CloudFront serves content from origin servers, usually static files and dynamic responses. These origin servers are often S3 buckets for static content and EC2 instances for dynamic content (meaning option C is valid). Containers can also be used in place of EC2 instances, making option A valid as well. This leaves B and D as invalid origin servers.

**96.** B. CloudFront stores content as cached content on edge locations across the world.

**97.** C, D. CloudFront is able to distribute content from an ELB, rather than directly interfacing with S3, and can do the same with a Route 53 recordset. These allow the content to come from multiple instances. This means that options C and D are invalid origin servers and therefore the correct answers.

**98.** D. A CloudFront distribution is a collection of edge locations across the world.

**99.** B. First, you can eliminate any answer where fewer availability zones are indicated than regions, because each region has multiple availability zones (A and D). This leaves B and C. There are more edge locations than availability zones, which means that B is correct.

**100.** A, B. This question is simple if you remember that, from most to least, the ordering goes edge locations (most) to availability zones to regions (least). Knowing that, options A and B are correct.

**101.** B, D. Availability zones are not content storage devices; they are virtual data centers. Edge locations are used by CloudFront distributions to store cached content (so correct). Route 53 is the Amazon DNS service. EC2 instances can serve content from processes (so also correct).

**102.** A, D. While edge locations are typically read from by clients, they are also writeable. You can store objects on edge locations as well as read from them.

**103.** A, C. The obvious answer here is an S3 bucket. EC2 locations are compute, not storage, and availability zones are virtual data centers. This leaves edge locations, which allow objects to be written directly to them.

**104.** A. TTL is Time to Live, the amount of time an object is cached on a CloudFront edge location.

**105.** B, D. You must perform *both* steps B and D, and you must perform B *before* D or the banner ad could get re-cached. Also note that expiring a cached object manually incurs a cost.

**106.** B. The default TTL for edge locations is 24 hours.

**107.** B. All new S3 buckets are private by default.

**108.** B, C. The correct answers are ACLs—access control lists—and bucket policies (B and C). NACLs are network access lists, used for securing VPCs and individual instances, and JSON is used for writing policies.

**109.** C. S3 bucket policies are written in JSON, the JavaScript Object Notation. XML is not used much in AWS, and YAML is often used for CloudFormation. AML is made up!

**110.** A. All data is backed up to S3 asynchronously when a stored volume is used. This ensures that no lag is incurred by clients that interact with the stored volumes on-site.

**111.** A. This is a little harder unless you've seen the term *virtual tape library (VTL)* before. A tape volume is in fact a virtual tape library. Fortunately, even if you've never heard of a VTL, you can reason it out based on the other incorrect options: A VPC is a virtual private cloud, a VPN is a virtual private network, and NetBackup is an application, not a tape volume.

**112.** A. This is an easy one: Snowball is the AWS solution for transferring large datasets.

**113.** D. Snowball actually does not support any code. You just transfer your data to the device and send it to Amazon. Additionally, CloudFormation is not a language; you use YAML (for example) to write CloudFormation templates.

**114.** A. AWS Direct Connect is a dedicated high-speed connection between your on-premises network and AWS. Because of this, a direct connect is almost always a better choice than shipping out a Snowball, loading data to it, and then shipping it back.

**115.** A. All Snowball devices provide data transfer, but Snowball Edge offers data processing ("at the edge") before that data is returned to AWS (option A).

**116.** C. Snowball can serve as both an import and export device, both to and from S3.

**117.** C. Decoupling separates application layers, primarily in an attempt to reduce failures across an entire application. Limiting application interdependence helps limit failures "crossing" application layers.

**118.** D. Redshift is an OLAP (online analytics processing) service suitable for data warehousing.

**119.** B. While using separate VPCs and subnets offers some degree of redundancy, there's nothing in the answers that suggests that additional VPCs or subnets are separated from the original. This makes B, launching instances in separate regions, the best choice. (Option D doesn't make much sense.)

**120.** A. CloudFront is the only option here that uses edge locations. Note that Snowball offers Snowball Edge, but that is not to be confused with an edge location.

**121.** C. This should be pretty basic: Running an application in two availability zones is going to provide fault tolerance, because if one AZ fails, the application will keep running.

**122.** B, D. While all of the answers are storage-based services, A and C are databases, and relational ones at that. Options B and D, S3 and EBS, offer file storage.

**123.** B, C. This is a little tricky. S3 is an obvious choice. Redshift is suited for analysis data, so probably not large objects. EC2 is compute, which leaves Oracle. It is possible—without any better answers—to use Oracle (via RDS or installed on EC2) to store large objects in a BLOB-type field.

**124.** B. S3 Transfer Acceleration speeds up transfers to S3, although it does add cost to your application. Snowball is for data migration (A). AWS manages the network, so C is not an option, and D doesn't make sense in this context.

**125.** B. It's the users who are the farthest from your buckets that benefit most. This is because these users are incurring the longest delays in uploading and are therefore affected the most by the benefits.

**126.** A, C. The key here is to understand which problems *will* be solved by Transfer Acceleration versus which ones *might* be solved. With Transfer Acceleration, you're generally looking at problems related to large datasets being transferred over significant distances. In this case, that's A and C. While performance (B) and latency (D) might be connected to transfer speeds, there's no guarantee of that, so those are both incorrect answers.

**127.** A, C. This should be easy. You can host websites on EC2 easily, and S3 also offers static website hosting.

**128.** C. This is a case of memorization. While B is a valid S3 bucket URL, it is not the URL for website hosting. D is in the incorrect region, and also has a dot instead of a dash between s3-website and the region name. This leaves A and C. C is correct, as A incorrectly breaks up the s3-website and region portions of the domain name, using a dot instead of a dash between s3-website and the region name.

**129.** D. First, ensure that the domain name is correct. Option A incorrectly separates s3-website from the region, and C has the wrong region. B does not have the bucket name in the URL, which it should for website hosting. This leaves D, the correct answer.

**130.** A. First, eliminate option D; the domain is incorrect, adding a separator between s3-website and the region. Then, eliminate option C, as it adds a `public_html` to the portion of the URL after the domain, which is also incorrect. This leaves A and B. Here, you need to realize that the portion of a URL after the domain is case sensitive and compare the two directories to the question. A is correct, using the correct capitalization of phoneboothPhotos.

**131.** A, D. To minimize compute resources, you should avoid EC2 and Lambda. Enabling static website hosting on an S3 bucket is a better option. To use a custom domain, you'd need to also use Route 53 to direct traffic from your custom domain to the S3 bucket.

**132.** C, D. If the website was static content, S3 bucket hosting would be ideal. However, to support dynamic content, you will need some sort of compute service: EC2 or Lambda. To minimize costs and avoid incurring additional costs when requests aren't occurring, Lambda allows as-you-need-it serverless responses to requests. Route 53 allows you to direct traffic aimed at your custom domain to Lambda.

**133.** A, C. EC2 is, by definition, a server-driven option. Both S3—through static website hosting—and Lambda offer serverless options for serving content. Route 53 provides DNS but is not on its own capable of serving website content.

**134.** B, C. While S3 can serve static content, it has no capability for processing code or otherwise providing dynamic content. EC2 provides for dynamic content through server-driven compute, and Lambda provides dynamic content through serverless compute.

**135.** A, C. Elastic Beanstalk is focused on code deployment. It provides that, and in the process, load balancing, Auto Scaling, health monitoring, and capacity provisioning (C).

**136.** B, D. Elastic Beanstalk is focused on code deployment (A). It provides that, and in the process, load balancing, Auto Scaling, health monitoring (C), and capacity provisioning. It does not provide security or log inspection.

**137.** A, D. This is a little far off the beaten AWS path, but you should know which languages and technologies are commonly used and cited by AWS and which are not. In general, Docker and containers are always supported; and Node.js, JavaScript, Java, PHP, and Perl are commonly supported. C++ and Scala are not in that list.

**138.** D. The best way to work this question is to immediately recognize the common languages: Node.js and Python. While it's possible that you forget Java is supported, once you've identified two options that are valid, the answer must be D.

**139.** A, B. Elastic Beanstalk supports all RDS options as well as DynamoDB. Oracle running on EC2 is a nonstandard option so not suitable for the auto-provisioning of Elastic Beanstalk. Redshift is also not a database technology.

**140.** C. You can absolutely deploy to production using Elastic Beanstalk. You simply need to configure different Elastic Beanstalk environments to use different databases.

**141.** D. EC2 and ECS are compute services but require knowledge and working with the required resources. DynamoDB is a database and cannot run code. Lambda is correct: It runs code without needing an underlying set of compute resources that are user managed.

**142.** A, D. Elastic Beanstalk and Lambda are very different services, but in this context, both are valid answers. Elastic Beanstalk is a sort of "code deployment wizard," and Lambda allows for serverless code deployment. Both handle provisioning of the environment without user intervention.

**143.** A. Code deployed via Lambda absolutely runs on servers; however, AWS abstracts the details away from user management.

**144.** A, B. You should know which languages and technologies are commonly used and cited by AWS and which are not. In general, Node.js, JavaScript, Java, PHP, and Perl are pretty commonly supported. C++ and Scala are not in that list.

**145.** B, C. A is a bit nonsensical. Installing Oracle requires using EC2 instances, so is not a use case for Lambda. Both B and C are valid: Lambda is ideal for responding to triggers or changes in state from other AWS services and also handles scaling quite well without user-driven code. D is not valid; that is a use case for EC2 or the Elastic Container Service.

**146.** A. Conversion of files to various formats is transcoding and therefore the function of Elastic Transcoder.

**147.** B. QuickSight is a cloud-powered business analytics service. It provides visualizations and analysis from multiple data sources.

**148.** B. SNS is the Simple Notification Service and provides notifications—alerts and alarms—when certain events occur in your AWS environment.

**149.** A. AWS Cognito allows you to add user sign-up, sign-in, and access control to web applications, as well as single sign-on. It also allows identity providers such as Facebook and Google to be used.

**150.** B. Every region is a specific geographic area, but none are as big as an entire continent.

**151.** D. A VPC is a virtual private cloud. It is a virtual network dedicated to a single AWS account.

**152.** C. ECS is the Elastic Container Service. It manages containers (Docker) that are each compute resources and can be spun up and spun down quickly.

**153.** B. RDS is the Relational Database Service, which provides managed database services for your applications.

**154.** D. Route 53 is the DNS service managed by AWS. It provides domain management and registration.

**155.** D. A customer gateway allows an on-premises site to connect with AWS through a peer-to-peer VPN. It is a virtual networking device.

**156.** A. Anything related to S3 is going to be storage-related. In this case, lifecycle management handles transitioning data from one S3 storage class to another.

**157.** D. Amazon Lightsail is a compute solution for web applications and involves compute, storage, and networking as well as database storage when needed. It launches servers and configures them with the needed services for web hosting. Note that while AWS considers Lightsail a compute service, it absolutely interfaces and controls additional resources.

**158.** C. Elastic Beanstalk is an Amazon service that spins up and manages a number of other services, in particular, compute. Even though you can configure other services, though, Beanstalk is considered to primarily be a code deployment tool and therefore is focused on compute services.

**159.** A. EFS is the Elastic File System, a scalable file system concerned with storage.

**160.** C. Redshift is one of AWS's OLAP (online analytics processing) tools and is a database service. While it does processing, it is primarily intended to receive large amounts of data and operate upon that data, as a database would (in loose terms).

**161.** B. CloudFront is AWS's distribution network. It's a content caching system that is ultimately a networking component of your AWS buildout.

**162.** D. Athena is a database and available through RDS but is ultimately intended for analytics, much like Redshift and Elastic MapReduce.

**163.** B. EMR is Elastic MapReduce and provides data processing and analysis of large datasets.

**164.** C. Cloud9 is a developer environment, intended as an IDE for AWS developers.

**165.** D. Direct Connect is an AWS service for creating a high-speed connection between an on-premises site and AWS.

**166.** D. Amazon Workspaces allows you to provide a desktop service via the cloud. The service allows people throughout the world to take advantage of scalable desktop provisioning.

**167.** B. Kinesis is a data analytic service capable of handling large data streams and providing real-time insights.

**168.** C. Elastic Transcoder processes video into formats suitable for a wide variety of devices across resolutions and formats.

**169.** D. OpsWorks is an operational management service, which AWS often classifies as "management tools" (especially in the AWS console). It allows integration with tools like Puppet and Chef.

**170.** A. Lex is the Amazon service for building voice recognition and conversation bots.

**171.** A. CloudWatch offers the ability to set up specific metrics (network throughput, requests, disk IO, and so on) and monitor those metrics via dashboards and reports.

**172.** D. An availability zone (AZ) is a virtual data center within a region, separated from other AZs by a distance sufficient to provide redundancy in the case of a disaster. B is incorrect as AZs do not have redundancy within them; that is the definition of a region.

**173.** B. A region is an area geographically that has redundancy within it, in the form of availability zones. Each AZ (which is defined in both A and C) is separate from other AZs and each is in essence a virtual data center.

**174.** A, B. AWS will always ask at least a few questions related to regions and availability zones. As long as you read these carefully, they should be easy correct answers. A region is a geographical area with redundancy within it, through at least two availability zones. Option A is false as services are not tied to regions. B is false because a region contains virtual data centers; it is not itself a virtual data center.

**175.** B, D. Availability zones are virtual data centers (which makes D false, as AZs do not *contain* data centers) and are isolated from each other except through low-latency network links. So C is true (and therefore incorrect), and A is true, as AZs definitely host compute resources. That leaves B in addition to D. B is not describing an AZ as an AZ does not itself provide redundancy. It's the *combination* of AZs that does this work.

**176.** C, D. Elastic IPs are assigned to an instance in a specific availability zone, but in the event of a failure, that elastic IP can be remapped to another AZ, making A false. B is false because regions will contain *at least* two availability zones, not exactly two. C is true, as different accounts may remap AZs to different names to ensure better resource distribution, and D is correct, even though many users simply accept the defaults and don't pick a specific AZ.

**177.** A, C. This is admittedly a tough question, but worth working through. You need to have at least a familiarity with AWS regions and know that there are several major regions: US, EU, and AP. There are a few others (CA, SA, for example), but the major ones are US, EU, and AP. Knowing those, you can spot that A and C are likely valid. JP (presumably for Japan) isn't correct, and UK you should recognize should be EU. There is no UK-specific region.

**178.** B, D. This is more of a formatting question than knowing the list of AWS regions. A region identifier is the region name, which is usually the country or area (eu, us, etc.), then the geographical area (southeast, west, east, etc.), then a number. This makes B and D correct. A is the "human readable" name, and C is an availability zone, due to the letter appended to the end.

**179.** A. An availability zone identifier is the region identifier with a letter appended on the end. A region identifier is the region name, which is usually the country or area (eu, us, etc.), then the geographical area (southeast, west, east, etc.), then a number.

**180.** C. EFS, Elastic File System, provides scalable storage accessible from multiple compute instances. EBS is Elastic Block Storage and is tied to one instance at a time and therefore not like a NAS (network attached storage). DynamoDB is a NoSQL database, and tape gateway is a client device for interacting with S3, but locally rather than in the cloud.

**181.** C. Be careful here; while ElastiCache is an AWS service for caching, it uses memcached and redis for actual caching. Therefore, memcached is the engine, not ElastiCache.

**182.** A, C. ElastiCache uses two different caching engines: memcached and redis.

**183.** C. You can choose reserved instances for EC2 and RDS, so the correct answer here is C.

**184.** D. You can use reserved instances for almost anything AWS offers. This applies to RDS (in any configuration) as well as ElasticCache nodes.

**185.** A, B. First, you need to know that you can manually force a failover (option A). Then, realize that a Multi-AZ setup is for disaster recovery, so it would take a failure of the primary availability zone to fail over (option B). The secondary AZ does not cause a failover (C), nor does a specific number of failed reads (D).

**186.** A, D. You can approach this in two ways: by knowing what you can do, and knowing what you cannot. You *can* select the database type and the AZ to which you want your instance deployed (A and D). You *cannot* specify extremely detailed issues related to performance and failover (B and C).

**187.** A, B. This is a bit tricky. While I/O may briefly suspend, the database never goes completely offline (C). However, the I/O suspension can result in increased latency (A) and a slowing down of responses (B). However, network requests will still be fulfilled; they will not fail (D).

**188.** C, D. RDS supports Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.

**189.** A, D. RDS supports Multi-AZ deployments. Automated backups are turned *on* by default. Some RDS databases—notably Maria and Aurora—are only supported through a managed service like RDS. And all RDS databases provide a SQL interface.

**190.** D. MySQL is by default available on 3306. You can also remember that databases are typically only available on unreserved ports, above 1024. Ports below 1024 are reserved for privileged services.

**191.** A. OLAP is online analytics processing, often associated with business intelligence. AWS services like Redshift are ideal for OLAP.

**192.** D. OLTP is online transaction processing and is generally the domain of relational databases in AWS.

**193.** A. Redshift is the prime example of AWS providing an OLAP service.

**194.** D. Aurora, as a managed service via RDS, is a relational database, and relational databases are generally the best answer for OLTP in AWS.

**195.** B, D. In OLTP questions, look for the relational databases. In this question, those are Oracle and SQL Server, and therefore the answers. memcache is one of the engines for ElastiCache and DynamoDB is a NoSQL database.

**196.** D. Redshift provides data warehousing on AWS, which is closely associated with OLAP.

**197.** A. EMR, Elastic MapReduce, is ideal for big data processing. Is uses the Hadoop and Spark frameworks and is a managed service for processing very large datasets.

**198.** C. This is a little trickier. The best way to remember how to answer a question like this is to associate Kinesis with streaming data, which implies real-time analysis. Kinesis can take in streams of data and do immediate processing on that.

**199.** B. QuickSight doesn't come up much in AWS study guides, but there is usually a single question on many AWS exams about it. QuickSight gives you "sight" into your application, which is the easiest way to remember it's about visualization.

**200.** D. This is another tough question, especially if both Kinesis and Athena appear in the answer choices. Kinesis handles streams of data and does real-time analytics; Athena is more on the interactive side. Athena analyzes data but allows standard SQL queries. That's why it's a better choice than Kinesis with this question.

**201.** B, D. EMR, or Elastic MapReduce, is most commonly used with Hadoop and Spark. Unfortunately, this simply has to be memorized; there's no good way to get at this unless you already know that Hadoop and Spark are ideal for data processing.

**202.** D. Aurora actually stores a whopping six copies of your data, across three availability zones, to ensure failover and disaster recovery.

**203.** B. Aurora actually stores a whopping six copies of your data, across three availability zones, to ensure failover and disaster recovery.

**204.** C. Aurora, under the RDS managed service, is about five times as fast as MySQL and three times as fast as PostgreSQL. Still, there's an easier way to remember this: Anytime an AWS exam asks you about speed or performance, it's generally the case that the AWS offering is the right answer. AWS won't ask you to choose MySQL or Oracle as a faster option than one of its own databases!

**205.** A. Aurora, under the RDS managed service, stores six copies of your data by default, across three availability zones. Additionally, there's an easier way to remember this: Anytime an AWS exam asks you about resilience, it's generally the case that the AWS offering is the right answer.

**206.** B, C. Aurora is compatible with both PostgreSQL and MySQL. These are also easier to choose because they are both relational databases, also managed through RDS.

**207.** B, D. RDS provides for SQL interaction as well as access through the RDS web APIs. RDS instances do *not* allow access via SSH or RDP.

**208.** C. RDS allows backup retention periods up to 35 days.

**209.** A. You can't use RDS because the question explicitly says you are installing Oracle on EC2 rather than using the managed service. In this case, then, you want the fastest disk space available, which will be EBS, Elastic Block Storage.

**210.** B, D. Anytime OLTP comes up, simply look for options that are RDS-supported databases, and if that fails, look for relational databases. In this question, the answers that fit these criteria are MariaDB and Aurora.

**211.** A, C. Anytime OLTP comes up, simply look for options that are RDS-supported databases, and if that fails, look for relational databases. In this question, the answers that fit these criteria are PostgreSQL and SQL Server. Since the question asks which are *not* suitable options, the correct selections are Kinesis (A) and Redshift (C).

**212.** A, C. A Multi-AZ setup provides disaster recovery options through a secondary database. This also implicitly provides data redundancy.

**213.** B, D. A read replica setup is intended to reduce the load on a single database instance by providing additional databases from which to read. This also has the "side effect" of reducing network latency via spreading out traffic across multiple instances.

**214.** A, C. A read replica setup is intended to reduce the load on a single database instance by providing additional databases from which to read. Applications can read from the replica (A) but not write to it (B). Only the primary instance—through RDS and AWS—can "write" changes to the replica (C).

**215.** C. Multi-AZ setups provide disaster recovery through a secondary instance (A and B), and all RDS databases support Multi-AZ (D). This just leaves C, which is not provided (and is the correct answer). Because only the primary instance is accessible, it is not any more performant than a standard RDS setup.

**216.** B. Multi-AZ setups use synchronous replication (B) to back up data to the secondary instance for the purposes of disaster recovery.

**217.** D. Read replicas use asynchronous replication (D), pushing data to the read replicas whenever possible, for improved read performance.

**218.** A. Read replicas are intended to provide scalability for your application by adding additional instances for increased reads from applications.

**219.** C. A Multi-AZ setup is about disaster recovery, and therefore durability. They provide automatic backups (so not A), upgrades happen on the primary database and then are replicated (so not B), and there is a primary and usually a single secondary instance (so not D). That leaves C: durability.

**220.** B. AWS provides up to five read replicas for a single database instance, configurable via the AWS console.

**221.** A, B. DynamoDB is a NoSQL database, and RDS is required for read replicas (so A is correct, as it does not support read replicas). Redshift is not a database, but rather a data warehousing tool, so also should be selected. Both MySQL and MariaDB support read replicas through RDS.

**222.** A, C. DynamoDB is a NoSQL database and does indeed offer "push-button" scaling (A). You can scale up the size of the database at any time, *without* needing to change the instance size (as you do with RDS instances). This makes C true.

**223.** B, C. DynamoDB is easier to scale than RDS as it does not require either read replicas or instance size changes to scale up, making A false. DynamoDB does use SSD drives, so B is true. It is also—according to AWS documentation—spread across three geographically distinct data centers, so D is correct.

**224.** A. DynamoDB uses eventually consistent reads by default, meaning a read might not immediately reflect the results of a very recent write. The remaining choices are not actual consistency models.

**225.** A, D. DynamoDB uses eventually consistent reads by default, meaning a read might not immediately reflect the results of a very recent write. It also offers a strongly consistent reads model, always reflecting the most recent write operations.

**226.** A, D. Delays occur in a strongly consistent read model when recently written data cannot be returned. Since a strongly consistent read model guarantees the latest data is returned, until that data is available, no response can be sent. This is the situation described in both option A and D. Option B involves replication, which is not relevant in this context, and C involves previous reads rather than writes.

**227.** D. VPCs put relatively few restrictions on the types and numbers of subnets supported. They can certainly support single and multiple private and public subnets alongside each other.

**228.** A, D. All instances in the default VPC get a public and private IP address by default at launch time.

**229.** C, D. All of the options are valid here. EC2 instances can have public and private addresses, elastic addresses, and both IPv4 and IPv6 addresses.

**230.** B, C. There is no concept of a VPC peering with itself, and VPCs can only peer with other VPCs, not subnets. This makes A and D incorrect. VPCs *can* peer with other VPCs, in the same account or different ones (B and C).

**231.** A. A /16 offers 65,536 IP addresses. The lower the number, the larger the pool of IP addresses when using CIDR notation.

**232.** A. SWF stands for Simple Workflow, and Amazon SWF is the Amazon Simple Workflow Service.

**233.** D. SWF places no language restraints on your workflow, as long as interactions can be managed via HTTP requests and responses.

**234.** B. SWF provides an API, but it is neither the AWS-specific API nor language specific. Instead, SWF supports standard HTTP requests and responses.

**235.** D. SWF stands for Simple Workflow, an AWS managed service. That should be a clue that the key factor here is workflow management. Tasks are handled and coordinated across application components with SWF.

**236.** C. SWF is typically thought of as an asynchronous service, but it also supports synchronous tasking when needed.

**237.** B. SES is the Simple Email Service and is used for sending and receiving emails for AWS applications and services.

**238.** D. SQS is the Simple Queue Service. This should be a tip-off, but it's actually *messaging* in the question that is the key word. SQS does provide queuing but is ultimately a queue-based message delivery system.

**239.** C, D. SNS is the Simple Notification Service and SQS is the Simple Queue Service. The two are not interchangeable (A is wrong). SNS pushes notifications, while SQS allows for pulls of its messages (so B is wrong, but D is correct). Finally, SNS handles notifications, and SQS handles messages (C is correct).

**240.** B, D. Worker nodes (D) can poll SQS for new messages (B) and then pull those messages when they are available. Tasks are associated with SWF, while notifications are associated with SNS.

**241.** B, C. SNS is a push-based service (C) that pushes notifications (B) to anything subscribed to an appropriate topic.

**242.** A, B. SWF is associated with tasks and is distinct from (for example) SQS, because it guarantees a single delivery of all tasks.

**243.** A, B. SNS provides topics that can be subscribed to; then notifications related to that topic are pushed to all the topic subscribers.

**244.** A. SWF tasks are assigned once and only once.

**245.** B. This is a bit esoteric, but even if you're unsure, you should be able to reason this one out. A topic is simply a name or "category" to which subscribers can attach and receive notifications. Therefore, a linked list and a named message don't make much sense. (They're also constructs that are never seen in AWS documentation for the most part.) An IAM role is an AWS construct, but roles are related to permissions. This leaves only B, an Amazon Resource Name, which is correct.

**246.** D. SQS will guarantee that a message is delivered at least once, but that message may be redelivered.

**247.** C. A SWF domain is a collection of related workflows.

**248.** D. SQS queues only make an "attempt" to deliver messages in order (more or less a FIFO approach) but do not guarantee FIFO. If strict FIFO is needed, that option can be selected.

**249.** B. SQS queues only make an "attempt" to deliver messages in order (more or less a FIFO approach) but do not guarantee FIFO. If strict FIFO is needed, that option can be selected. Option B will ensure that orders are processed in the order in which they were received.

**250.** C, D. Other than the slightly odd answer choices (which sometimes comes up!), all VPCs can communicate with the hub, so C and D cover all the options.

**251.** B, D. Any spoke in a hub-and-spoke model can only directly communicate with the hub (option B), as well as any other peered VPCs (option D).

**252.** B, C. Any spoke in a hub-and-spoke model can only directly communicate with the hub (option B is true, while A is false). And the hub (VPC G) can communicate with all spokes (so C is true, but D is false).

**253.** C, D. Any spoke in a hub-and-spoke model can only directly communicate with the hub. This makes A and B true and C and D false; so the right answers are C and D.

**254.** B. NACLs are stateless—rules and must exist for inbound and outbound. Security groups are stateful—anything allowed in is allowed back out automatically.

**255.** A. NACLs are stateless—rules must exist for inbound and outbound—and security groups are stateful—anything allowed in is allowed back out automatically.

**256.** B. NACLs are stateless—rules must exist for inbound and outbound—and security groups are stateful—anything allowed in is allowed back out automatically.

**257.** B. ALBs are redundant across at least two subnets.

**258.** A, D. This is a little tricky. While the default VPC automatically creates a subnet, additional VPCs do not. You do automatically get a security group, route table, and NACL, so in this case, you'd want to choose options A and D.

**259.** C, D. The key here is "default VPC." While subnets are not created in additional custom VPCs, the default VPC does get a subnet automatically (as well as an internet gateway). And all new VPCs get route tables, NACLs, and security groups.

**260.** A, C. The key here is "default VPC." While subnets are not created in additional custom VPCs, the default VPC does get an internet gateway automatically (as well as a subnet). And all new VPCs get route tables, NACLs, and security groups.

**261.** A. This is really tough and requires pure memorization. The default VPC has a CIDR block of /16, but the default subnet in each AZ is a /20.

**262.** B. This is a case of rote memorization. Default VPCs get a /16 CIDR block assigned to them.

**263.** D. There is no default CIDR block for custom VPCs. While the default VPC has a /16 CIDR block, custom VPCs must have this entered in.

**264.** B. In general, the smaller the number after the slash, the larger the CIDR block. /16 is the largest valid block. A /16 offers 65,536 IPv4 addresses.

**265.** C, D. Default VPCs have a default subnet, along with a NACL, security group, and internet gateway, and a route table as well.

**266.** B. The default VPC has an internet gateway, and instances are given public IP addresses, so option B is correct. You do not create the default VPC (A), and security groups control specific access, not the public or private nature of the VPC and instances within it (C).

**267.** A, B. The default VPC does have an internet gateway attached to it, but custom VPCs do not. This is an important exam topic!

**268.** A, C. Option A is true for both the default and custom VPCs: All VPCs have NACLs automatically created. While all outgoing traffic is allowed out by default (C), incoming traffic is restricted by default (B)—this includes inbound HTTP traffic (D).

**269.** A, D. All VPCs have NACLs, security groups, and route tables automatically created. However, only the default VPC has a default subnet and an internet gateway created as well.

**270.** B, D. All VPCs have NACLs, security groups, and route tables automatically created. However, only the default VPC has a default subnet and an internet gateway created as well, different from the custom VPC.

**271.** B, C. All EC2 instances in the default VPC have both a public and private IP address. They do *not* have an elastic IP address, and the security group that is created by default does not allow any inbound traffic (until changed manually).

**272.** C, D. All EC2 instances in the default VPC have both a public and private IP address. Therefore, the only addition to serve web content would be to allow the web traffic in via security group.

**273.** C, D. Instances in any non-default VPCs need to be made public via an elastic or public IP (A), and the VPC itself needs an internet gateway (B). Further, you need to allow in web traffic via the security group (C). So this is an "All of the above" situation, translating into options C and D.

**274.** B, C. A VPC endpoint provides a connection over the Amazon network between your VPC and a service, such as S3 (B). This avoids leaving the network and routing over the public Internet, which inherently provides greater security for the traffic involved (C).

**275.** D. A VPC endpoint does not require any of these to connect; it is a private connection outside of these constructs altogether, which is part of why it is an attractive solution for internal AWS communication.

**276.** B, C. A VPC endpoint is a virtual device that provides redundancy via AWS (and automatically). This makes options B and C correct, and A wrong. VPC endpoints scale horizontally, not vertically.

**277.** B, D. A VPC endpoint can connect to S3 and DynamoDB, as well as a host of additional AWS services, so B is true. It does not require an internet gateway or a VPN connection and does not route traffic over the public Internet (D).

**278.** A, C. A VPC endpoint comes in two flavors: an interface endpoint, which provides an elastic network interface and a private IP address, and a gateway endpoint, targeted for a specific route in your route table.

**279.** A, D. This is pretty tough and is arguably right at the boundary of what the CSA Associate exam might ask. A gateway endpoint handles all traffic for a supported AWS service. Further, it's not a specific portion of that service, so you can rule out a particular Kinesis data stream (C). That leaves A, B, and D. A and D make sense, while routing private traffic to Route 53 does not.

**280.** A, C. This is another tough question. An interface endpoint provides a private IP address for connecting to a specific entry point for a specific AWS service. Anything that's more general—like DynamoDB—isn't a valid candidate. Additionally, a VPN (B) doesn't make sense, as a VPN is a different type of connection altogether. In this case, that leaves a specific API gateway and a specific Kinesis data stream (A and C).

**281.** C, D. Instances that take advantage of a VPC endpoint do not need to have a public IP address or use a NAT instance. Instead, assuming they have a route to the endpoint (D), they send traffic over the AWS network to the connected service (C).

**282.** C. The best way to remember this is to consider the process for creating an instance: you must select the security group for every instance. So security groups operate at the instance level (C).

**283.** A. Security groups only provide for allow rules (A). All other traffic is automatically denied, so allow rules are the only means of allowing traffic in.

**284.** A, C. Security groups disallow all traffic unless there are specific allow rules for the traffic in the security group.

**285.** A. Security groups evaluate all the rules on the group before deciding how to handle traffic.

**286.** D. Security groups evaluate all the rules on the group before deciding how to handle traffic.

**287.** B. Five VPCs are allowed per region, per account, unless you contact AWS to raise this default limit.

**288.** B. All custom VPCs have a route table (so A is false) and a NACL (so C is false) and will *not* have an internet gateway (D is false). This leaves B, which is true: subnets can communicate with each other across availability zones by default.

**289.** D. Only a bastion host (D) makes SSH available to private instances. You can use a NAT gateway or NAT instance to route traffic from these instances out, but a bastion host allows for SSH into private instances.

**290.** A, C. Both a NAT instance and a NAT gateway provide for outgoing traffic to route to the Internet from instances within a private subnet.

**291.** A. A VPC can only have a single internet gateway.

**292.** C. A single region can only have five VPCs by default, but this limit can be raised by contacting AWS.

**293.** C. A single VPC can have a single internet gateway. This limit isn't based on region (D) but on VPC (C).

**294.** A, D. First, realize it's possible that almost any of these answers could be a part of a larger solution. However, the question asks for the simplest—or most direct—solutions. Given that, the solutions that are best are giving the instances public IP addresses (D) and adding an internet gateway to the VPC. You also will likely need routes in and out, security groups, etc.

**295.** B, C. Given the internet gateway, the most likely issues are the instances being accessible via IP (which C addresses) and traffic for web/HTTP being disallowed (B).

**296.** D. VPCs can have a single internet gateway and multiple subnets. However, instances within a VPC with a public address have that address released when it is stopped and are reassigned a new IP when restarted.

**297.** B. A VPC can peer with unlimited other VPCs, so B is false. A subnet cannot span AZs, a VPC can peer with VPCs in other accounts, and a VPC having an internet gateway has no bearing on the public or private status of subnets within it.

**298.** D. All of the statements about NAT instances are false in A through C. Further, a NAT gateway is preferable to a NAT instance because it is managed by AWS rather than you, the architect.

**299.** D. A VPC cannot be changed from dedicated hosting tenancy to default hosting. You have to re-create the VPC.

**300.** A. Changes to a security group take place immediately. As a note, option D is a bit misleading. While security groups operate at various levels, they absolutely affect VPCs, so D is false.

**301.** A. This is a routing question. Instances need to have their outbound traffic directed to the internet gateway on the VPC, and then that traffic can flow outward to the Internet.

**302.** A. CloudFront supports both static and dynamic content.

**303.** A, C. With only the information presented, the best options are to focus on the database and the dynamic content; the web application servers (from the question's limited information) are not the issue. That means look at the database instance size (A) and caching dynamic content (C). B and D focus on the web app instances, which would not appear to be the issue.

**304.** B, C. An internet gateway is required to handle Internet traffic, and a VPC endpoint is ideal for connecting the instances to S3. A customer gateway is used in setting up a VPN or site-to-site connection, and if NACL changes are required, you'd make them to the existing NACL, not a new one.

**305.** C, D. The key here is recalling that the default VPC already has an internet gateway attached, so you wouldn't need one (B). A customer gateway is for a VPN or direct connection. This leaves C, a VPC endpoint for communication with S3, and D, updated NACL rules for the endpoint and the gateway (potentially).

**306.** A, D. The most likely culprits are the routing table of the VPC subnet and the virtual private gateway. A storage gateway (B) is not part of a Direct Connect solution, nor is a NAT instance (C).

**307.** B. Route propagation is a routing option that automatically propagates routes to the route tables so you don't need to manually enter VPN routes. It's most common in a Direct Connect setup. A is too broad a statement—not all routes are automatically copied. C is incorrect, and in D, a storage gateway is not part of a Direct Connect solution (it can be, but isn't required).

**308.** B. This is a matter of rote memorization. All metadata for instances is available at `http://169.254.169.254`, at `/latest/meta-data`. `/latest/instance-data` is actually not a URL that is responsive to requests.

**309.** A. S3 is highly durable and stores data as key-value pairs.

**310.** B. B is the only answer that doesn't presume at least semi-frequent access. Glacier is best for files that are rarely accessed and do not require quick access times.

**311.** A, D. The best answer here is to enable MFA Delete (D). However, to do this, you'll also need versioning (A). It is not practical to disallow developers from all delete access (B), and signed URLs do not help the issue.

**312.** C. For all new AWS accounts, 20 instances are allowed per region. However, you can increase this limit by requesting it via AWS support.

**313.** C. The only one of these that makes sense is C, increasing the size of the NAT instance. It is impossible to add an additional internet gateway to a VPC that already has one (A), and adding an additional elastic IP requires using a newer EC2 instance, and it will not affect performance in this case (B).

**314.** B. If instances are scaling up and down quickly, this means that the thresholds for adding and removing instances are being met frequently. Since you don't want to reduce the scaling up to meet demand, you should increase what it takes for the system to scale down; that's what B suggests. Proactive cycling (A) won't help the situation and C is completely made up.

**315.** A, C. Routing is one of the most important steps (A); you must set the route to the public Internet to go to the NAT instance. Additionally, you need to disable source/destination checks, a commonly forgotten step (C). The NAT instance *cannot* be in a private subnet (B), and D doesn't make sense in this context.

**316.** B. This is a tough one because it must simply be memorized. CloudWatch provides disk read operations, CPU usage, and inbound network traffic but does *not* provide memory usage by default.

**317.** A, B. The instance will need an elastic IP for public communication (A) and should be behind the same ELB as the other instances (B). Adding it into a private subnet (C) will remove its ability to communicate with the public Internet. D looks good, but if the instance is in the same subnet as the other instances, it automatically gets their routes; routing tables apply to the subnet, not a specific instance.

**318.** D. The public Internet is addressed via 0.0.0.0/0.

**319.** A. The public Internet is addressed via 0.0.0.0/0, so if that's the destination, the target should be the internet gateway within the VPC.