

Chapter 10

Determining High-Performing Database Solutions

This chapter covers the following topics:

- [AWS Cloud Databases](#)
- [Amazon Relational Database Service](#)
- [Amazon Aurora](#)
- [Amazon DynamoDB](#)
- [Amazon ElastiCache](#)
- [Amazon Redshift](#)

This chapter covers content that's important to the following exam domain and task statement:

Domain 3: Design High-Performing Architectures

Task Statement 3: Determine high-performing database solutions

Just as it's likely that you're using networking services at AWS, you are almost certainly using one or more databases for your workload data records. I can't think of a single application that's hosted in the cloud or on premises that doesn't have a

backend database. As the saying goes, “If you have data, it should be stored in a database.”

SQL databases have been around for decades, but there are other database choices available for consideration. There are a number of NoSQL databases, such as Amazon DynamoDB service, for customers who need databases that provide single-digit-millisecond performance with unlimited throughput and storage, and automatic multi-region replication if desired. In order to pass the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to know about the architecture of the database offerings at AWS, how they are deployed, and how they are constructed for durability and failover. You don’t have to be a database administrator, but you need to be able to provide basic technical advice about the available AWS database offerings.

“Do I Know This Already?”

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 10-1](#) lists the major headings in this chapter and

their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A](#), “[Answers to the ‘Do I Know This Already?’ Quizzes and Q&A Sections.](#)”

Table 10-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
AWS Cloud Databases	1, 2
Amazon Relational Database Service	3, 4
Amazon Aurora	5, 6
Amazon DynamoDB	7, 8
Amazon ElastiCache	9, 10
Amazon Redshift	11, 12

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment.

Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What is Amazon's key database offering that supports key-value tables?

1. Amazon ElastiCache for Memcached
2. RDS
3. Amazon DynamoDB
4. Amazon Keyspaces

2. Which cloud database is suggested for storing user session information?

1. Amazon DynamoDB
2. Amazon DocumentDB
3. Amazon RDS
4. Amazon ElastiCache for Memcached

3. After ordering and configuring Amazon Relational Database Service, which of the following is true regarding the ongoing maintenance of the database infrastructure?

1. Each customer must perform all the maintenance.
2. AWS performs all maintenance and failover management.
3. The customer is responsible for making manual snapshots of the database instances.
4. The customer is responsible for changing the size of the database instances.

4. Which of the following is a suggested best practice when deploying Amazon RDS?

1. Use a single availability zone.
2. Use just a single primary database instance.
3. Use multiple primary database instances.
4. Use multiple availability zones.

5. What type of storage is used by Amazon Aurora?

1. EBS storage arrays
2. A virtual SSD cluster storage volume
3. Ephemeral storage
4. SSDs

6. How many copies of data does Amazon Aurora maintain?

1. Four copies stored in multiple availability zones
2. Multiple copies stored in Amazon S3 storage
3. Six copies stored across three availability zones
4. A single copy

7. Amazon DynamoDB performance is defined using what type of units?

1. Read units
2. Capacity units
3. Write units
4. Read and write units

8. What type of Amazon DynamoDB table can you create to span multiple AWS regions?

1. Regional table
2. Multi-region table
3. Global table
4. Amazon DynamoDB table

9. Why is Amazon ElastiCache operationally fast?

1. It performs execution in RAM.

2. It executes using SSDs.
3. The size of the instances selected for the cluster is optimal.
4. It uses SSDs with maximum IOPS.

10. Which of the following deployments of ElastiCache is not persistent?

1. Memcached
2. DAX
3. Amazon Redis
4. Amazon DynamoDB

11. Where do Amazon for Redshift continuous backups get stored?

1. Local hard drives
2. Leader nodes
3. Compute nodes
4. Amazon S3 storage

12. What type of queries can be performed on an Amazon for Redshift cluster?

1. Python queries
2. SQL queries
3. JSON queries

4. Complex queries

Foundation Topics

AWS Cloud Databases

Amazon offers over 15 database engines to support a variety of data models, including relational, key-value, document, in-memory, graph, time-series, wide-column, and ledger databases. AWS fully managed database services are continually monitored, have self-healing storage, and in many cases offer automated scaling solutions. Databases can be deployed across multiple availability zones and multiple regions. [Table 10-2](#) summarizes the available database types and services at AWS.



Table 10-2 Database Choices at AWS

Database Type	Use Case	AWS Service
---------------	----------	-------------

Database Type	Use Case	AWS Service
Relational	E-commerce, traditional applications, ERP, CRM	Amazon Aurora, Amazon RDS, Amazon Redshift
Key-value	High- performance web applications, e- commerce, gaming systems	Amazon DynamoDB
In- memory	Caching, user session management, gaming leaderboards	Amazon ElastiCache for Memcached, Amazon ElastiCache for Redis, Amazon MemoryDB for Redis

Database Type	Use Case	AWS Service
Document	Content management, user profiles	Amazon DocumentDB (with MongoDB compatibility)
Wide-column	Equipment maintenance, fleet management	Amazon Keyspace
Graph	Social networking, recommendation engines	Amazon Neptune
Time-series	IoT applications, DevOps	Amazon Timestream

Database Type	Use Case	AWS Service
Ledger	System of record, supply chain, banking transactions	Amazon Ledger Database Services (QLDB)

Amazon Relational Database Service

Databases at AWS can be hosted by several managed database services, the most popular of which is Amazon Relational Database Service (RDS). Amazon RDS hosts a variety of popular relational database engines, as shown in [Table 10-3](#). RDS is a completely managed database service. Focus on your data records and leave the running, monitoring, backup, and failover of your database instances to AWS. If you want to maintain complete control, however, you can build your own EC2 database instances and manage every aspect of your database infrastructure as a self-managed infrastructure as a service (IaaS) deployment by deploying RDS Custom.

Table 10-3 RDS Database Engines

Database Engine	Data Recovery Support	SSL/TLS Support	Replication	Encryption
MariaDB 10.0.1– 10.0.6	InnoDB Version 10.2 and XtraDB Versions 10.0 and 10.1	yaSSL, Open SSL, or TLS	Point-in- time restoration	AES
MySQL 5.5–8.0	InnoDB	yaSSL, Open SSL, or TLS	Point-in- time restoration	AES
SQL Server 2008–2019	All versions support data recovery	SSL	SQL Server database mirroring, SQL Server AlwaysOn	TDE

Database Engine	Data Recovery Support	SSL/TLS Support	Replication	Encryption
Oracle Database 12c and 11g	All versions support data recovery	SSL or NNE	Point-in-time restoration	AES
PostgreSQL 9.6, 10–14	All versions support data recovery	SSL	AWS synchronous replication	AES
RDS Custom	The Amazon RDS database service supports custom operating system and database environment. You can have access to the database and underlying operating system to install patches, and enable native features to meet your applications' requirements. RDS Custom supports many database engines.			



Thousands of AWS customers have decided that they need databases but don't want to manage the database infrastructure anymore. The essential component that makes up the relational database service hosted by RDS and managed by AWS includes the complete management of the instances hosting your data, automatic backups, synchronous replication from the **primary database** instance to the **standby database** instance, and automatic failover and recovery, as required.

Amazon RDS Database Instances

Under the hood of all Amazon RDS deployments is a familiar compute component: the EC2 instance. When you order an RDS database instance, you order the CPU, memory, storage, and required storage performance (input/output operations per second [IOPS]). These initial selections can also be resized later. RDS supports a variety of standard, memory-optimized, and burstable performance EC2 instances that support Intel hyper-threading technology, which allows multiple threads to run concurrently on a single vCPU core. Threads connect the physical processor on the bare-metal server to the virtual CPUs (vCPUs). AWS RDS pricing supports on-demand and RDS Reserved Instance (RI) pricing. The odds are that you want to use RI pricing because it provides price breaks up to 70%.

Amazon RDS database instance data is stored on Amazon Elastic Block Store (EBS) volumes, which are automatically striped to provide additional performance. EBS volume types can be either general-purpose SSDs, provisioned IOPS SSDs, or magnetic hard drives.

Note

Magnetic drives, which are limited to 4 TB, are available for backward compatibility because some customers still use them. For production databases, Amazon does not recommend using magnetic drives.

MySQL, MariaDB, Oracle, and PostgreSQL volumes can be from 20 GB to 64 TiB in size. Microsoft SQL Server is limited to a maximum of 16-TiB EBS storage volumes. General-purpose SSD storage EBS volume uses burst credits, which provide sustained burst performance depending on the size of the volume. Customers can deploy up to 40 Amazon RDS database instances for hosting MySQL, MariaDB, or PostgreSQL databases; these numbers are defined by your AWS account service quota limits and can be increased upon request. For typical production databases, the recommendation is to use EBS data storage

volumes with provisioned IOPS, which can be set from 1,000 to 256,000 IOPS, depending on the database engine.

Note

For production databases, use Multi-AZ deployments with provisioned IOPS for the primary and standby databases and ***read replicas***.

As mentioned earlier, as needs change, you can vertically scale the size of your RDS database compute and resources and also change the size, type, and IOPS of your storage volumes, as shown in [Figure 10-1](#). While changes are being made to a database instance, the database instance being resized is unavailable. Since RDS database instances are hosted and maintained by the Amazon RDS service, there is no direct access to a database instance; AWS carries out the backup, patching, monitoring, and recovery of each RDS instance.



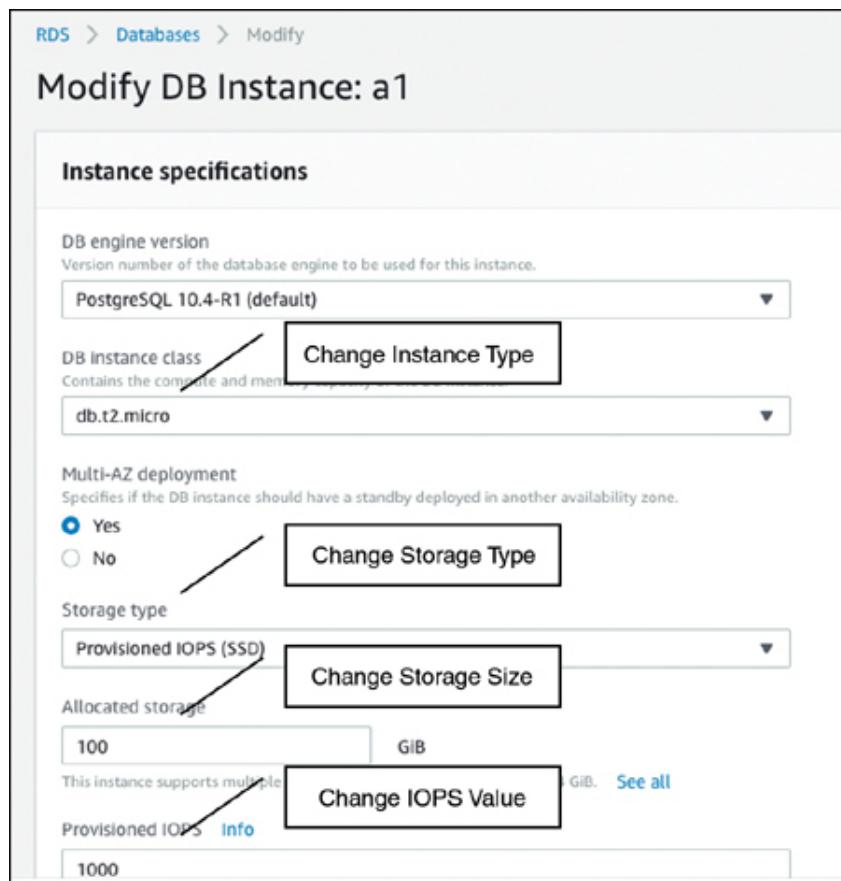


Figure 10-1 Changing Database Instance Parameters

Amazon RDS supports MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server database engines. Note that not every version of a supported database engines is supported by Amazon RDS. You must carefully check out the database engine versions that are currently supported and decide whether Amazon RDS suits your requirements.

If you do decide to build and maintain your own database instances, you will have to provision the desired EC2 instance

size, attach the appropriate EBS volumes and provisioned IOPS, and define and maintain your own backup and DR schedule.

For custom database deployments customers are also responsible for monitoring the database instances and carrying out all the required maintenance, including failover and recovery when required.

Database Instance Class Types

The database instance class you choose determines the amount of compute, memory, and network speed assigned to the RDS instance. Amazon RDS supports three types of instance classes:

- **Standard:** These instance classes range from general-purpose instances optimized for low latency and high random I/O performance and high sequential read throughput. A good modern example to start testing with would be the general-purpose m5 instance, which is hosted by the Nitro hypervisor supporting local NVME SSD storage up to 3.6 TiB with network speeds up to 100 Gbps.
- **Memory optimized:** These instance classes are designed for memory-intensive databases. For example, the r5b instance is powered by the Nitro hypervisor and can deliver up to 60-Gbps bandwidth and 260,000 IOPS of EBS performance.

- **Burstable performance:** These instance classes provide a baseline of performance with the ability to burst to full CPU usage when required. These general-purpose instances may be useful for initial testing purposes.

Note

You can review the full specifications for RDS instance classes at

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.DBInstanceClass.html>.

High-Availability Design for RDS



Production databases can and should at a minimum utilize Multi-AZ deployments, which provide automatic failover support with the primary database instance located in one AZ and the standby database instance, or instances located in another AZ. The technology used for RDS failover from the primary to a standby instance depends on the database engine that is deployed. Most Amazon RDS deployments use AWS's failover technology; however, Microsoft SQL Server uses SQL

Server mirroring to perform failover. Note that Amazon RDS failover design is an active-passive design that will typically take 30–40 seconds to complete the failover process once enabled.

When you order a Multi-AZ RDS deployment, Amazon provisions and maintains the primary and standby replicas in different AZs, as shown in [Figure 10-2](#). The primary database instance data records and any changes are automatically synchronously replicated to the standby replica, ensuring that there are always two copies of up-to-date database data records and data redundancy. The transaction logs are also backed up every 5 minutes. The process of synchronous data replication between the Amazon RDS database instances creates increased write and commit latencies between the primary and the standby instances. Therefore, your Amazon RDS database instances and volumes must be properly sized with the required computer power to be able to perform synchronous replication quickly and reliably without affecting the overall performance required for your database.

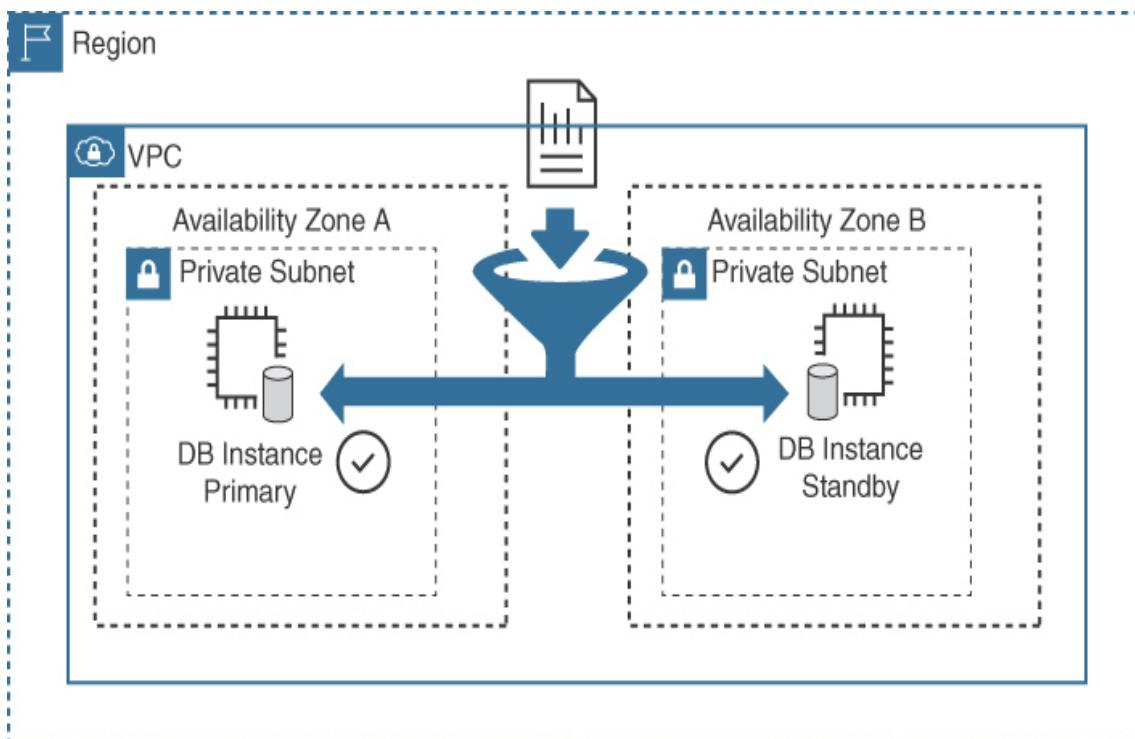


Figure 10-2 RDS Multi-AZ Synchronous Replication

Careful monitoring of both Amazon RDS database instances utilizing CloudWatch metrics such as CPU Utilization, Read Latency, Write Latency, and FreeableMemory help indicate when the existing database instances may need to be resized. Each Amazon RDS metric can be associated to an alarm; for example, an alarm that signals when the database instance CPU utilization exceeds 65%. The Amazon Simple Notification Service (SNS) can notify when a defined alarm threshold is exceeded.

It is important to choose a database instance with adequate storage and speed: either general-purpose SSDs or provisioned IOPS, depending on your production requirements. (IOPS represents how fast your hard drive can read and write per second.) A starting point might be CPU, RAM, and storage values equal to what the database is currently using while running on premises. When deploying Amazon RDS in production, make sure you have allocated enough RAM to each database instance, ensuring that your working data set can reside completely in memory. To test whether your working data set can operate while being completely contained in memory, use the RDS metrics ReadIOPS and WriteIOPS. When your Amazon RDS database instances are under load, both metric values should be less than the allocated IOPS. For further details, check out the RDS documentation, which will help you fine-tune requirements based on the RDS engine you are considering deploying:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BestPractices.html#CHAP_BestPractices.DiskPerformance

Note

For RDS deployments, the standby database instance is treated as a true standby and is not

used for any query requests.

The standby database records are kept up to date through synchronous replication from the primary database instance. The standby DB is kept up to date and ready to take over as the primary database when problems occur. To help improve the performance of database queries, additional read replicas can be created for MariaDB, MySQL, Oracle, PostgreSQL, and Microsoft SQL Server engines using AWS RDS. Deploy the read replicas if necessary in a specific AWS region to support additional query requests, and make sure to size the read replica instances the required compute power and storage size. Read replicas can be located in the same AWS region or in a different region from where the primary database is located, depending on your requirements.

A single AZ, two AZs, or a Multi-AZ DB cluster can be selected for deploying the MariaDB, MySQL, Oracle, PostgreSQL, and Microsoft SQL Server engines. Selecting the Multi-AZ cluster option creates a DB cluster with a primary DB instance and at least one standby DB instance deployed in a

different AZ. Amazon Aurora is deployed across three AZs per region or as a global database across multiple AWS regions.

The automated Amazon RDS database failover process swings into action when problems occur, such as a failure of the primary database instance or an AZ failure. Failover can also occur when maintenance is being performed, such as when the primary and secondary database instance types are resized, or software patching is required.

When failure occurs, during the failover process, RDS automatically switches over to the standby replica, as shown in [Figure 10-3](#). The standby replica database instance becomes the primary database instance. Route 53, the AWS DNS service, modifies the Amazon RDS [**endpoint**](#) to point end users to the new primary database instance (formerly the standby replica); this process should happen quickly, typically within a few minutes. Re-establishing the availability of a new standby database instance might take a bit more time because the standby EC2 instance has to be built and backups (snapshots) will have to be restored. After the new standby replica is re-created, to ensure that all changes have propagated from the primary database to the standby database, Amazon replays the

redo log from the last database checkpoint, making sure all changes have been applied before the new standby database instance is available. AWS recommends that the time to live (TTL) for the CNAME record of 30 seconds be set at the end user location to ensure a timely failover.

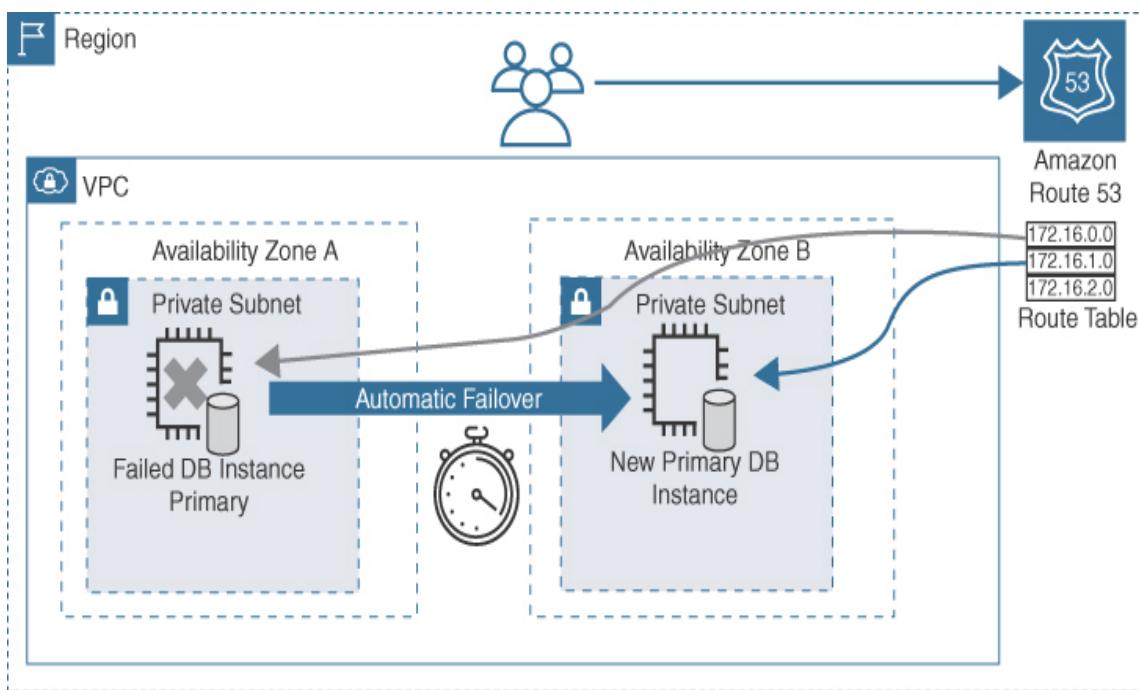


Figure 10-3 RDS Failover

The failover and recovery process isn't magic. It's simply an automated recovery process that Amazon carries out on your behalf. Other real-world issues might also be happening across the network; Amazon Route53 updates or database restorations may result in a longer recovery time than expected.

Note

You can create database SNS notifications so that you are notified via text or email when a failover or recovery is under way. In addition, you can configure your Amazon RDS database backups to automatically replicate the associated snapshots and transaction logs to a specific AWS region for safekeeping.

Multi-AZ RDS Deployments

An Amazon RDS Multi-AZ deployment provides additional availability and durability for your data records; after all, with such a deployment, there are at least two database instances running in separate AZs with separate EBS storage volumes. When you order a Multi-AZ database deployment, after the primary database instance is created, Amazon RDS takes a snapshot of the primary database instance's EBS volume and restores it on a newly created standby database replica located in another AZ and then synchronizes the two database instances' database volumes.

Big-Picture RDS Installation Steps

The process for installing a database using Amazon RDS is similar for all the supported database engine types except for Amazon Aurora. After you select the database engine to deploy, choose the database instance details. [Table 10-4](#) details the initial options depending on the database engine selected.

Key Topic

Table 10-4 Initial Amazon RDS Setup Options

Database Instance Setting	Details
License model	Bring your own license (BYOL) or general-purpose license included in the price
Database engine version	Select desired version to deploy

Database Instance Setting	Details
Database instance	Standard, memory-optimized, or burstable performance
Multi-AZ deployment	Synchronous AWS replication service; Native Mirroring or Always On for SQL Server
Multi-AZ DB cluster	DB cluster with a primary DB instance and two readable standby DB instances in different availability zones.
Storage type	SSD, provisioned IOPS, or HDD volumes
Amount of storage to allocate	1–64 TB (based on EBS volume types chosen)

Database Instance Setting	Details
Database instance identifier	Unique identifier, if required by database engine
Primary username and password	For database authentication and access

[**Table 10-5**](#) shows the advanced database instance options you can configure.

Table 10-5 Advanced Amazon RDS Setup Options

Advanced Database Instance Setting	Details
---	---------

Advanced Database Instance Setting Details

Database port The database engine default value

VPC The virtual private cloud (VPC) to host/link to the database instances

Database subnet group A predefined subnet for the database instance

Public accessibility Private by default

Availability zone The number of AZs to use

Security group Firewall settings for controlling access to a database instance

Advanced Database Instance Setting Details

Database name	A unique database name
Database port	The default access port of the database engine
Parameter group	A predefined group with a defined database engine, database instance specs, and allocated EBS storage
Option group	Additional features for the database engine, such as encryption
Copy tags to snapshot	Tags to be added to database snapshots
Encryption	The encryption type, which depends on the database engine deployed

Advanced Database Instance Setting Details

Backup retention The number of days automatic backups of the database are retained

Backup window The specific time for database backup

Enhanced monitoring Gathering of metrics in real time

Log exports Select logs published to CloudWatch log groups

Auto minor version upgrade Minor database engine version upgrades that occur automatically

Maintenance window Defined window to apply database engine modifications

Monitoring Database Performance

Once your Amazon RDS database has been deployed, establish an Amazon CloudWatch baseline using AWS RDS metrics to monitor the ongoing performance of the database at different times of the day to establish an acceptable level of operation. You can use the RDS Management Console to select a variety of metrics that allow you to monitor the number of connections to the database instance, read and write operations, and the amount of storage, memory, and CPU being utilized. [Table 10-6](#) lists some of the CloudWatch metrics that can be linked with alarms alerting you when issues occur.

Table 10-6 CloudWatch RDS Metrics

Metric	Description	Reporting	Values
Read I/O per second	Input and output and write operations per second	Average read/write per second	IOPS

Metric	Description	Reporting	Values
Read and write latency	Time it took from request to completion	1-minute interval	IOPS
Throughput	Bytes transferred to or from the database volume	1-minute interval	Megabytes per second
Queue depth	I/O requests waiting to be carried out	1-minute interval	From zero to several hundred queue entries

Best Practices for RDS

**Key
Topic**

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to be aware of a number of best practices related to AWS RDS deployment:

Note

For real-world deployments, be sure to check the best practices for each database engine by reviewing the documentation from each vendor.

- Define scaling storage capacity limits that allow your MySQL database instance storage to ***scale out*** as required.
- Match provisioned IOPS storage with the desired EC2 instance for the best performance.
- Monitor your infrastructure with CloudWatch metrics and alarms to ensure that you are notified when you are about to overload your capacity.
- Monitor AWS RDS database performance to define what is acceptable as normal operation. Define baselines for the minimum, maximum, and average values at defined intervals (hourly, one-half day, 7 days, 1 week, and 2 weeks) to create a normal baseline.

- Evaluate performance metrics such as CPU utilization, available memory, amount of free storage space, read/write metrics (IOPS, latency, throughput), network receive and transmit throughput, database connections, high CPU or RAM consumption, and disk space consumption.
- For the best performance, ensure that each AWS RDS database instance has enough allocated RAM so that the working set of the database resides in memory.
- Use AWS Identity and Access Management (IAM) users and groups to control access to AWS RDS resources.
- Use AWS RDS metrics to monitor your memory, CPU, replica lag, and storage usage.
- Enable automatic backups and define the backup window, picking a time when backups will be least disruptive (for example, in the middle of the night).
- For client applications caching the DNS data records of your AWS RDS DB instance, set a TTL value of less than 30 seconds to ensure faster connectivity after a failover has occurred.
- Test the failover process for your AWS RDS DB instances and document how long the failover process takes. Also confirm that the application that regularly accesses your database can automatically connect to the new database instance after failover has occurred.

Amazon Relational Database Service Proxy

Amazon Relational Database Service (RDS) Proxy is a fully managed, highly available database proxy for Amazon RDS that makes it easier to connect to your database from your applications. RDS Proxy can improve the reliability and performance of your database-driven applications by automatically routing connections to the appropriate RDS DB instance, based on connection and workload patterns. It also helps you scale your applications more easily by automatically distributing connections among multiple RDS DB instances.

Here are some key benefits of using RDS Proxy:

- **Application availability:** It can automatically failover to a standby RDS DB instance if the primary instance becomes unavailable, ensuring that your application remains available even if the database fails.
- **Better connection performance:** It can cache connections and reuse them for subsequent requests, reducing the overhead of establishing new connections.
- **Application scaling:** It can distribute connections among multiple RDS DB instances, helping you scale your application more easily as demand increases.
- **Enhanced security:** It supports SSL/TLS encryption for connections to your RDS DB instances, helping you protect

sensitive data.

To use Amazon RDS Proxy, configure your application to connect to the RDS Proxy endpoint instead of the RDS DB instance (see [Figure 10-4](#)). RDS Proxy then routes connections to the appropriate backend RDS DB instance based on the configured routing rules.

Proxy configuration

A proxy makes applications more scalable, more transparent to database failures, and more secure.

Engine family [Info](#)

MySQL
Supports Aurora MySQL, RDS for MariaDB, and RDS for MySQL

PostgreSQL
Supports Aurora PostgreSQL and RDS for PostgreSQL

SQL Server
Supports RDS for SQL Server

Proxy identifier
Enter a name for your proxy. The name must be unique across all proxies owned by your AWS account in the current AWS Region.

Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens or end with a hyphen.

Idle client connection timeout
Idle connection from your application are closed after the specified time.

▾ hours ▾ minutes

The minimum is 1 minute and the maximum is 8 hours.

Figure 10-4 RDS Proxy Configuration

Amazon RDS Cheat Sheet

**Key
Topic**

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of RDS:

- A database subnet group is a collection of subnets designated for database instance deployment.
- When using encryption at rest, database snapshots, backups, and read replicas are all encrypted.
- With Multi-AZ deployments, you are not charged for synchronous database data replication from the primary database to the standby database in a single AZ or across multiple AZs.
- Changing the DB instance class or adding additional storage can be set to be applied during the specified maintenance window.
- During failover, configuration information, including the updated DNS primary location, is updated to point to the new primary database instance.
- You can have five read replicas for MySQL, Maria, PostgreSQL, and SQL.
- A read replica can be manually promoted to become a primary database instance.

- A restored database is a separate new RDS instance with a new DNS endpoint.
- The AWS Database Migration Service migrates the most widely used commercial and open-source databases to RDS.

Amazon Aurora

Another SQL-type database is Amazon Aurora, a fully compatible MySQL- or PostgreSQL-managed database as a service (DBaaS) solution. If you are currently using either MySQL or PostgreSQL on premises and are considering moving your database to the AWS cloud, Aurora is well worth evaluating. Amazon Aurora provides much faster performance than Amazon RDS MySQL deployments. Amazon Aurora has performance increases of up to five times the throughput of AWS RDS MySQL and three times the throughput of Amazon RDS PostgreSQL. Amazon Aurora achieves this performance by using an SSD virtual SAN cluster storage array that is replicated across three AZs maintaining six copies of data.

The following are features to know about the Amazon Aurora DB engine:

- **Backtracking:** Return the state of an Amazon Aurora cluster to a specific point in time within seconds without having to

restore data from a backup.

- **Amazon Aurora Global Database:** A single database consisting of a primary DB cluster in one region and up to five secondary DB clusters in different regions, enabling very low latency reads and recovery from regional outages.
- **Machine learning (ML):** Machine learning models are exposed as SQL functions, using standard SQL queries to build applications that call ML models.
- **Parallel queries:** Parallel queries can speed up queries while maintaining a very high transactional throughput.
- **Aurora Serverless:** Automatically start up, shut down, and scale capacity as required by your application.

When deploying Aurora (see [Figure 10-5](#)), there are four choices:

- **Aurora Provisioned:** This is the standard deployment, where the customer defines the database engine (MySQL or PostgreSQL), instance class, and the advanced details of placement, accessibility, encryption, backup, and failover required.
- **Aurora versions that support the parallel query feature:** A single query can be distributed across all the available CPUs in the storage layer to greatly speed up analytical

queries. More than 200 SQL functions, equijoins, and projections can run in parallel format.

Edition

Amazon Aurora with MySQL compatibility
 Amazon Aurora with PostgreSQL compatibility

Capacity type [Info](#)

Provisioned
You provision and manage the server instance sizes.

Serverless
You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.

► Replication features [Info](#)
Single-master replication is currently selected

Engine version [Info](#)

View the engine versions that support the following database features.

Show versions that support the global database feature
 Show versions that support the parallel query feature

Version

Aurora (MySQL 5.7) 2.09.0 ▼

To see more versions, modify the capacity types. [Info](#)

Figure 10-5 Aurora Deployment Options

- **Serverless:** This deployment option supports the MySQL- and PostgreSQL-compatible edition of Amazon Aurora. As mentioned, when a serverless Aurora database cluster is deployed, it operates, and scales based on the minimum and

maximum performance requirements that have been defined at creation. When your application enters a period of light or minimal activity, the associated Amazon Aurora database is scaled down to the minimum allocated size but remains online and continues to service application requests, scaling back out as required. When no activity is detected for a prescribed period of time, the Amazon Aurora database is paused, and you are not charged for the inactivity.

Rather than define the database instance class size for the Serverless option, you set the minimum and maximum capacity required. Behind the scenes, the database endpoint points to a fleet of resources that are automatically scaled based on your minimum and maximum requirements. Amazon Aurora Serverless scales up when monitoring reveals capacity constraints at any of the processing or database connections. For example, your serverless database cluster could be defined to scale up if the CPU utilization rises above 60%, or the connections are at more than 80% of the available connections. The Aurora cluster could also scale down if the application load is below 25% utilization and fewer than 30% of the connections are used. Consider a retail environment with multiple branch locations using a centralized point-of-sale system. As more customers enter the store and begin purchasing, the Amazon Aurora database scales up and down based on demand. After

hours, after a defined period of inactivity, the Aurora database is paused until the next day. When database connections are requested after an Amazon Aurora database has been paused, resuming the cluster operation will take a few seconds or more. To speed up operations, Amazon Aurora preloads the buffer pool with pages for known common queries stored in an in-memory page cache.

- **Global deployments:** Amazon Aurora deployments are for globally distributed Aurora deployments across multiple AWS regions. Amazon Aurora storage-based replication typically has latency of less than 1 second. In the case of a regional outage, a secondary region can be promoted to read/write capability in less than 1 minute. Typical use cases include financial, travel, or gaming applications that have strict uptime requirements. Up to 16 database instances can be deployed in each AWS region of a globally deployed Amazon Aurora database. Deploying Amazon Aurora as a global database has the following advantages:
 - **Global reads with local latency:** A global deployment keeps the main database updated in the primary AWS region. Users in other regions access the database records from their own secondary cluster hosted in their own AWS region.

- **Scalable secondary Amazon Aurora DB clusters:**
Secondary clusters can be scaled by adding more read-only instances to the secondary AWS region.

Amazon Aurora Storage

When comparing Amazon Aurora to a standard RDS deployment of MySQL, with Aurora deployments there is no need to provision storage in advance for future growth. The internal design of the Amazon Aurora storage engine allows for the automatic scaling of its distributed storage architecture from a starting point of 10 GB up to 64 TB, in 10-GB increments; this storage is spread across multiple SSDs across multiple AZs (see [Figure 10-6](#)). As previously mentioned, Amazon Aurora replicates six copies of its data records across three AZs, providing enhanced durability and greater than 99.99% availability.

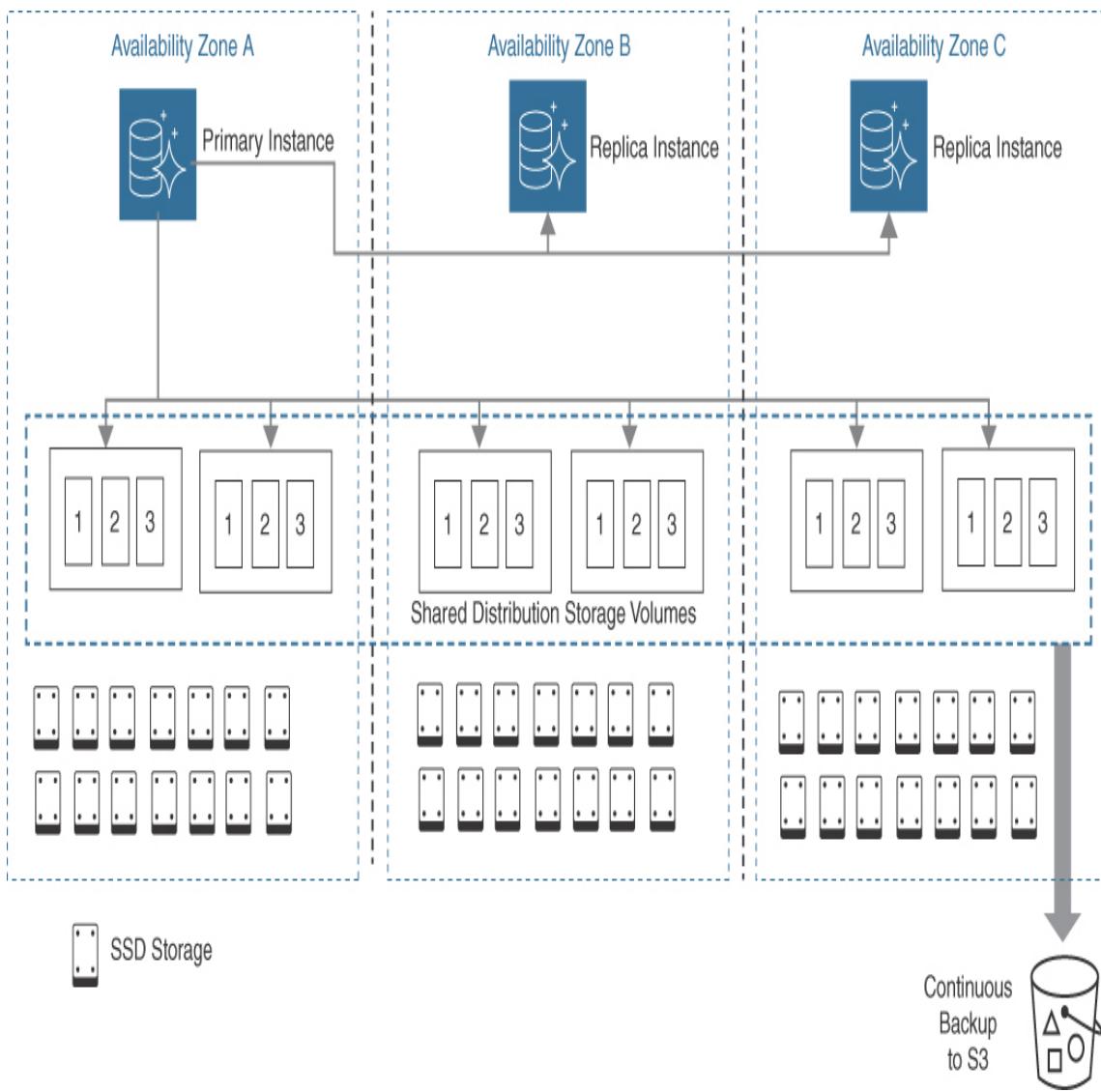


Figure 10-6 Aurora Data Storage Architecture

Amazon Aurora data is stored in a cluster volume, a single virtual volume supported by SSD drives.

Amazon Aurora data records are stored in a shared cluster storage volume of data stored on multiple SSDs across multiple

AZs. The cluster quorum deployed across the three AZs includes the six data nodes; the write set is four nodes, and the read set is the two remaining nodes spread across the three AZs. As a result, each AZ has a current copy of the database cluster data.

Aurora performance enhancements are due to the design of the storage plane, which is an SSD-backed virtualized storage array. Aurora has at a minimum a primary database instance DB, and can have up to 15 additional Amazon Aurora DB replicas. To boost the storage durability, the data transaction logs are continuously backed up to Amazon S3 storage.

Amazon Aurora is designed so that the underlying SSD storage nodes that make up Amazon Aurora's shared storage volume are deployed in a cell-like design spread across the three AZs, which helps to limit the size of the blast radius when failures occur. Reads only require three out of six nodes to be available; writes require four out of six nodes to be available. A failure of a single AZ results in four storage volumes still being available to carry out database operations.

Amazon Aurora's cluster design has the primary and replica database instances connecting to the same storage plane. The primary database DB instance carries out the write operations to the cluster volumes in each AZ and offloads the read

operations to the available replica database DB instances, as shown in [Figure 10-7](#). This design is different from standard RDS MySQL or PostgreSQL deployments, where standby database instances do not perform read requests.

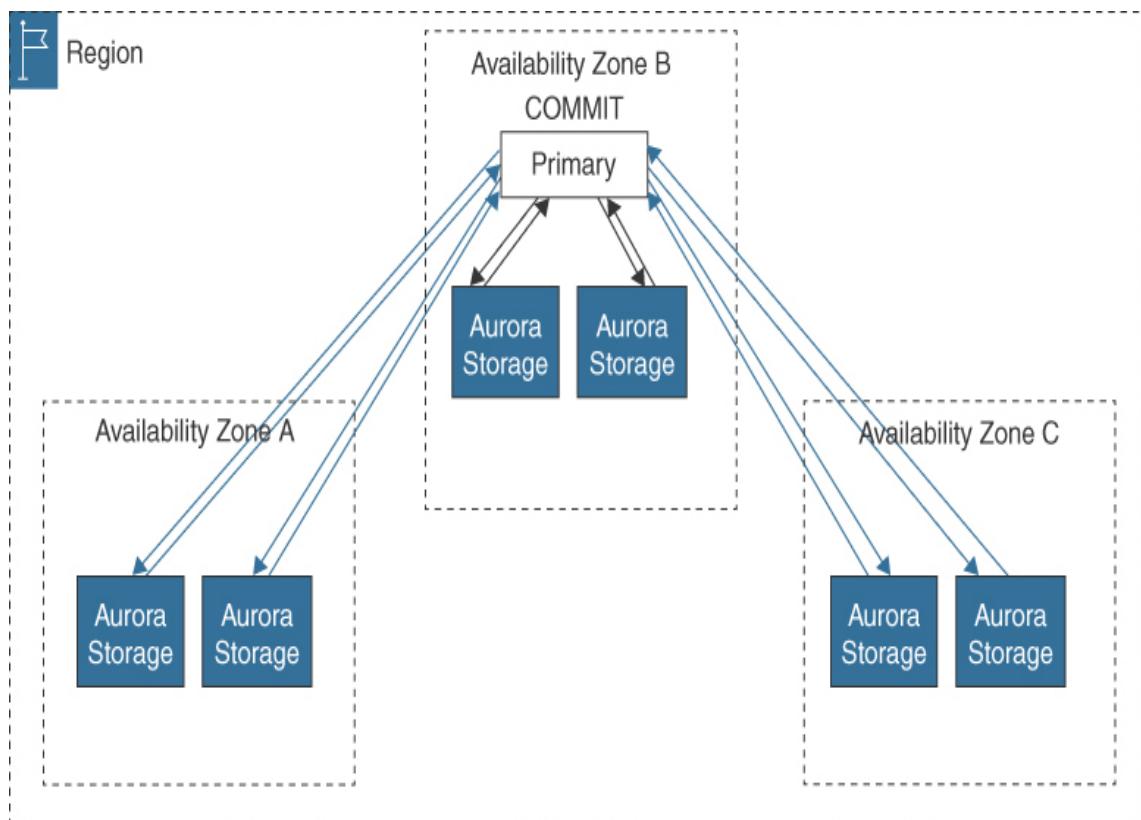


Figure 10-7 Aurora Cluster Design

The Amazon Aurora shared storage architecture results in data records being independent from the DB instances in the cluster; therefore, computation capacity is separated from the storage array. Adding an Amazon Aurora DB instance results in the new DB instance connecting to the shared storage volume that

already contains the existing data records. Removing an Amazon Aurora DB instance from the cluster does not remove any existing data from the cluster. Data records remain in the cluster until the entire cluster is deleted.

Amazon Aurora Replication

Each Amazon Aurora DB cluster has automatic built-in replication between the multiple DB instances located in the same cluster. Replication can also be set up with an Amazon Aurora cluster as either the source or the target of the replication.

As you create additional DB instances in an Aurora provisioned DB cluster, replication is automatically set up from the writer DB instance to all the other DB instances in the cluster. These additional DB instances are read-only and are defined as Amazon Aurora Replicas, as reader instances. Amazon Aurora Replicas are fully dedicated to performing read operations, write operations are managed by the primary instance to the cluster volume.

Amazon Aurora Replicas are used for queries that have been issued from the reader endpoint of the cluster, spreading the query load across the available Aurora Replicas in the cluster. If

a writer instance in a cluster becomes unavailable, one of the reader instances is automatically promoted to take its place as a writer instance. An Amazon Aurora DB cluster can contain up to 15 Amazon Aurora Replicas, which are distributed across the AZs within a single AWS region. An Auto Scaling policy can be created to automatically add or remove Amazon Aurora Replicas based on the following metrics: Average CPU Utilization or Average Connections of the current Amazon Aurora Replicas.

Note

To increase the availability of your Amazon Aurora DB cluster, it is recommended that you create at least one or more Amazon Aurora Replicas in two or more availability zones.

The data contained in the Aurora DB cluster volume had its own high-availability and reliability design completely independent from the DB instances in the cluster. The DB cluster volume is physically made up of multiple copies of the data for the cluster; the primary instance and the Amazon Aurora Replicas in the DB cluster see the data in the cluster volume as a single logical volume. As a result, queries have minimal replica lag, usually less than 100 ms after data records

have been updated. If the primary database instance becomes unavailable, there is automatic failover to an Amazon Aurora Replica. You can also define the failover priority for the available Amazon Aurora Replicas.

When failover occurs, the failover process usually takes seconds; the canonical name record (CNAME) is changed to point to the Amazon Aurora replica that has been promoted to be the new primary database.

With Amazon Aurora, you can lose access to two copies of data without affecting the writing process, and you can lose up to three copies of data without affecting the ability to read your data. In the background, Amazon Aurora constantly checks and rechecks the data blocks and discs, performing repairs when necessary, automatically using validated data records from the other volumes in the cluster. Amazon Aurora Replicas can also be created in different AWS regions. The first Amazon Aurora Replica created in a different region acts as the primary Amazon Aurora replica DB in the new region. You can also add Amazon Aurora Replicas in different AWS regions that will then share the same storage plane.

Note

Amazon Aurora can also operate with multiple read/write primary database instances deployed across multiple AZs, improving Aurora's high-availability design. If one of the primary database instances fails, other instances in the cluster can take over immediately, maintaining both read and write availability for the cluster.

Communicating with Amazon Aurora

Communication with an Aurora cluster is performed with specific endpoints, as shown in [Figure 10-8](#):

- **Cluster endpoint:** This endpoint to the Amazon Aurora database cluster is a single URL containing a host address and a port address to simplify connectivity to the primary database instance (M in [Figure 10-7](#)) for all writes, including insertions, updates, deletions, and changes. When failover occurs, the cluster endpoint automatically points to the new primary database instance.
- **Reader endpoint:** This endpoint connects to one of the available Aurora Replicas (R in [Figure 10-8](#)) for the database cluster; if there are multiple Amazon Aurora Replicas, the endpoint uses load balancing to support the read requests. If

your Amazon Aurora deployment is small, containing a single primary instance, the reader endpoint services all read requests from the primary database instance.

- **Instance endpoint:** This endpoint points to the current primary database instance (M in [Figure 10-8](#)) of the database cluster.

Key Topic

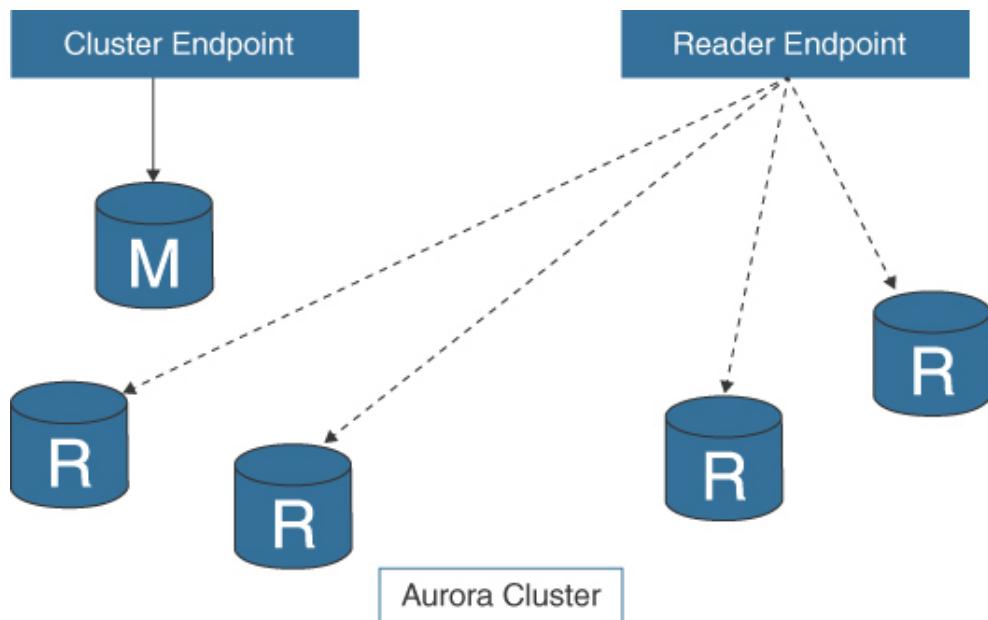


Figure 10-8 Aurora Endpoints

Endpoints can be used to map each connection desired to a specific DB instance. For write statements, directly connect to the primary instance DB. To perform queries, connect to the

reader endpoint. Amazon Aurora will automatically load balance the queries among the available Aurora Replicas. Custom endpoints can also be created to connect to a specific instance endpoint (for example, for diagnostics testing).

Note

Dow Jones used the AWS Data Migration service to migrate its legacy environment to Amazon Aurora (see <https://aws.amazon.com/solutions/case-studies/dow-jones/>). It uses a 1-TiB Amazon Aurora cluster that can handle 200 transactions per second. Alfresco used Aurora to scale to more than 1 billion documents with throughput of 3 million transactions per hour.

Amazon Aurora Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C02) exam, you need to understand the following critical aspects of Amazon Aurora:

- Each Amazon Aurora cluster has a set of compute nodes and a copy of the shared storage volume.
- The storage volume consists of six storage nodes located across three availability zones.
- Each database node in a cluster is also a writer node and can execute read and write statements.
- A database change carried out by a writer node is written to six storage nodes and three availability zones.
- Amazon Aurora scales storage up in 10-GB increments.
- Amazon Aurora can lose two copies of data without affecting database writes.
- Amazon Aurora can have up to 15 read replicas per region.
- Amazon Aurora Serverless is an on-demand auto-scaling Aurora deployment.
- Aurora Serverless scales up and down based on the database requirements.
- Automated backups are stored in S3 storage.
- Amazon Aurora does not support Local Zones.

Amazon DynamoDB

Another popular database service offered by AWS is Amazon DynamoDB. Amazon developed Amazon DynamoDB internally in 2006 and initially started using it to host the familiar shopping cart in the online Amazon store. Amazon DynamoDB

was publicly launched as a **NoSQL** database service in 2012, designed for Internet performance at scale for applications hosted at AWS. Today, the Amazon e-commerce store is mostly backed by Amazon DynamoDB and Amazon Aurora. Amazon DynamoDB is a fully managed NoSQL database service providing fast and predictable performance with horizontal scalability across availability zones and regions. Amazon DynamoDB tables can be created for storing and retrieving any amount of data with a very high number of requests. The following are the major Amazon DynamoDB features to know for the AWS Certified Solutions Architect – Associate (SAA-C03) exam:

- **Key-value and document data models:** These data models enable a flexible schema. Each row can have any number of columns at any point in time, allowing customers to easily adapt their table design when requirements change.
- **Amazon DynamoDB Accelerator (DAX):** In-memory cache provides fast read performance for Amazon DynamoDB tables, improving table performance from milliseconds to microseconds at millions of read requests per second.
- **Global tables:** Replicate Amazon DynamoDB table data automatically across multiple AWS regions scaling capacity to match workload requirements, providing single digit

millisecond read and write performance within each AWS region.

- **Supports streaming applications:** Capture item-level changes in an Amazon DynamoDB table as a Kinesis data stream. Kinesis Data Streams and Amazon DynamoDB can work together to store and process large amounts of streaming data from Amazon Kinesis in near-real time.
- **Read/write capacity modes:** On-demand capacity modes can manage capacity automatically, or provision capacity with automatic scaling of throughput and storage based on defined capacity limits.
- **Track item data with triggers:** Integrate with AWS Lambda functions with custom triggers when item-level changes are detected.
- **ACID transactions:** Amazon DynamoDB has native service-side support for transactions for multiple items within and across tables. (ACID is described in the section “[ACID and Amazon DynamoDB](#)” later in this chapter.)
- **Encryption at rest:** All data is encrypted at rest by default with encryption keys stored in the AWS Key Management service.
- **Point-in-time recovery (PITR):** Continual backups of Amazon DynamoDB table data to Amazon S3 allows

organizations to restore table data at any point in time, up to the second, during the preceding 35 days.

- **On-demand backup and restore:** Create full backups of Amazon DynamoDB tables for data archiving of any size.
-

Note

For older applications that are designed to use a SQL database, there may be no reason to make any database design changes. AWS has use cases for customers using SQL databases with millions of customers. For newer applications with no legacy concerns or requirements, a nonrelational database such as Amazon DynamoDB might be a consideration.

Amazon DynamoDB has been designed as a NoSQL database service that doesn't follow the same rules as a standard SQL database, as outlined in [Table 10-7](#).



Table 10-7 SQL and Amazon DynamoDB Comparison

Feature	SQL Server	Amazon DynamoDB
Database type	Relational database management system (RDBMS)	NoSQL database management system
Structure	Tables with rows and columns	Collection of JavaScript Object Notation (JSON) documents (key/value pairs)
Schema	Predefined	Dynamic
Scale	Vertical	Horizontal
Language	SQL structured	JavaScript

Feature	SQL Server	Amazon DynamoDB
Performance	Good for online analytical processing (OLAP)	Built for online transaction processing (OLTP) at scale
Optimization	Optimized for storage	Optimized for read/write
Query type	Real-time ad hoc queries	Simple queries

With an SQL database, there is a defined set of data rules, called the *schema*, which could be one or more interlinked tables, columns, data types, views, procedures, relationships, or primary keys. With SQL, the database rules are defined *before* any data is entered into the rows and columns of the relational databases table, according to the rules of **Structured Query Language (SQL)**.

In contrast, Amazon DynamoDB stores its data in tables but doesn't follow the same rules as a relational database. First, its data is stored in structured JSON key/value data values. There's more to Amazon DynamoDB than just a simple table, but before we get to those details, let's first think about databases and why we store data there. Databases keep our precious data safe, secure, and reliable. Relational databases have stored and secured our data reliably for years. And some relational databases at AWS can now automatically scale their compute performance and data storage; for example, Aurora Serverless can scale on demand, and all versions of Aurora scale data records automatically. At AWS, some of the infrastructure architecture designs between Amazon Aurora and Amazon DynamoDB are similar, even if the use cases are different.

Note

When Amazon looked at how its data operations on its internal Oracle databases were being carried out internally at AWS, it found the following:

- 70% of the queries were single SQL queries against the primary key of a single table with a single row of information being delivered.

- 20% of the queries were queries against multiple rows of a single table.
- 10% of the queries were complex relational queries.

This information helped Amazon realize that Amazon DynamoDB could completely replace its Oracle databases—a task that it mostly completed in 2019.

Amazon DynamoDB Tables

An Amazon DynamoDB table stores data as groups of attributes, also known as *items*. This concept is similar to the rows and columns found in other relational databases. Each item stored in an Amazon DynamoDB database can be stored and retrieved using a primary key that uniquely identifies each item in the table.

When you construct a table in Amazon DynamoDB, you must define a primary key. In [Figure 10-9](#), the primary key is `station_id`. A hash value is computed for the primary key, and the data in the table is divided into multiple partitions, each linked to the primary key hash for the table; in this case, it's

station_id. You can also choose to have a secondary index, such as LastName.

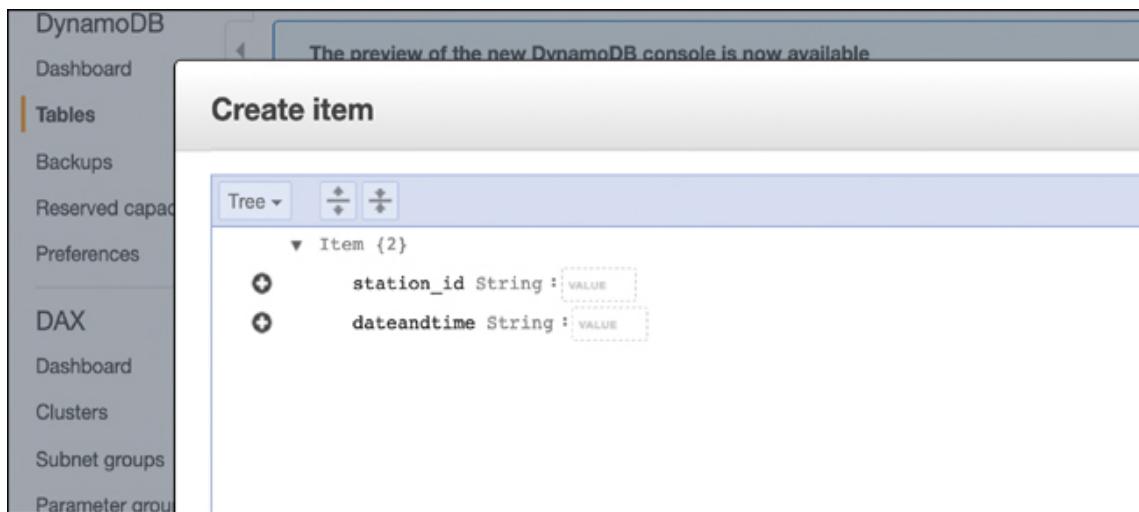


Figure 10-9 An Amazon DynamoDB Table

Provisioning Table Capacity

Amazon DynamoDB performance is defined in terms of **capacity unit** sizes:

- A single read capacity unit (RCU) means a strongly consistent read per second, or two eventually consistent reads per second for items up to 4 KB in size.
- A single write capacity unit (WCU) means a strongly consistent write per second for items up to 1 KB in size.

Amazon DynamoDB table design has a default level of read and write capacity units, as shown in [Figure 10-10](#). A design might

require only a defined amount of read and write performance because your tables could initially be small. The default provision capacity is five RCUs and five WCUs. However, over time, your design needs might change, and you might have to—or wish to—scale your table performance to a much higher level. With Amazon DynamoDB, you can make changes to the read and write capacity units for your table by switching from the default provisioned read/write capacity to on-demand and quickly adjusting the amount of scale that your application and, therefore, your Amazon DynamoDB table, require.

Key Topic

Read/write capacity

The read/write capacity mode controls how you are charged for read and write throughput and how you manage capacity.

Capacity mode

Provisioned

Table capacity

Read capacity auto scaling

On

Write capacity auto scaling

On

Provisioned read capacity units

1

Provisioned write capacity units

1

Provisioned range for reads

1 - 10

Provisioned range for writes

1 - 10

Target read capacity utilization

70%

Target write capacity utilization

70%

Figure 10-10 Adjusting Table Capacity

With Amazon DynamoDB, you can define both RCUs and WCUs for a table; the RCU value indicates how many reads you need per second for your table, and the WCU value indicates how many writes you need per second. A single read allows you to read up to 4 KB of data. If your object is under 4 KB, then a single read allows you to gather all the information; a 20-KB object would need 5 RCUs to perform the full read of the object. The same math applies to WCUs.

If you provision 300 RCUs for a table, Amazon DynamoDB splits up the reads across the three storage partitions. RCUs work on a system using the available tokens for the required read performance. Each token bucket has a *fill rate* that matches the defined RCUs. Say that the token bucket is refilled at the RCU rate of 100 tokens per second, ensuring that the table has enough tokens for the requested performance. Tokens are emptied from the token bucket at the rate of one token per read request. The number of tokens deducted from the bucket depends on the number of read requests and the size of the item read. The larger the item, the more tokens that are required to read the entire item.

When a read request is performed, if there are no tokens left in the token bucket, the read request is throttled. To get around this problem, the token bucket also has a burst token added to your bucket, which is calculated based on the rate of the number of provisioned RCUs multiplied by 300. This equals 5 minutes of additional performance at your defined RCU baseline; for spikes in read and write traffic to your table, you have up to 5 minutes of performance credits available to handle the increased load. When your Amazon DynamoDB table is not being read or written to, burst tokens are being added to your token bucket, up to a maximum of 30,000 tokens.

If you need to exceed read and write capacity throughput units higher than the maximum of 40,000, you can contact Amazon directly to request the desired unit increase.

Adaptive Capacity

To solve the problem of a table being throttled when it runs out of burst credits, Amazon DynamoDB has introduced a feature called *adaptive capacity* that increases the fill rate to the token bucket based on several parameters: the traffic to the table, the provisioned RCU capacity, the throttling rate, and the current multiplier. Adaptive capacity also provides additional burst tokens, so you have a longer period for bursting and don't run

out of tokens as quickly as you would if adaptive capacity were not enabled. There are still limits to how many burst credits you get, which is why Amazon DynamoDB introduced Auto scaling.

Auto scaling, as shown in [Figure 10-11](#), allows you to set lower and upper limits of performance capacity and a desired level of utilization. Amazon DynamoDB metrics for monitoring table performance and alarms are defined to alert Auto scaling when additional or less performance capacity is required for table reads and writes.



Table capacity

Read capacity

Auto scaling | [Info](#)
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

On
 Off

Minimum capacity units	Maximum capacity units	Target utilization (%)
1	10	70

Write capacity

Auto scaling | [Info](#)
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

On
 Off

Minimum capacity units	Maximum capacity units	Target utilization (%)
1	10	70

Figure 10-11 Amazon DynamoDB Auto Scaling Settings

Let's consider an example of a gaming company that uses Amazon DynamoDB as its database for millions of gamers. The information pieces that are being stored (such as game scores) are small, but potentially millions of scores need to be stored at scale. Data is stored in Amazon DynamoDB by first issuing a **PUT** request that is sent to a request router, which checks with the authentication services to see if the requested task is allowed. If everything checks out with security, the request is sent to the Amazon DynamoDB storage services, which determine where to first write the items to disk and then

replicate to the other storage nodes. Amazon DynamoDB employs hundreds of thousands of request routers and storage nodes, following a cell-based architecture design, to limit when failures occur by storing request routers and storage nodes in multiple partitions and AZs throughout the AWS region, as shown in [Figure 10-12](#).

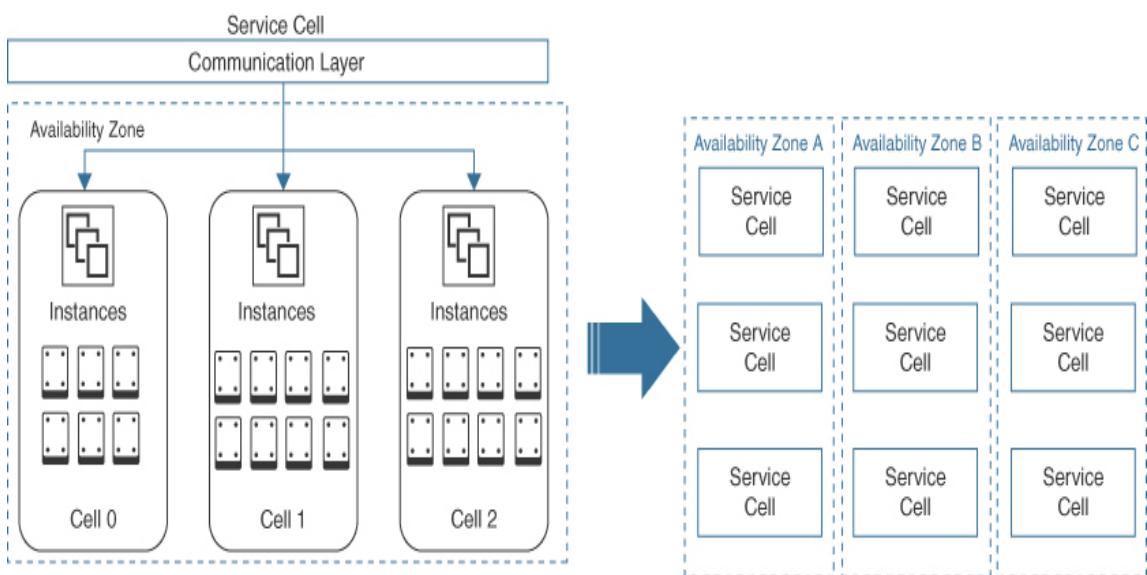


Figure 10-12 Amazon DynamoDB Cell-Based Design

Data Consistency

Because data is written into three partition locations across each AZ, data is not initially consistent in all storage partitions; however, after some time, all data locations across all AZs will be consistent. With Amazon DynamoDB, you have a choice of how consistent you want your data to be:

- **Strongly consistent:** If you want your data to be strongly consistent, a strongly consistent read produces a result from the storage nodes that performed a successful write of the information being requested.
- **Eventually consistent:** If you want your data to be eventually consistent, the leader node makes a random decision about which of the storage nodes that are hosting the partition to read from.

The odds are that you will get a consistent read even with eventual consistency because two storage partitions out of the three will always contain up-to-date data. Typically, the single storage node that is not consistent with the other two nodes is only milliseconds away from being up to date. One of the associated storage nodes will be assigned as the leader node—that is, the node that performs the first data write.

Once two of the associated storage nodes have acknowledged a successful write process, the leader storage node communicates with the request router that the write process is successful, and that router passes that information back to the application and the end user.

Each **PUT** request talks to the leader node first. Then the data is distributed across the AZs, as shown in [Figure 10-13](#). The leader

node is always up to date, as is one of the other storage nodes because there must be an acknowledgment that the **PUT** is successful in storage locations for a write process to be successful.

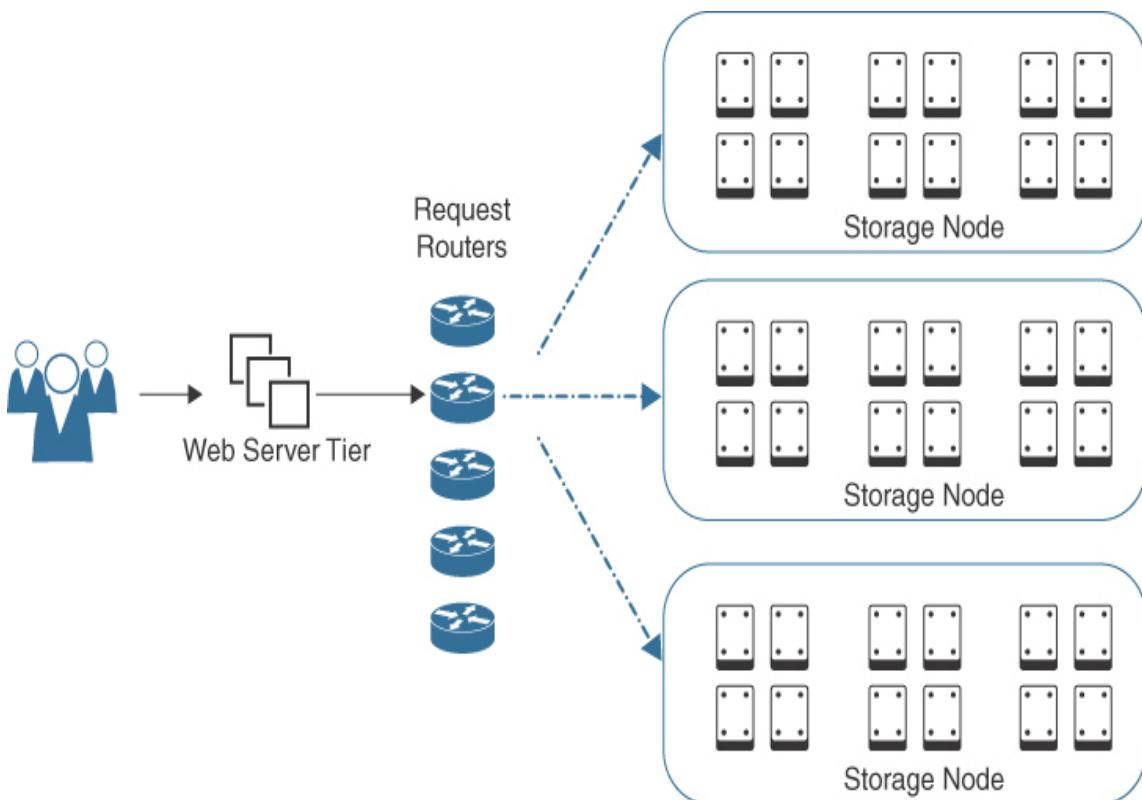


Figure 10-13 Amazon DynamoDB Storage Design

Paxos is the defined technical method to get the multiple storage systems to agree on a particular leader for the peer storage nodes. The leader storage node is always up to date. The leader and the peer storage nodes are also joined with a heartbeat that fires every 1.5 seconds with the associated

storage peers. If the peer storage nodes fall out of sync with the leader storage node, an election is performed, and one of the peer storage nodes becomes the new leader node.

The request routers are themselves stateless devices; any selected request router communicates with the leader node of the associated storage partition where your database is located.

As your Amazon DynamoDB database table scales in size, the internal design ensures predictable performance through a process called **burst capacity**. When partitions start to get overloaded, the partition is automatically split into multiple partitions so that the current read and write capacity units are spread across the available partitions to be able to better serve the required reads and writes of the Amazon DynamoDB table.

ACID and Amazon DynamoDB

Relational databases promise and deliver great reliability in the exact content of the data being stored. Relational database transactions achieve extremely high levels of storage consistency due to design principles such as ACID, which states that your database transactions have a high level of validity due to the properties of atomicity, consistency, isolation, and durability. The **ACID** standard has been adopted for years by

relational database engines such as Oracle, MySQL, PostgreSQL, and SQL Server. Transactions follow the ACID principles as a single process with four conditional variables:

- **Atomicity:** Each database transaction completes successfully, or it's not accepted.
- **Consistency:** Database transactions are successfully written to disk and validated.
- **Isolation:** Database transactions are isolated and secure during processing.
- **Durability:** Database transactions are committed to persistent storage and logged.

Amazon DynamoDB also supports ACID across tables hosted within a single or multiple AWS regions. Two internal Amazon DynamoDB operations handle these transactions:

- **TransactWriteItems:** A batch write operation with multiple **PUT**, **UPDATE**, and **DELETE** item operations that check for specific conditions that must be satisfied before updates are approved.
- **TransactGetItems:** A batch read operation with one or more **GET** item operations. If a **GET** item request collides with an active write transaction of the same item type, the read transactions are canceled.

With replicated Amazon DynamoDB data, the records must also be exact copies stored on the primary and standby database instances. The process of data replication *can* be fast, but updating replicated data records always takes some time, and the process of verification takes additional time.

Global Tables

An Amazon DynamoDB global table is multiple synchronized copies of a local Amazon DynamoDB table with the same data records replicated across multiple AWS regions, as shown in [Figure 10-14](#). Data is transferred from one AWS region to another using a synchronized replication engine in the source and destination AWS regions. AWS IAM service-linked roles ensure that the proper level of security is enforced when writing records to the global Amazon DynamoDB table partitions.

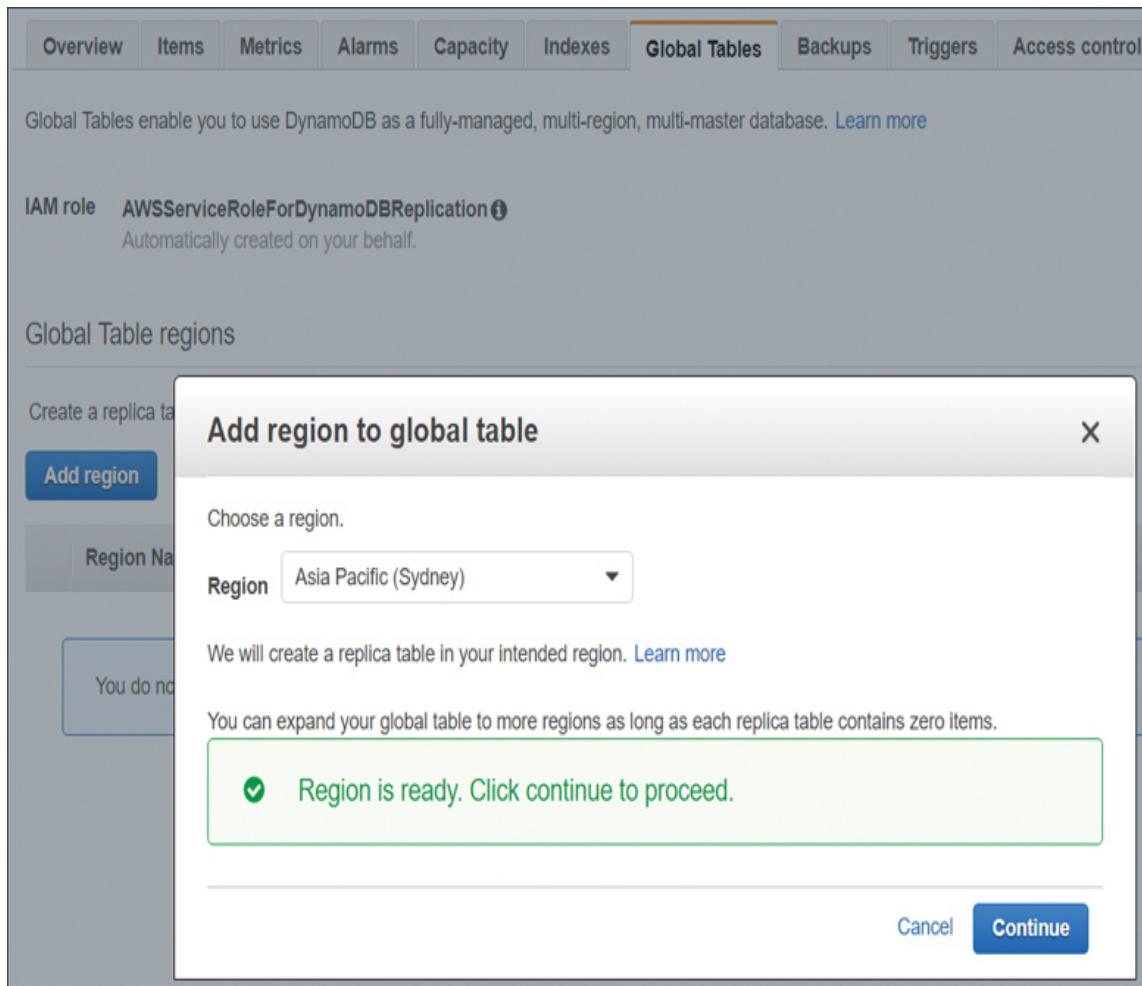


Figure 10-14 Amazon DynamoDB Global Tables

This can be useful for a variety of reasons, such as reducing latency for users in different regions, improving the availability of your data in the event of a region-wide outage, and enabling disaster recovery.

To set up global tables, first create a table in a primary region and then specify one or more secondary regions. Once the

global table is set up, DynamoDB automatically replicates updates made to the table in the primary region to the secondary regions.

First, the local replication engine compiles all the **PUT**, **UPDATE**, and **DELETE** items in the local copy of the primary Amazon DynamoDB table. Next, the local changes are replicated across the private AWS network to the secondary Amazon DynamoDB table in the destination AWS region. Global tables are replicated and updated in all region locations; updates first performed in one region are then updated in the other AWS regions. The outbound and inbound replication engines determine what updates are local, what updates need to be shipped outbound to the other linked copies of the Amazon DynamoDB table, and what updates need to be accepted inbound from other regions. Inbound replicated updates are compared using version numbers and millisecond timestamps to ensure ***data consistency*** is maintained across the Amazon DynamoDB global table, using a process called last-write conflict resolution; if the timestamps are the same, the local AWS region where changes were last initiated is proclaimed the winner. The data in the secondary regions is eventually consistent with the primary region, which means that it may take some time for the data to be fully replicated.

Amazon DynamoDB Accelerator

You can increase Amazon DynamoDB response times to eventually consistent data levels with microsecond latency by adding an in-memory cache to the design. E-commerce online sales, applications with read-intensive needs, and applications performing in-depth analysis over a long-term time frame are some of the use cases that can take advantage of Amazon DynamoDB Accelerator (DAX). Your DAX cluster, once provisioned, will be hosted in the VPC of your choice.

Applications can use the DAX cluster after the DAX client is installed on the EC2 instances hosting the associated application.

DAX can be designed to be highly available, with multiple DAX nodes hosted across multiple AZs within an AWS region, and can scale out up to ten replicas. Read operations that DAX responds to include **GetItem**, **BatchGetItem**, **Query**, and **Scan** API calls. Write operations are first written to the table and then to the DAX cluster. Write operations include **BatchWriteItem**, **UpdateItem**, **DeleteItem**, and **PutItem** API calls.

Backup and Restoration

Amazon DynamoDB provides several options for backing up and restoring table data, allowing organizations to protect their data from accidental deletion or corruption and recover from data loss or data corruption. There are two options for Amazon DynamoDB backup:

- **Point-in-time recovery (PITR) backup:** This option allows you to restore your Amazon DynamoDB table to any point in time up to 35 days. This is a fixed maximum value of retention and cannot be changed. A point-in-time restore point can be chosen up to 1 second before the current time. Once PITR has been enabled for Amazon DynamoDB, continuous backups are performed to controlled S3 storage.
- **On-demand backup:** This option allows you to create full backups of Amazon DynamoDB tables for long-term storage. An on-demand backup is created asynchronously, applying all changes that are made to a snapshot stored in S3 storage. Each on-demand backup backs up the entire Amazon DynamoDB table data each time.

A restored table, regardless of the backup type, includes local and global secondary indexes, encryption settings, and the provisioned read and write capacity of the source table at the time of restoration. After a table has been restored you must manually re-create any Auto scaling policies, IAM policies, tags,

and TTL settings that were previously applied to the backed-up table.

Amazon DynamoDB Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of Amazon DynamoDB:

- Amazon DynamoDB supports both key/value and document data models.
- Amazon DynamoDB Global tables replicate your data across AWS regions.
- Amazon DynamoDB automatically scales capacity to match your workload demands.
- Amazon Kinesis Data Streams can capture item-level changes in your Amazon DynamoDB table as a Kinesis data stream.
- Amazon DynamoDB has two capacity modes: On-demand and Provisioned.
- Amazon DynamoDB performs automatic scaling of throughput and storage.

- Amazon DynamoDB triggers integrate with AWS Lambda functions when item-level changes occur in an Amazon DynamoDB table.
- Amazon DynamoDB supports ACID transactions.
- Amazon DynamoDB encrypts data at rest by default.
- Amazon DynamoDB supports point-time recovery up to the second, up to 35 days.

Amazon ElastiCache

To improve the performance of existing applications and supported databases, you can deploy [**Amazon ElastiCache**](#), a fully managed in-memory caching service supporting Amazon ElastiCache for Redis, Amazon ElastiCache for Redis—Global Datastore, and Amazon ElastiCache for Memcached.

Amazon ElastiCache is designed to improve application performance by reducing the reads and writes to persistent storage and directing the traffic to an in-memory cache. Common uses include deploying ElastiCache as a read-only database replica or storage queue or as an in-memory read/write NoSQL database.

Amazon ElastiCache for Memcached

ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as either a cache or a data store:

- As a cache, ElastiCache for Memcached helps increase throughput and decrease access latency from RDS deployments or NoSQL databases such as Amazon DynamoDB.
- As a session store, ElastiCache for Memcached can be deployed using the Memcached hash table, which can be distributed across multiple nodes.

ElastiCache for Memcached use cases include application caching for database performance as an in-memory cache and session stores. ElastiCache for Memcached uses EC2 instances as nodes, and each node utilizes a fixed chunk of secure network-attached RAM running as an instance of Memcached deployment. ElastiCache for Memcached nodes are deployed in clusters, and each cluster is a collection of one single node or up to 40 nodes. For additional fault tolerance, place your Memcached nodes in select AZs across the AWS region. Features of ElastiCache for Memcached include the following:

- Automatic recovery from cache node failures

- Automatic discovery of nodes added or removed within a cluster
- Availability zone placement of nodes and clusters

One of the key features of ElastiCache for Memcached is the ability to deploy cache clusters across multiple AZs within a region (see [Figure 10-15](#)). Deploying a cache cluster across multiple AZs can improve the availability and durability of the cache. If one AZ becomes unavailable due to a failure or maintenance event, the cache cluster can continue to operate from the remaining AZs. This can help ensure that your application remains available and responsive even in the event of an infrastructure failure.

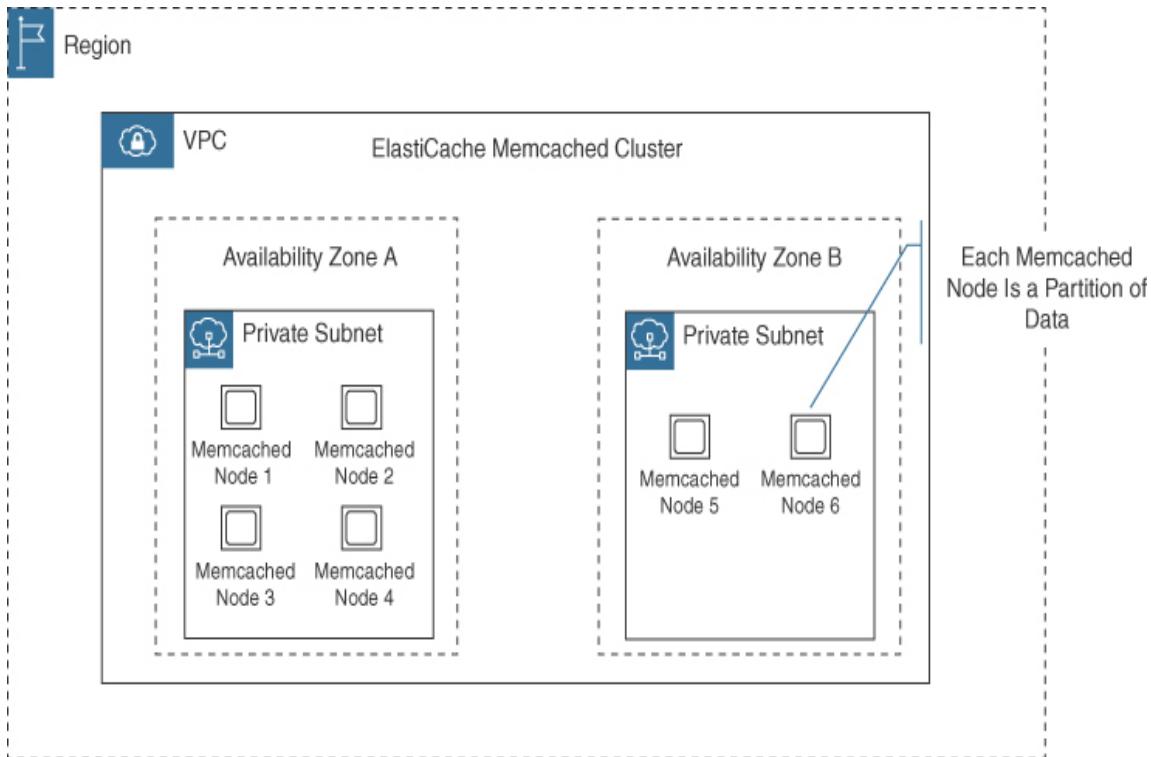


Figure 10-15 ElastiCache for Memcached Cluster

Amazon ElastiCache for Memcached Cheat Sheet

Key Topic

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of ElastiCache for Memcached caches:

- ElastiCache for Memcached does not provide persistent data storage.

- Each node represents a partition of data.
- ElastiCache for Memcached cannot be used as a data store.
- ElastiCache for Memcached scales out and in through the addition and removal of nodes.
- ElastiCache for Memcached can be deployed as a read replica for RDS and Amazon DynamoDB databases.
- ElastiCache for Memcached is useful for storing users' session state.
- ElastiCache for Memcached does not support multi-region failover or replication.
- Local Zones are supported for ElastiCache clusters.

Amazon ElastiCache for Redis

Amazon ElastiCache for Redis is a fully managed in-memory cache service that makes it easy to deploy and operate a distributed cache environment in the cloud. It is based on the popular open-source Redis cache engine and enables you to store and retrieve data from memory using the Redis data model.

ElastiCache for Redis is well suited for use cases that require fast data access and low latencies, such as real-time analytics, gaming, and social media. It can be used to cache frequently accessed data in memory, which can significantly improve the

performance of applications that rely on databases or other persistent storage systems.

ElastiCache for Redis would be a good choice for storing user state session state for a user session; the user session information needs to be stored, but not for the long term, as shown in [Figure 10-16](#). Rather than storing the user session information on the web instance that the user is connecting to, you store the user information in an in-memory cache; if the web instance fails, when the user is routed to another web instance, the user session information is still held in the memory cache and remains available for the duration of the user session.

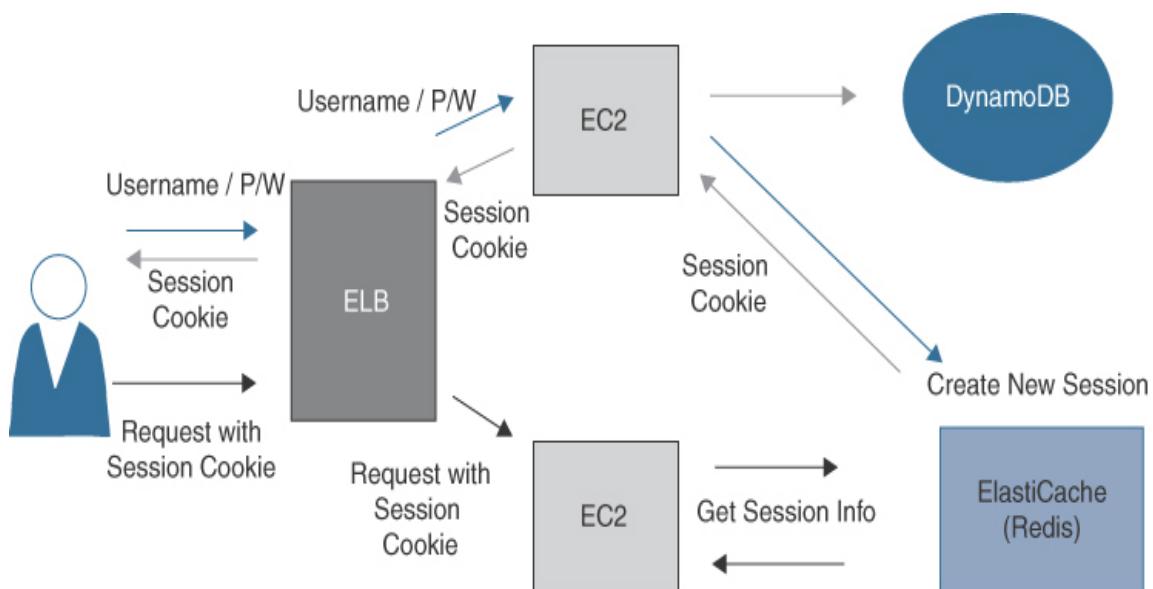


Figure 10-16 User State Information Stored in ElastiCache

Features of ElastiCache for Redis include the following:

- ElastiCache for Redis has automatic recovery from cache node failures.
- Multi-AZ deployment is supported for ElastiCache for Redis cluster nodes.
- ElastiCache for Redis cache data can be partitioned up to 500 shards.
- ElastiCache for Redis supports encryption in transit and encryption at rest, with authentication for HIPAA-compliant workloads.
- ElastiCache for Redis manages backups, software patching, failure detection, and recovery.

Amazon ElastiCache for Redis Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of ElastiCache for Redis caches:

- Redis is widely adopted as an in-memory data store for use as a database, cache, message broker, or queue.

- The ElastiCache for Redis data store is persistent.
- ElastiCache for Redis can be used as a data store.
- ElastiCache for Redis scales through the addition of shards, which is a grouping of one to six related nodes.
- Each multiple-node shard has one read–write primary node and one to five replica nodes.
- Nodes are charged on a pay-as-you-go basis or reserved nodes.
- ElastiCache for Redis supports automatic and manual backups to S3.
- Maximum backup retention limit is 35 days.
- ElastiCache for Redis supports automatic detection and recovery from cache node failures.
- ElastiCache for Redis autoscaling allows you to increase or decrease the desired shards or replicas automatically.
- Redis Version 3.2 and later supports encryption in transit and at rest for HIPAA-compliant applications.

ElastiCache for Redis: Global Datastore

ElastiCache for Redis provides a fast and secure cross-region replication designed for real-time applications such as media streaming, real-time analytics, and gaming operating with a global footprint across multiple AWS regions. The Global Datastore supports cross-region replication latency under 1

second between primary and secondary clusters. A multiple-region deployment of the Global Datastore provides geo-local reads closer for end users operating in each AWS region. The global data store consists of a primary active cluster which accepts writes that are then replicated to all secondary clusters within the defined Global Datastore, as shown in [Figure 10-17](#). The primary cluster also accepts read requests.

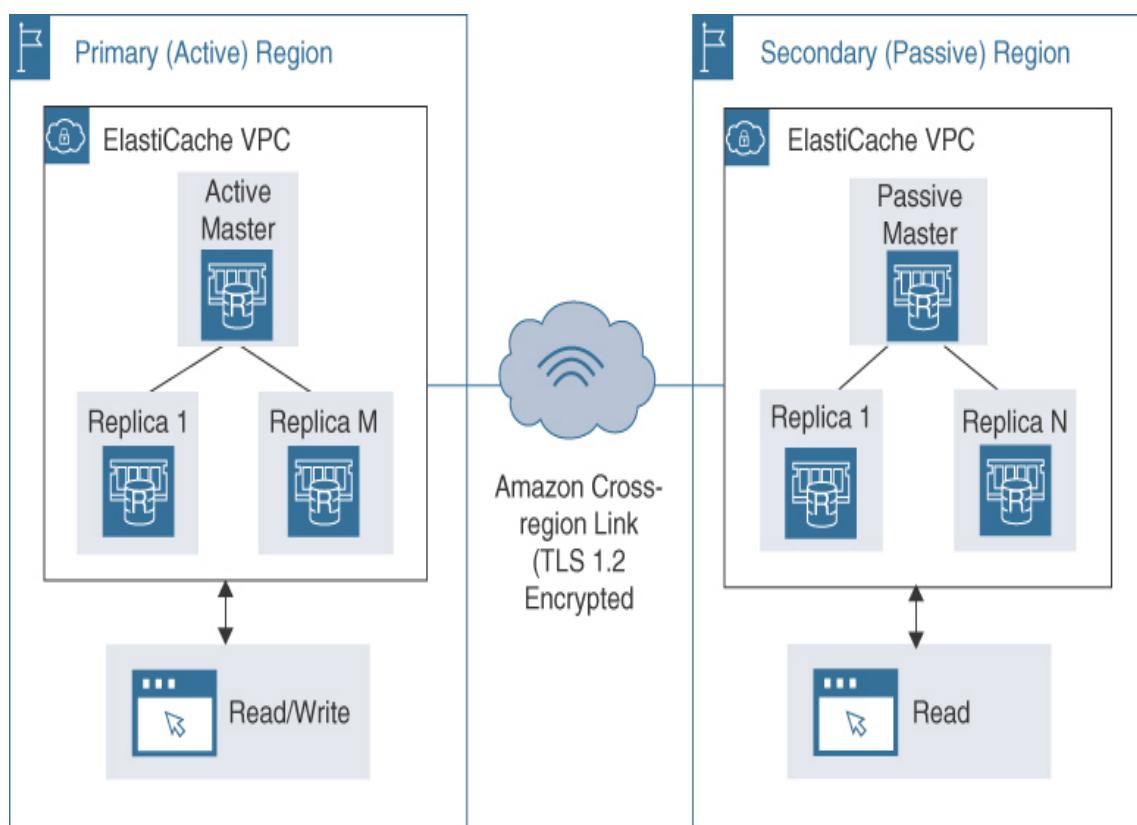


Figure 10-17 ElastiCache Global Data Store

A secondary cluster accepts only read requests and replicated data updates from the associated primary active cluster.

Secondary clusters must be located in a different AWS region than the primary cluster; data records are replicated from the primary active cluster to the secondary cluster using automatic asynchronous replication. Designs using remote replica clusters in other AWS regions with synchronized data records help reduce data latency by serving geo-local reads across each region.

Amazon Redshift

Amazon Redshift is a SQL-based data warehouse service that allows you to analyze your data by using standard SQL and business intelligence (BI) tools and standard Microsoft Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) connections. Redshift is designed as an online analytical processing (OLAP) database service that allows you to run complex analytical queries against petabytes of data.

An organization might use Redshift when you need to pull data sets together from many different sources, such as inventory, financial, and retail systems. In comparison, Amazon EMR is designed for the processing of extremely large data sets, such as for machine learning or streaming data, using data processing frameworks such as Spark or Hadoop.

Redshift uses *columnar data storage*, where data records are stored sequentially in columns instead of rows; this makes it ideal for data warehousing storage and analytics. This format of data storage allows a very high level of parallel processing across all data stores, resulting in enhanced query performance. Less storage space is required due to the high level of compression of the data stores.

Data storage and queries are distributed across all nodes, which are high-performance local disks attached to the supported EC2 instance nodes shown in [Figure 10-18](#). Each Redshift node is a minimum of 128 TB of managed storage across a two-node cluster. Depending on the instance size chosen, clusters range from 160 GB up to 5 PB. Choices for instances include the following options:

- **RA3 nodes:** Data is stored in a separate storage layer that can be scaled independently of compute. The data warehouse is sized based on the query performance required.
- **Dense Compute (DC):** High-performance requirements for less than 500 GB of data can utilize fast CPUs, large amounts of RAM, and SSD drives.
- **Dense Storage (DS2):** Create large data warehouses with a lower price point using HDDs with three-year-term reserved instances (RIs).

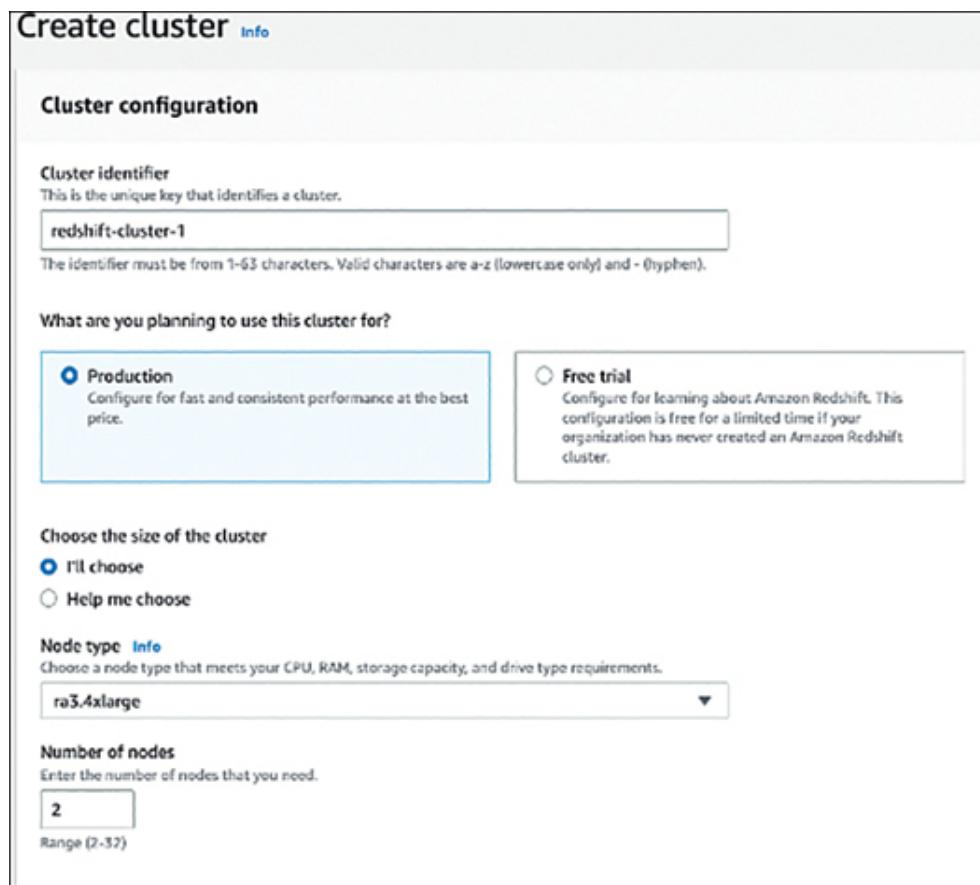


Figure 10-18 Creating a Redshift Cluster

The multimode design of a Redshift cluster includes both leader and compute nodes:

Key Topic

- **Leader nodes:** These nodes manage client connections and receive and coordinate the execution of queries. However, the queries themselves are performed by the compute nodes.

- **Compute nodes:** These nodes store all data records and perform all queries under the direction of the leader nodes. All compute work is performed in parallel, including queries, data ingestion, backups, and restores.

The size of a cluster can be automated using a feature called Concurrency Scaling, where Redshift adds additional cluster capacity as required to support an unlimited number of concurrent users and queries.

Redshift supports identity federation and SAML single sign-on, multifactor authentication, and additional security by hosting the Redshift cluster in an AWS VPC. Data encryption is supported using AWS Key Management Service (KMS).

To ensure data availability, Redshift replicates your data within your defined data warehouse cluster and continually backs up your data to Amazon S3 using snapshots. Redshift maintains three copies of your data:

- The original copy of data
- A replica copy that is stored on compute nodes in the cluster
- A backup copy that is stored in Amazon S3 and can be retained for 1 to 35 days

Redshift also supports SSL/TLS encryption in transit from the client application to the Redshift warehouse cluster.

Note

The Amazon Redshift Spectrum feature allows you to run SQL queries directly against exabytes of unstructured data stored in S3 data lakes.

Amazon Redshift Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of Amazon Redshift:

- Redshift ML can use SQL statements to train Amazon SageMaker models on data stored in Redshift.
- Advanced Query Accelerator (AQUA) allows Redshift to run up to ten times faster.
- RedShift Spectrum can be used to run queries against petabytes of stored Redshift data in S3.
- Redshift supports end-to-end encryption.

- Redshift can be hosted inside a VPC to isolate your data warehouse cluster in your own virtual network.
- Redshift can be integrated with AWS Lake Formation, which allows you to set up a secure data lake to store your data both in its original form and prepared for analysis.

Exam Preparation Tasks

As mentioned in the section “[How to Use This Book](#)” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. [Table 10-8](#) lists these key topics and the page number on which each is found.



Table 10-8 [Chapter 10](#) Key Topics

Key Topic Element	Description	Page Number
<u>Table 10-2</u>	Database Choices at AWS	481
<u>Figure 10-1</u>	Changing Database Instance Parameters	484
Section	High-Availability Design for RDS	485
<u>Table 10-4</u>	Initial Amazon RDS Setup Options	489
Section	Best Practices for RDS	491
Section	Amazon RDS Cheat Sheet	493
<u>Figure 10-6</u>	Aurora Data Storage Architecture	496
<u>Figure 10-8</u>	Aurora Endpoints	500

Key Topic Element	Description	Page Number
Section	Amazon Aurora Cheat Sheet	500
<u>Table 10-7</u>	SQL and Amazon DynamoDB Comparison	502
<u>Figure 10-10</u>	Adjusting Table Capacity	505
<u>Figure 10-11</u>	Amazon DynamoDB Auto Scaling Settings	506
Section	Amazon DynamoDB Cheat sheet	512
Section	Amazon ElastiCache for Memcached Cheat Sheet	514
Section	Amazon ElastiCache for Redis Cheat Sheet	516

Key Topic Element	Description	Page Number
List	Leader nodes and compute nodes	519
Section	Amazon Redshift Cheat Sheet	519

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

[primary database](#)

[standby database](#)

[read replica](#)

[endpoint](#)

[scale out](#)

[NoSQL](#)

Structured Query Language (SQL)

capacity units

burst capacity

ACID

data consistency

Amazon ElastiCache

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep Software Online.

- 1.** What is the advantage of using Amazon RDS to set up a database?
- 2.** What is the disadvantage of using Amazon RDS to set up a database?
- 3.** How can read replicas help improve database performance?

- 4.** What two options are available at AWS for hosted databases with global multi-region solutions?
- 5.** What is the difference between eventual consistency and strong consistency?
- 6.** How does Amazon Aurora have an advantage over a standard MySQL deployment with Amazon RDS?
- 7.** Where are continuous backups stored for all AWS database servers?
- 8.** What is an advantage of using Amazon ElastiCache for Redis to store user state?

Chapter 11

High-Performing and Scalable Networking Architecture

This chapter covers the following topics:

- [Amazon CloudFront](#)
- [AWS Global Accelerator](#)
- [Elastic Load Balancing Service](#)
- [AWS VPC Networking](#)
- [Subnets](#)
- [IP Address Types](#)
- [Connectivity Options](#)

This chapter covers content that's important to the following exam domain and task statement:

Domain 3: Design High-Performing Architectures

Task Statement 4: Determine high-performing and/or scalable network architectures

Domain 3 focuses on designing high-performing and scalable networking solutions for a workload. The network services that support hosted workloads that can adapt and scale include

Amazon CloudFront, Amazon’s content delivery network (CDN) service, designed to deliver high performance to end users across the globe in milliseconds. The AWS Global Accelerator provides improved application performance and availability using edge locations and the AWS global network. The Elastic Load Balancing (ELB) service also assists in delivering applications to end users with high availability and automatic scaling built in. Connectivity options for clients connecting to AWS include AWS Virtual Private Network connections, AWS Client VPN, or high-speed connections using AWS Direct Connect direct connections. Finally, all hosted workloads will reside on a logically isolated virtual private network (VPN); subnet options, IP addresses, and virtual private cloud (VPC) connectivity options are also covered in this chapter.

Note

Network connection options, Direct Connect, and AWS VPN connections are covered in [Chapter 4](#), “[Designing Secure Workloads and Applications](#),” which covers Domain 1, “[Design Secure Architectures](#),” Task Statement 2, “[Design secure workloads and applications](#).”

“Do I Know This Already?”

The “Do I Know This Already?” quiz enables you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you doubt your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 11-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A](#), “[Answers to the ‘Do I Know This Already?’ Quizzes and Q&A Sections.](#)”

Table 11-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Amazon CloudFront	1, 2
AWS Global Accelerator	3, 4
Elastic Load Balancing Service	5, 6
AWS VPC Networking	7, 8
Subnets	9, 10

Foundation Topics Section	Questions
IP Address Types	11, 12
Connectivity Options	13, 14

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment.

Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What is the purpose of deploying Amazon CloudFront?

1. To speed up video compression
2. To deliver shared data files to multiple servers
3. To cache data files close to end users
4. To secure data access

2. What special user makes content accessible only from CloudFront?

1. AWS IAM user
2. AWS account Root user
3. Origin Access Identity (OAI)
4. AWS IAM role

3. What network is utilized by AWS Global Accelerator to speed up application access?

1. The public Internet
2. AWS private network
3. AWS Direct Connect
4. AWS VPN connection

4. What kind of static IP addresses are assigned by AWS Global Accelerator?

1. Elastic IP addresses
2. Global IP addresses
3. Private IP addresses
4. IPv6 addresses

5. Which of the following ELB load balancer components determines what application traffic is accepted?

1. Target group

2. Listener

3. Health check

4. Access log

6. What is the standard methodology utilized by the EBS Application Load Balancer for delivering incoming traffic to registered instances?

1. Connection draining

2. SSL termination

3. Round robin

4. Sticky session

7. Once created, what does each AWS VPC span?

1. Subnets

2. Internet gateway

3. Virtual private gateway

4. Availability zones

8. Which type of networking is utilized by a VPC?

1. Layer 2

2. Layer 3

3. VLAN

4. MPLS

9. Where are the subnets hosted at AWS?

1. In availability zones
2. In regions
3. Across availability zones
4. Across regions

10. What type of IP address is automatically associated with every EC2 instance at creation?

1. Non-routable IP address
2. Elastic IP address
3. Public IP address
4. Private IP address

11. What type of IP address is not auto-assigned by default?

1. Bring-your-own IP address
2. Private IP address
3. Elastic IP address
4. Public IP address

12. Which of the following options can be used to connect VPCs?

1. With an Internet gateway

2. With route tables
3. With security groups
4. With peering connections

13. Which of the following does an endpoint provide?

1. Public connections to AWS services
2. Private connections to AWS services
3. Private connections to the AWS Marketplace
4. Public connections to the Internet

14. What does a gateway connection require to function?

1. A VPC
2. Route table entry
3. Subnet
4. EC2 instance

Foundation Topics

Amazon CloudFront

As previously mentioned, **Amazon CloudFront** is AWS's global CDN service. It is located at each edge location data center that optimizes the delivery of both static and dynamic web content, such as website images, videos, media files, and updates.

(Details on AWS edge locations can be found in [Chapter 4](#).)

When the *viewer* (which is the end user in CloudFront terminology) requests content that is served by CloudFront, the viewer's request is sent to the closest edge location with the lowest latency, ensuring the requested content is delivered to the viewer with the best performance possible.

How Amazon CloudFront Works

Amazon CloudFront delivers content using the following steps:

Step 1. A viewer (user) makes a request to a website or application configured with CloudFront.

Step 2. The DNS service (Route 53) routes the viewer's request to the CloudFront edge location closest to the viewer.

Step 3. If the requested content is already in the edge location cache, it is delivered quickly to the viewer.

Step 4. If the requested content is not in the regional edge cache or edge location cache, CloudFront requests the content from the origin location and delivers the request.

Step 5. Copies of the delivered content are stored in multiple edge locations, providing redundancy and availability (see [Figure 11-1](#)). Persistent connections to each origin service

location are kept open by CloudFront to fetch requested objects from the origin locations as quickly as possible. Deploying a CloudFront distribution also provides increased resiliency for your applications, as multiple edge locations are available for accessing the requested content. In addition, Amazon Route 53 records are stored redundantly in each region to ensure the reliability of the AWS global DNS service.



Figure 11-1 CloudFront Operation

Regional Edge Caches

A ***regional edge cache*** is an additional caching location within select AWS regions with a large amount of additional cache resources to ensure that more objects remain cached. A

regional edge cache is located between the origin (Amazon S3 bucket or web server) and the edge location and helps speed up access to frequently accessed content by keeping cached content as close as possible to the end user. A simple request for content that is available is served by the regional edge cache; RESTful API methods such as **PUT**, **POST**, and **DELETE** are sent directly to the edge location and do not proxy through regional edge cache locations.

Requests served from the regional edge cache don't go back to the edge cache or to the origin location (see [Figure 11-2](#)).

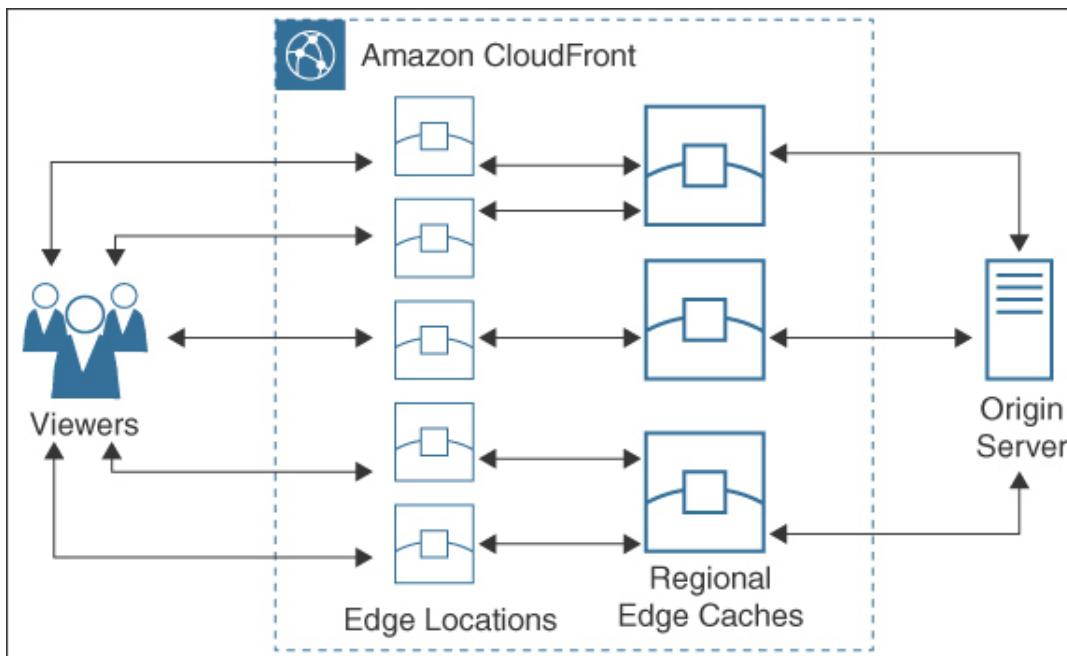


Figure 11-2 Regional Edge Location Placement

CloudFront Use Cases

**Key
Topic**

You should be aware of the following use cases for the AWS Certified Solutions Architect – Associate (SAA-C03) exam regarding CloudFront distributions:

- **Speeding up static website content delivery:** This is the number-one reason for deploying a CloudFront distribution. Static content includes images, videos, CSS style sheets, and JavaScript files.
- **Providing video-on-demand or live streaming video:** Video-on-demand options can stream formats such as MPEG DASH, Apple HLS, Microsoft Smooth Streaming, and CMAF to any network-enabled device. Live streaming supports the caching of media fragments at the edge location; the proper order to stream the media fragments is documented in the associated manifest file.
- **Encrypting content:** You can add field-level encryption to protect specific data fields against tampering during system processing. This ensures that only select applications can view the encrypted data fields.
- **Customized requests:** Using CloudFront functions or Lambda@Edge functions allows for customization of both ingress and egress requests with custom functions. Details on

CloudFront Functions and Lambda@Edge functions are provided later in this chapter. Details on AWS Lambda are provided in [Chapter 9, “Designing High-Performing and Elastic Compute Solutions.”](#)

HTTPS Access

CloudFront can be configured to require the use of the HTTPS protocol to request content, ensuring that all connections remain encrypted. CloudFront is configured from the properties of each distribution. Selecting HTTPS ensures that communication remains encrypted for both ingress and egress data transfer. The steps taken in the HTTPS communication process are as follows:

Step 1. A request for content using HTTPS is submitted to CloudFront in an encrypted format.

Step 2. If the object is present in the regional edge cache (if present) or the CloudFront edge cache, it is encrypted and returned to the viewer. If the object is not in either of the cache locations, CloudFront communicates with the origin via SSL/TLS, receiving the requested content from the origin location and sending the encrypted content to the viewer.

Step 3. The object is saved in the edge cache and, if present, in the regional edge cache.

Serving Private Content



There are two methods available for securing the distribution of CloudFront private content to select viewers:

- Use signed URLs or signed cookies. You can create a signed URL or signed cookie that grants temporary access to a private file.
- Use an origin access identifier (OAI) to grant CloudFront permission to access your Amazon S3 bucket or custom origin and serve your private content.

Using Signed URLs

Using signed URLs and/or signed cookies helps you distribute private content across the Internet to a select pool of viewers. When you create a signed URL or cookie, the content is signed using the private key from the associated public/private key pair. When access to the content is requested, CloudFront compares the signed and unsigned portions of the signed URL

or cookie. If the public/private keys match, the content is served; if the keys don't match, the content is not served. To use signed URLs with CloudFront, you must set up a trusted signer, which is an AWS account or an IAM user in your AWS account that has permission to create signed URLs and signed cookies. You also must configure your CloudFront distribution to use signed URLs as an additional layer of security. When creating signed URLs and/or signed cookies, conditions for accessing the URL are dictated by a JSON policy statement like the example shown in [Example 11-1](#), which mandates which of the following restrictions are to be enforced:

- The date and time after which the URL is accessible
- The date and time after which the URL is no longer accessible
- The IP address range of devices that can access content

CloudFront checks the expiration date and time for signed URLs at the time of the viewer request.

Example 11-1 Accessing a File from a Range of Trusted IP Addresses

[Click here to view code image](#)

```
{  
    "Statement": [  
        {  
            "Resource": "http://*",  
            "Condition": {  
                "IpAddress": {  
                    "AWS:SourceIp": "192.0.4.0/32"  
                },  
                "DateGreaterThan": {  
                    "AWS:EpochTime": 1367034400  
                },  
                "DateLessThan": {  
                    "AWS:EpochTime": 1367120800  
                }  
            }  
        ]  
    }  
}
```

Using an Origin Access Identifier



If the CloudFront origin is an S3 bucket, direct access to the S3 bucket can be restricted using a special CloudFront user called an **origin access identity (OAI)** that is associated with your CloudFront distribution, as shown in [Figure 11-3](#). Configuring S3 bucket permissions allows the OAI to access the requested objects from the S3 bucket for CloudFront serving the objects to the viewer. The OAI is a special AWS Identity and Access Management (IAM) user associated with your CloudFront distribution. Once the OAI is created, only the OAI user can directly access objects in the S3 bucket origin; permissions need to be configured to allow only the OAI to access the bucket.

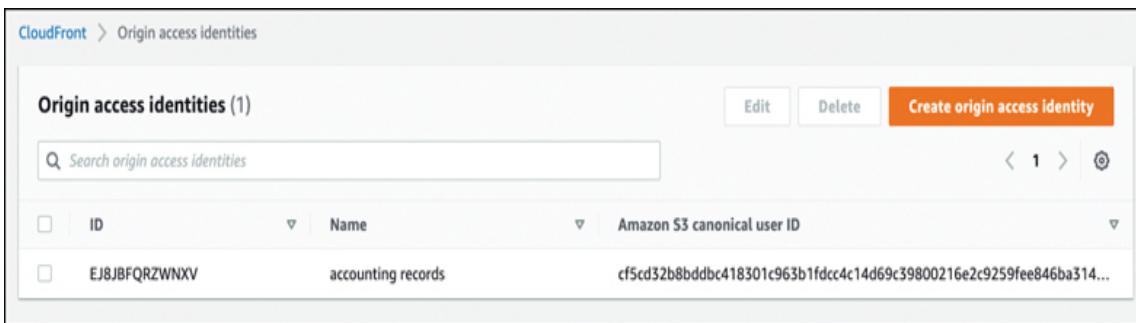


Figure 11-3 Ordering an Origin Access Identity

Restricting Distribution of Content

When an end user requests content from CloudFront, the content is served regardless of the physical location of the end user. To allow users only from approved countries to access

cached content, geo-restrictions can be enabled by defining CloudFront access lists, as shown in [Figure 11-4](#).

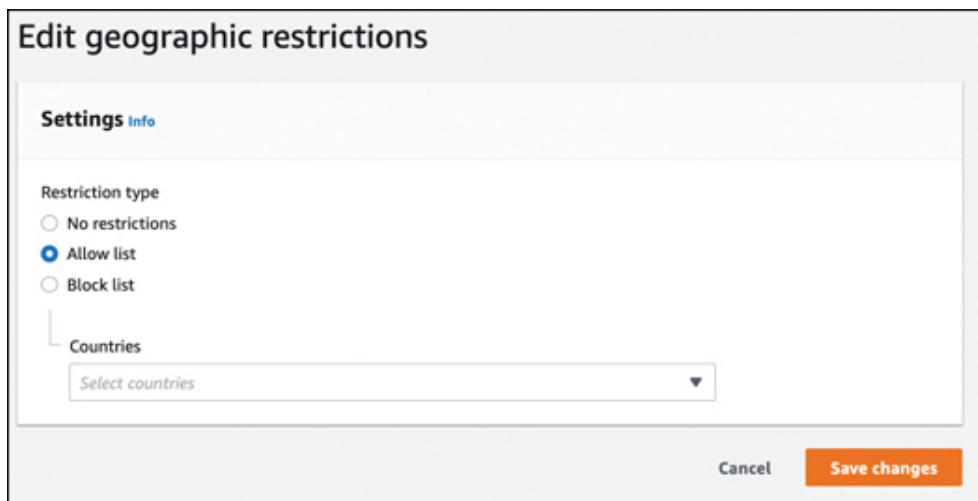


Figure 11-4 CloudFront Geographic Restrictions

CloudFront Origin Failover

Key Topic

CloudFront also has an additional option to assist with data reliability and resiliency called [***origin failover***](#).

To set up origin failover, create a CloudFront distribution with at least two origins in place and define cache behavior to use the primary origin group for content requests, as shown in [Figure 11-5](#). Next, in the CloudFront distribution origin group,

define the HTTP status codes to be used as failover criteria to the secondary origin; for example, 500, 502, 503, or 504 codes.

Create origin group

Settings

Origins
Choose the origins for this group, then put them in priority order.

1: S3-313858614000-awsmacietrail-dataevent (primary)

Name
Enter a name for this origin group.

Failover criteria
Select the origin errors to use as failover criteria.

400 Bad Request
 403 Forbidden
 404 Not found
 416 Range Not Satisfiable
 500 Internal server error
 502 Bad gateway
 503 Service unavailable
 504 Gateway timeout

Figure 11-5 Origin Failover Setup

With origin failover enabled, CloudFront operates normally and relies on the primary origin. When one of the specified HTTP

status codes is received, failover to the secondary origin occurs if present. The speed of failover can be controlled by adjusting the Origin Connection Timeout and the Origin Connection Attempts default values for the respective CloudFront distribution.

Video-on-Demand and Live Streaming Support

CloudFront can deliver video on demand (VOD) or live streaming video from any HTTP origin. Video content must be packaged together with a supported encoder (MPEG DASH, Apple HLS, CMAF) before CloudFront can distribute the streaming content. CloudFront and AWS Media Services can be used together to deliver live streaming video.

- **Video on demand:** Content is stored on a server and can be watched at any time. Content can be formatted and packaged using AWS Elemental MediaConvert. After content is packaged, it can be stored in Amazon S3 and delivered upon request using CloudFront.
- **Live streaming video:** AWS Elemental MediaConvert can be used to compress and format the live streaming video delivered by CloudFront to end users.

Edge Functions

Key Topic

Serverless custom functions called *edge functions* can be written to customize how a CloudFront distribution processes HTTP viewer requests and responses. Edge functions can be written using CloudFront functions and Lambda@Edge functions.

CloudFront Functions

JavaScript can be used to create what are called “lightweight” functions to monitor viewer requests and responses for customizations. CloudFront Functions must finish executing within sub-milliseconds. Use cases include

- **Modifying the HTTP request from the viewer:** Return the modified request to CloudFront for processing. Headers, query strings, and URL paths can be modified.
- **Header manipulation:** Insert, modify, or delete HTTP headers for the viewer request or response.
- **URL redirects:** Redirect viewers to other pages based on information contained in the request, as shown in [Figure 11-6](#).

The screenshot shows the AWS CloudFront Functions console. At the top, the navigation path is "CloudFront > Functions > Change_content-colour". The main title is "Change_content-colour". On the right, there are "Edit" and "Delete" buttons. Below the title, under the "Details" section, there are two rows of information: "Name" (Change_content-colour) and "Last modified" (June 3, 2022 at 7:31:53 PM UTC), and "Description" (empty) and "ARN" (arn:aws:cloudfront::313858614000:function/Change_content-colour). Below the details, there are three tabs: "Build" (selected), "Test", and "Publish". Under the "Function code" section, there are two tabs: "Development" (selected) and "Live". The code editor shows the following JavaScript code:

```
1 function handler(event) {
2     // NOTE: This example function is for a viewer request event trigger.
3     // Choose viewer request for event trigger when you associate this function with a distribution.
4     var response = {
5         statusCode: 200,
```

Figure 11-6 CloudFront Functions

Lambda@Edge Functions

Lambda@Edge is a managed AWS service that allows you to craft custom functions to carry out any task written in a variety of programming languages, including Python, Go, C#, Node.js, or Java. Lambda@Edge sits in the middle of the ingress and egress communication paths. Lambda@Edge could send specific

content to users sending requests from a smartphone and send different specific content to users sending requests from a traditional computer. Lambda functions can be executed when the following requests occur:

- When CloudFront receives a request for content from a viewer
- When CloudFront forwards a request to the origin server (S3 bucket or web server)
- When CloudFront receives a response from the origin server (S3 bucket or web server)
- When CloudFront sends back a response to the viewer

Lambda@Edge Use Cases

AWS Certified Solutions Architect – Associate (SAA-C03) exam questions are based on scenarios, and some of the scenarios expect that you know the use cases for the services being considered by the question's solution. Lambda@Edge use cases include the following:

- You could return different objects to viewers based on the devices they're using. In this case, the Lambda@Edge function could read the User-Agent header, which provides information about a viewer's device.

- Perhaps you're selling clothing in different sizes. You could use cookies to indicate which size the end user selected when looking at clothing choices. The Lambda@Edge function could show the image of the clothing in the selected color and size.
- A Lambda@Edge function could inspect and confirm the validity of authorization tokens to help control access to your content.
- A Lambda@Edge function could be used to confirm viewer credentials to external sources.

CloudFront Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of CloudFront:

- Control access to your public-facing content by mandating access via HTTPS endpoints using TLS 1.3.
- Origins include S3 buckets, AWS Elemental MediaStore container, an Application Load Balancer, a Lambda function URL, or a custom origin web server.

- Securing content access by using signed URLs and cookies.
- Use origin access identity (OAI) to restrict direct access to S3 bucket access, making it only accessible from CloudFront.
- Origin failover automatically serves content from the secondary origin when the primary origin is not available.
- Lambda@Edge functions support customizations that take from milliseconds to seconds to execute.
- CloudFront functions are lightweight functions that take less than one millisecond to execute.

AWS Global Accelerator

The Amazon Global Accelerator service routes traffic over the AWS private network to the closest available edge location endpoint that is closest to the end user. End user traffic enters the closest edge location and the Global Accelerator routes traffic to the closest application endpoint. The application outbound traffic returns over the AW private network back to the end user using the optimal endpoint (edge location).

Application endpoints can be created in single or multiple AWS regions. Global Accelerator uses accelerators to improve the performance of applications for local and global users.

The Global Accelerator uses listeners to process inbound connection requests from end users based on the TCP port or

port range specified for a single or multiple listeners. Each listener has one or more endpoint groups associated with it. Endpoint groups use endpoints in the defined AWS region (see [Figure 11-7](#)), and traffic is forwarded to the available endpoints in one of the groups. Both listener and endpoint ports can also be remapped to custom ports using port overrides.

Listener: 443 TCP

Each listener can have multiple endpoint groups. Each endpoint group can only include endpoints that are in one Region. You aren't required to add an endpoint group, but until you do, traffic to this listener won't reach any endpoints.

Region Info

us-east-1

Traffic dial Info

100

[Remove](#)

▶ [Configure port overrides](#)

▶ [Configure health checks](#)

[Endpoint group Region](#)

100

[Remove](#)

A number from 0 to 100.

▶ [Configure port overrides](#)

▶ [Configure health checks](#)

[Add endpoint group](#)

Listener: 80 TCP

Each listener can have multiple endpoint groups. Each endpoint group can only include endpoints that are in one Region. You aren't required to add an endpoint group, but until you do, traffic to this listener won't reach any endpoints.

Region Info

us-east-1

Traffic dial Info

100

[Remove](#)

A number from 0 to 100.

▶ [Configure port overrides](#)

▶ [Configure health checks](#)

[Add endpoint group](#)

Figure 11-7 Adding Listeners to Accelerator

Two types of accelerators can be deployed:

Key Topic

- **Standard accelerator:** A standard accelerator automatically routes traffic to the optimal AWS region with the lowest latency, using static Anycast IP addresses that are globally unique and do not change. For a standard accelerator with IPv4 addresses, endpoints can be Network Load Balancers, Application Load Balancers, EC2 instances, or Elastic IP addresses. With dual-stack addresses (IPv4/IPv6), only Application Load Balancer endpoints that have been configured to support dual-stack are supported. The use case for a standard accelerator is for applications that require a consistent, low-latency connection, such as web applications, mobile applications, and gaming applications. Each listener created in a standard accelerator can include one or more endpoint groups; each listener has a Traffic dial (see [Figure 11-7](#)) that can be used to increase or decrease traffic to each endpoint in the selected AWS region.
- **Custom accelerator:** A custom routing accelerator maps listener port ranges to a specific Amazon EC2 private IP address and port destination in an AWS VPC and subnet. A custom accelerator can logically map one or more end users to a specific destination, such as a gaming application with

multiple players or a training application that needs to assign multiple end users to a specific server for video training sessions. A custom accelerator is mapped to an AWS VPC endpoint with a destination port range that maps the incoming client connections.

The AWS Global Accelerator Speed Comparison Tool can also be used to review Global Accelerator download speeds compared to Internet downloads across AWS regions.

There are several additional use cases to know for the AWS Certified Solutions Architect – Associate (SAA-C03) exam:

- **Single-region applications:** End users' traffic is sent over 90 global edge locations onto Amazon's private network and sent to your application origin.
- **Multi-region applications:** Static IPs can be mapped to multiple application endpoints across AWS regions, as shown in [Figure 11-8](#).
- **Multi-region storage:** S3 Multi-Region Access Points rely on the AWS Global Accelerator for accessing data sets stored in S3 buckets across multiple AWS regions.

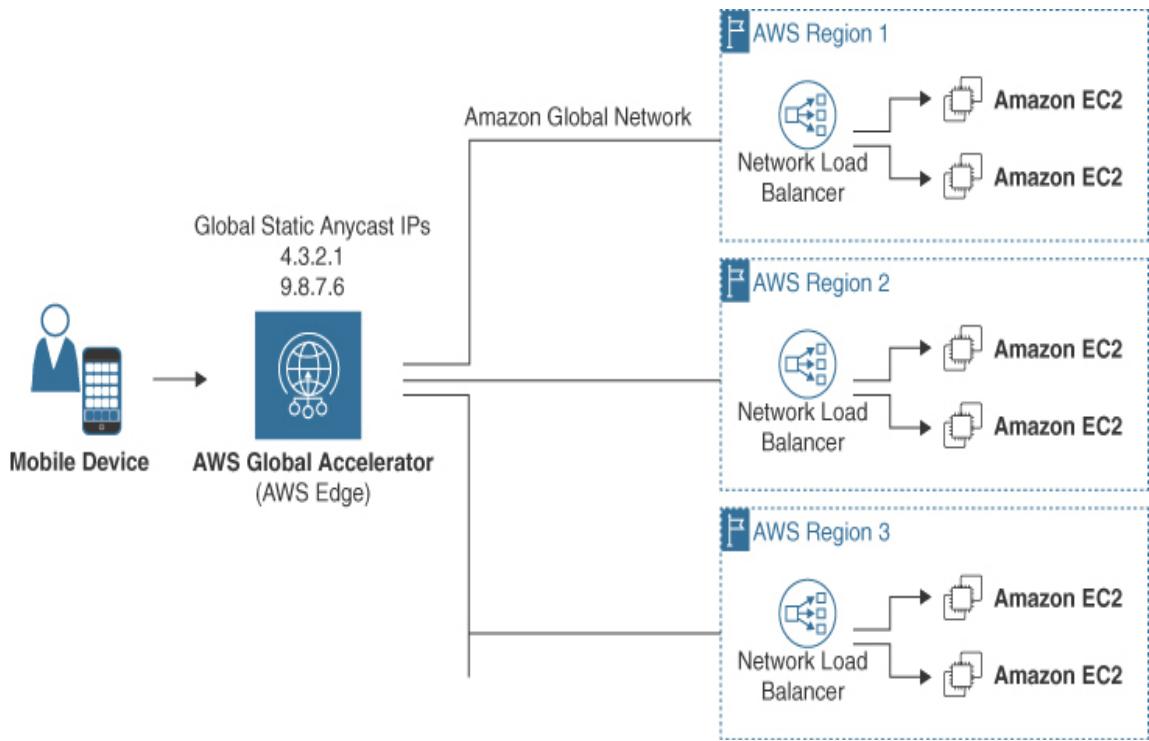


Figure 11-8 Multi-Region Global Accelerator Operation

Elastic Load Balancing Service

Amazon Elastic Load Balancer (Amazon ELB) is a load balancing service that distributes incoming application traffic across Amazon EC2 instances, Amazon ECS containers, AWS Lambda functions, and IP addresses.

ELB helps to ensure that your application is highly available and scalable by distributing incoming traffic across multiple resources. It can also help to improve the performance of your application by evenly distributing traffic across your resources and automatically scaling them to meet demand. The Elastic

Load Balancing Service (ELB) provides the Application Load Balancer for HTTP/HTTPS workloads and the Network Load Balancer for TCP/UDP workloads. The Gateway Load Balancer can deploy and manage third-party load balancer virtual appliances such as Nginx, Cisco, and Broadcom (see [Figure 11-9](#)).

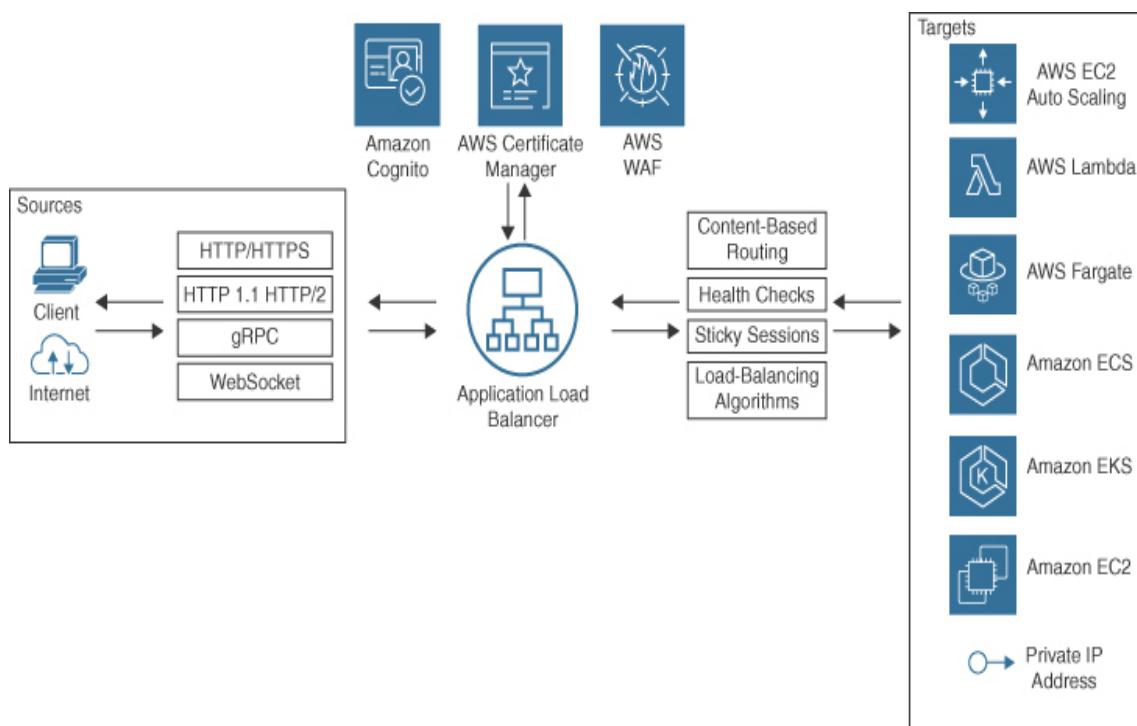


Figure 11-9 Application Load Balancer Targets

Each load balancer ordered is assigned a traffic profile with a prescribed amount of throughput capacity. The ELB service monitors the performance and requirements of each online

load balancer and scales the performance and capacity required based on the incoming user requests.

For each load balancer ordered, you are charged a small monthly fee, plus data transfer charges based on the number of Load Balancer Capacity Units (LCUs) used every hour. The LCU is the hourly aggregate total of incoming traffic requests based on new and active connections, consumed bandwidth, and the number of ***listener*** rules evaluated.

Application Load Balancer Features

Application load balancers have features used to manage and route incoming public traffic from the Internet:

- **SSL/TLS traffic decryption:** Application load balancers deploy SSL offloading performing decryption on the incoming connection request; SSL traffic is terminated on the load balancer sending the decrypted request to the registered target.
- **Server Name Indication (SNI):** Application load balancers support hosting multiple certificates per ALB, enabling multiple websites with separate domains to be hosted by a single ALB. Up to 25 certificates can be attached per ALB. SNI enables the assignment of the correct SSL/TLS certificate to the associated server; the ALB sends the website's or

domain's public key to the end user to establish a secure connection with the load balancer. ALB supports classic Rivest-Shamir-Adleman (RSA), the industry standard in asymmetric keys, and the newer Elliptic Curve Digital Signature Algorithm (ECDSA) for elliptic-curve cryptography. When ECDSA is compared to RSA with regard to the TLS handshake, ECDSA communication is nine times faster. ECDSA has become popular because it is used by Bitcoin, the Apple iOS, and iMessage.

- **Dynamic port mapping:** Application load balancers support load-balancing containers running the same service on the same EC2 instance where the containers are hosted. When Amazon EC2 Container Service (ECS) task definitions are launched multiple times on the same EC2 instance, the containers are running duplicates of the same service; dynamic port mapping process assigns a random port to each container task.
- **Connection draining:** When an EC2 instance that is registered with a load balancer is tagged as unhealthy by failing its health checks, the connections to the instance are closed through a process called ***connection draining***. From the point of view of the load balancer, the connection draining process keeps existing connections open until the

client closes them but prevents new requests from being sent to the instances that are tagged as unhealthy.

Connection draining removes select EC2 instances from a load balancer target group when maintenance is required—for example, when it's time to update a healthy EC2 instance with a new Amazon Machine Image (AMI). Performing the deregistration process on an EC2 instance (see [Figure 11-10](#)) starts the connection draining process, keeping the existing connections open to provide enough time to complete all ongoing requests. An EC2 instance that is in the process of deregistering will not accept new connection requests.

Details	Targets	Monitoring	Health checks	Attributes	Tags
Registered targets (1/1)					
<input type="text"/> Filter resources by property or value					
Instance ID	Name	Port	Zone	Health status	Health status details
<input checked="" type="checkbox"/> i-06371ecb1dbb60aed		80	us-east-1b	 draining	Target deregistration is in progress

Figure 11-10 Connection Draining to Deregister Instances from Target Groups

- **Cross-zone load balancing:** The nodes for the load balancers distribute incoming traffic requests evenly across the registered targets in the enabled availability zones. If cross-zone load balancing is disabled, each load balancer node

distributes traffic across the registered targets in its assigned AZ. Cross-zone load balancing can be disabled at the target group level; it is enabled by default for ALBs and disabled by default for NLBs.

- **User authentication:** Application Load Balancer allows you to offload the authentication process so the load balancer can authenticate users as they request access to cloud applications. ALB integrates with AWS Cognito, which allows both web-based and enterprise identity providers to authenticate through the ALB.
- **HTTP/2 and gRPC Support:** HTTP/2 allows multiple requests to be sent across the same connection. ALB can load balance gRPC traffic between microservices and gRPC-enabled clients and services. gRPC uses HTTP/2 for routing communications for microservice architectures.

Note

If you're deploying an ALB, you can also add a Web Application Firewall (WAF) ACL for additional protection against malicious incoming public traffic.

Application Load Balancer Deployment

When ordering an Application Load Balancer, choose whether the load balancer accepts public inbound traffic (Internet-facing) or private inbound traffic (internal), as shown in [Figure 11-11](#). Also select the IP address type to be used: IPv4 or Dualstack (IPv4 and IPv6).

Basic configuration

Load balancer name
Name must be unique within your AWS account and cannot be changed after the load balancer is created.
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme cannot be changed after the load balancer is created.
 Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.
 IPv4
Recommended for internal load balancers.
 Dualstack
Includes IPv4 and IPv6 addresses.

Figure 11-11 Initial Configuration of ALB

Next, select the VPC, availability zone(s), and the subnets the Application Load Balancer will be linked to. The ALB is always hosted in public subnets for Internet-facing applications. Public-facing load balancer deployments also require that an Internet

gateway be attached to the VPC where the load balancer is being installed. When you enable an AZ for an ALB, the ELB service creates an ALB node in each AZ. For ALB deployments, at least two AZs are required, ensuring that if one AZ becomes unavailable or has no healthy targets, the ALB will route traffic to the healthy targets hosted in another AZ (see [Figure 11-12](#)).

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be confirmed until you confirm the VPC for your targets, view your [target groups](#).

Dev VPC

vpc-6d30d915
IPv4: 192.168.0.0/16

Mappings [Info](#)

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will automatically select subnets from the chosen Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once selected.

us-east-1a

Subnet

subnet-f74284bc	Public Subnet for ALB and NAT AZ-A ▾
-----------------	--------------------------------------

IPv4 settings

Assigned by AWS

us-east-1b

subnet-265f5f7c	Private Subnet for Web Servers AZ - B
subnet-e15959bb	Public Subnet for ALB and NAT AZ-B
subnet-265f5f7c	Private Subnet for Web Servers AZ - B ▲

Figure 11-12 Choosing AZs and Subnets

A security group must be created or selected to allow traffic requests from clients to the ALB. Allowed client traffic and health check traffic is sent to the respective target groups on the

listener port—for example, port 80 or port 443. The security group for the ALB controls the traffic that is allowed to reach the load balancer; it does not affect the traffic that is forwarded to the targets in the target group. The ALB must be able to communicate with registered targets on both the listener port and the defined health check port, both inbound and outbound.

Listeners and Routing

A *listener* continuously checks for incoming connection requests based on the defined ports and protocols configured. Incoming connection requests that match are forwarded to a target group. Common protocol options are port 80 and port 443 (see [Figure 11-13](#)). An ALB listener supports HTTP/HTTPS and ports from 1-65535. Redirect actions can also be deployed to redirect client requests from one URL to another, such as HTTP to HTTPS, or HTTP to HTTP.



Figure 11-13 ALB Listener Setup

After an initial listener has been configured and the ALB has been launched successfully, additional listeners can be added by editing the ALB properties. ALB HTTPS listeners use a feature called SSL offload, which supports encrypted traffic between the client and the load balancer and decrypts traffic sent from the load balancer to registered targets. To ensure that registered targets decrypt HTTPS traffic instead of the ALB, create a Network Load Balancer with a TCP listener on port 443. With a TCP listener, the load balancer passes encrypted traffic directly to the targets without decrypting it first.

Elastic Load Balancing uses security policy to negotiate SSL connections between a client and the load balancer. For HTTPS listeners listening on port 443, an X.509 certificate must also be associated with the secure listener. Use AWS Certificate Manager (ACM) to first upload your organization's SSL/TLS website or domain certificate; then select the certificate and select a security policy that is applied to all frontend connections. Uploading a custom security policy to secure the backend communications is not allowed. Each request accepted by the listener uses two connections:

- **A frontend connection between the client and the load balancer:** Organizations choose the security policy for frontend connections. During the connection negotiation

between the client and the ALB, a set of ciphers and protocols is presented by the client and ALB and a cipher is selected for the secure connection.

- **A backend connection between the load balancer and the associated target:** The ELBSecurityPolicy-2016-08 security policy is always used for securing backend connections. Application Load Balancers do not support custom security policies.

By default, when frontend or backend load balancing connections have not processed data for 60 seconds, the connections are closed. Connections can be left open for a longer time by editing the default attributes of the load balancer.

Note

Certificates uploaded to the ALB from Certificate Manager are automatically renewed by the Certificate Manager service.

Rules, Conditions, and Actions



A *rule* consists of a set of conditions and an action. When the ALB receives a request, it evaluates the conditions in the rule to determine whether the action should be taken. If the conditions are met, the action is performed and the request is routed to the specified target group. If the conditions are not met, the next rule is evaluated. Each listener has at least one default rule and action defined for routing traffic to a target group (see [Figure 11-14](#)). Default rules don't have conditions. Additional rules can be created with defined conditions; if the conditions are met, the rules' actions are performed; if the conditions are not met, the default rule is used instead. With multiple rules, the rules are evaluated in priority from the lowest to the highest value. Multiple rules can be created for an ALB, and each rule can have multiple conditions. The order of the rules is important because the ALB evaluates the rules in the order in which they are specified. You can specify the order of the rules being processed using the priority field.

Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTPS:443 Remove

Protocol	Port	Default action	Info
HTTPS ▾	: 443 1-65535	Forward to tg	HTTP ▾ C
Create target group C			

[Add listener](#)

Secure listener settings [Info](#)

These settings will apply to all of your secure listeners. Once created, you can manage these settings per listener if desired.

Security policy

Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration, known as a security policy, to negotiate SSL connections with clients.

ELBSecurityPolicy-FS-1-2-Res-2020-10 ▼

[Compare security policies](#) [Request new ACM certificate](#) Select a certificate ▾ C

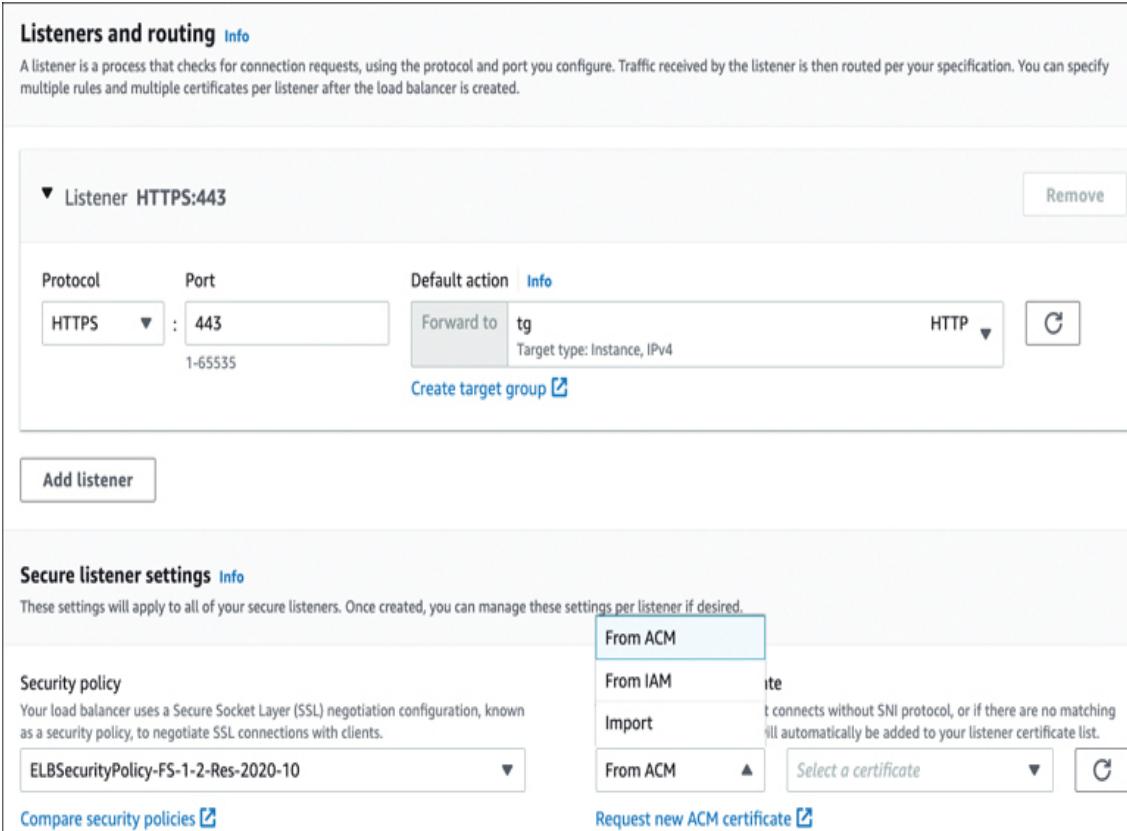


Figure 11-14 ALB Listener and Routing Setup

Each rule must include one of the following actions; the action with the lowest defined value is performed first:

- **forward:** This routing option forwards the request to a specific target group.
- **redirect:** This routing option redirects the request from one URL to another. Usable components include the protocol (HTTP to HTTP, HTTP to HTTPS, and HTTPS to HTTPS), hostname, port, or path.

- **fixed-response:** This routing option sends a custom HTTP response to the end user.

The following routing conditions are supported for rules:

- **host-header:** This routing option forwards requests to a target group based on the domain name contained in the host header. When the hostname in the host header matches the hostname in the listener rule, the request is routed. Wildcard characters can be used in the first part of the hostname but not in the part of the name after the period (**name*.com*). For example, requests to a.example.com could be sent to one target group, and requests to b.example.com could be sent to another target group. Rules can be created that combine the path and host-based routing, allowing you to route requests to a specific path, such as /productiondocs.
- **http-header:** This routing option uses the HTTP headers (for example, Chrome or Safari).
- **path-pattern:** This routing option is based on the path pattern of the URL (for example, /images/*). If the path in the URL matches the path pattern defined in the listener's rule, as shown in Figure 11-15, the request is routed. Instead of just the root domain used as the path to send requests, endpoints can be defined at the ALB, directing the traffic requests. Both path and host-based routing allow you to

control the compute environment where the requests are being directed. Certain requests, such as API calls, could be directed to be processed on a target group of compute-optimized EC2 instances; other requests could be directed to another target group containing memory-optimized EC2 instances.

- **query-string:** This routing option is based on key/value pairs or values in the query name configuration.
- **source-ip:** This routing option is based on the source IP address for each request.

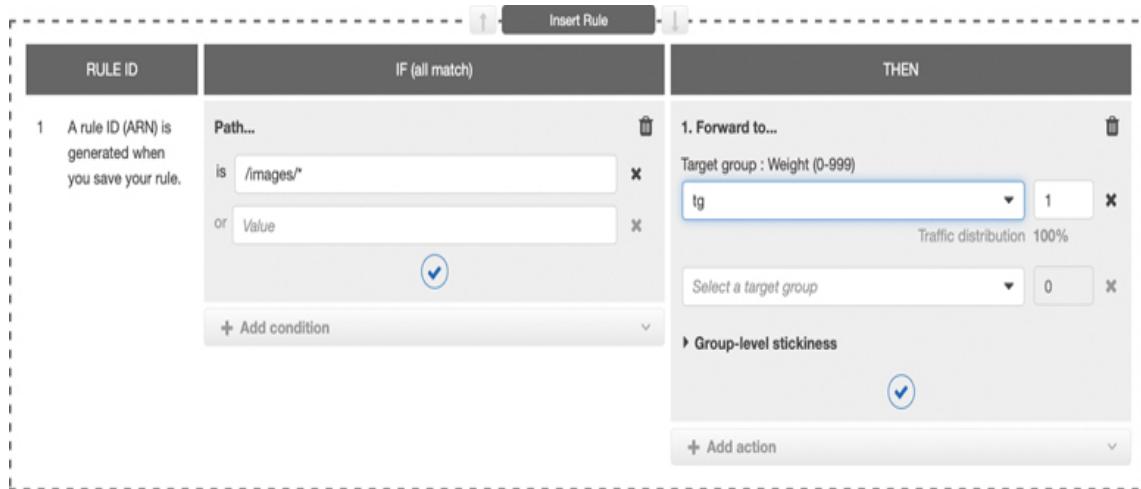


Figure 11-15 Host- and Path-Based Rules Defined for Precise Traffic Flow

There are also authentication actions for authenticating users using Cognito (**authenticate-cognito**) or a compliant OpenID Connect identity provider (**authenticate-oidc**).

Target Groups



A **target group** routes requests to one or more registered targets. Once a registered target has passed its health checks, the load balancer will route connection requests to the target. There are four choices for target groups, as shown in [Figure 11-16](#):

- **EC2 Instances:** EC2 instances that are located in the AWS VPC defined by the target group. Load balancers that are linked to an Auto Scaling group use EC2 instances that are defined by instance ID.
- **IP addresses:** IPv4 or IPv6 addresses for cloud hosted EC2 instances or on-premises servers.
- **Lambda function:** Register a Lambda function as targets and configure a listener rule forwarding requests to the target group for the Lambda function.
- **Application Load Balancer:** Associate an ALB as the target for NLB traffic.

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Figure 11-16 Adding a Target Group to ALB

Health Checks

Health checks are used to monitor the status of targets in a load balancer's target group. Requests for online availability status are sent to registered targets at a configured interval to verify that registered targets are available to serve traffic. If a target

fails a health check, it is removed from the target group and will not receive traffic until it is deemed healthy again.

The frequency and nature of the health checks depend on the protocol and type of target group. For HTTP and HTTPS target groups, the load balancer sends a request to the target and expects a response using a certain status code. For TCP target groups, the load balancer establishes a connection to the target and verifies that it can send and receive traffic.

The ***health check*** settings for a target group can be customized to define the ping target, the interval between health checks, and the number of consecutive failures required before marking a target as unhealthy (see [Figure 11-17](#)). You can also specify a healthy threshold and an unhealthy threshold, which determine the number of successful or unsuccessful health checks required before marking a target as healthy or unhealthy.

Targets	Monitoring	Health checks	Attributes	Tags
Health check settings				
Protocol HTTP	Path /	Port Traffic port	Healthy threshold 5 consecutive health check successes	
Unhealthy threshold 2 consecutive health check failures	Timeout 5 seconds	Interval 30 seconds	Success codes 200	

Figure 11-17 Load Balancer Health Check Settings

If the EC2 instance responds within the defined response timeout period, the load balancer marks the EC2 instance as in service, and incoming user requests are routed to the healthy targets. Both the ALB and NLB perform health checks against all registered EC2 instances at specific intervals. Health checks are configured during the load balancer setup and configuration and can be changed at any time. The following language is used to describe health checks:

- A registered target is typically defined as *healthy* or *unhealthy*.
- A target newly added to the target group is defined as *initial*; once its health check is successful, the target is defined as *healthy*.
- When a registered target is being removed and connection draining is underway, the target is marked as *draining*.

[Table 11-2](#) lists the options that can be defined to customize advanced health checks. Health checks can be defined or modified for each target group by selecting the health check tab.

Table 11-2 Health Check Settings

Health

Check

Setting

Description

Health

Either HTTP or HTTPS.

Check

Protocol

Health

The port used for performing health

Check

checks on targets. The default is the

Port

communications protocol port, which is

either 80 or 443.

Health

The destination ping path on the target.

Check

The default is /.

Path

Health

The amount of time (from 2–60 seconds)

Check

after which a health check is considered

Timeout

failed.

Seconds

Health Check Setting	Description
Health Check Interval Seconds	The time between health checks (in the range of 5–300 seconds).
Healthy Threshold Count	The number of consecutive health checks required from an unhealthy target before the target is considered healthy.
Unhealthy Threshold Account	The number of consecutive failed health checks that result in an unhealthy target.
Status code	The HTTP code (in the range 200–499) that indicates a healthy target.

Resilient workloads use health checks to ensure that resources placed behind load balancers (ALB/NLB) are available. EC2 Auto Scaling can also monitor ELB health checks when EC2 instances

are automatically scaled using Auto Scaling groups. [Chapter 9](#) provides additional details on EC2 Auto Scaling.

Target Group Attributes



Each EC2 instance or target group's operation can be controlled by modifying the target group attributes, as shown in [Table 11-3](#).

Table 11-3 ALB Target Group Attributes for Instance or IP Targets

Attribute	Description
Deregistration delay	How much time before a target (instance or IP address) is deregistered. The default is 300 seconds.

Attribute	Description
Slow start duration	The time before a new target is sent a gradually increasing number of connection requests. It can be set to up to 15 minutes, and there is no default setting.
Round-robin load-balancing algorithm	Enabled or disabled.
Least-outstanding requests load-balancing algorithm	Enabled or disabled.
Stickiness	Enabled or disabled.

Configure health checks for each target group shown in [Figure 11-18](#) by editing the target group health checks attributes.

Healthy threshold	The number of consecutive health checks successes required before considering an unhealthy target healthy.
<input type="text" value="5"/>	2-10
Unhealthy threshold	The number of consecutive health check failures required before considering a target unhealthy.
<input type="text" value="2"/>	2-10
Timeout	The amount of time, in seconds, during which no response means a failed health check.
<input type="text" value="5"/> seconds	2-120
Interval	The approximate amount of time between health checks of an individual target
<input type="text" value="30"/> seconds	5-300

Figure 11-18 Configuring Health Checks

Sticky Session Support

If a load balancer is supporting an application that is providing generic information, maintaining a specific user session might not be required. However, for applications where the end user begins communication with an initial server, maintaining the session between the end user and the backend resource is important. If you are buying something online, you expect your session to begin and end properly, without problems.

An ALB supports ***sticky sessions***, which allow the load balancer to bind the user's active session to a specific EC2 instance. With sticky sessions enabled on a load balancer, after a request is routed to a target, a cookie is generated by the load balancer or application and returned to the client, ensuring that requests are sent to the EC2 instance where the user session is located. All requests from the client to the load balancer include the identifying cookie, ensuring that all requests are routed to the same backend server. The enabling of sticky sessions and the parameters for the stickiness of the cookie are defined by editing the target group attributes tab, as shown in [Figure 11-19](#).

Targets	Monitoring	Health checks	Attributes	Tags
Attributes				
Stickiness			Deregistration delay	
Enabled			300 seconds	
Stickiness type			Stickiness duration	
lb_cookie			1 day	
Slow start duration			Load balancing algorithm	
0 seconds			Round robin	

Figure 11-19 Target Group Attributes

You can enable sticky sessions for an ALB by specifying a duration for the stickiness period. The stickiness period is the length of time that the ALB should route requests from the same user to the same target.

To enable sticky sessions you must define a stickiness policy when you create a target group. A stickiness policy defines the method that the ALB should use to bind a user's session to a target. The available stickiness policies are

- **Source IP:** This policy uses the client's IP address to bind the session to a target.
- **Application-based cookies:** This policy uses a cookie to bind the session to a target. You can specify the name and duration of the cookie.

What happens when the backend server that the user is connected to fails and is no longer available? The load balancer automatically chooses a new healthy EC2 instance and moves the user to a new server for the remainder of the session, even if the old instance becomes available. Sticky sessions are useful when everything works, but they're not useful when servers fail, as the new EC2 instance knows nothing about the user's previous session.

Instead of enabling sticky sessions, consider using a central storage location for user session information, for example a hosted ElastiCache for Redis cluster or ElastiCache for Memcached nodes. For applications with a large number of concurrent user sessions, one of these choices will be a better option to provide resilient storage for user session information.

Access Logs

You can choose to enable access logs, which provide detailed information about all incoming requests sent to the load balancer. Once access logs are enabled, ELB captures the logging details and stores them in the desired Amazon S3 bucket. Additional security can be provided by enabling server-side encryption on the bucket to encrypt each access log file. Use S3 managed encryption keys to ensure that each log file is encrypted with a unique Amazon S3 managed key. Automatic key rotation is carried out by the Key Management Service (KMS) service.

Log files are published every 5 minutes. Log details include the type of request or connection (that is, HTTP, HTTPS, HTTP2, WebSocket, or WebSocket over SSL/TLS) and the timestamp, client port, target port, request and target processing time, and sent and received bytes. Details provided by CloudWatch

logging can also be provided by access logs for a fraction of the cost of using CloudWatch metrics and alarms.

ALB Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of ALB:

- ALB operates at Layer 7, routing traffic to registered targets—EC2 instances, containers, and IP addresses—based on the content of the incoming request.
- ALB supports HTTP/HTTPS applications and HTTPS termination between the client and the load balancer.
- ALB supports HTTP/2, which allows multiple requests to be sent on the same connection.
- SSL/TLS certificates are managed using AWS Certificate Manager.
- Server Name Indication (SNI) enables you to secure multiple websites using a single secure listener.
- ALB supports IPv4 and IPv6 for Internet-facing load balancers; for internal load balancers, it supports only IPv4.

- ALB can be integrated with Amazon Cognito to provide end-user authentication.
- ALB uses either round-robin or a least-available-request algorithm for targeting registered EC2 instances.
- ALB supports AWS Outposts.
- AWS Certificate Manager or AWS IAM can be used to manage server certificates.

Network Load Balancer

ELB Network Load Balancer (NLB) is a load balancing service that distributes incoming traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses, in one or more AZ. Network Load Balancer provides TCP and UDP load balancing at Layer 4 of the OSI stack. NLB uses a flow-based algorithm to distribute traffic to the targets in a target group. This means it distributes traffic based on the number of connections rather than on the amount of data transferred. The NLB can scale to handle millions of requests per second at very low latencies. It can also integrate with EC2 Auto Scaling, Amazon ECS, and AWS ACM. NLB supports end-to-end encryption using TLS.

You should know the following NLB features for the AWS Certified Solutions Architect – Associate (SSA-C03) exam:

Key Topic

- **TLS offloading:** Client TLS session termination is supported, allowing TLS termination tasks to be carried out by the load balancer.
- **Server Name Indication (SNI):** Serves multiple websites using a single TLS listener.
- **AWS Certificate Manager:** Manages server certificates.
- **Sticky sessions:** Can be defined per target session.
- **Preserve client-side source IP address:** Backend servers can see the IP address of the client.
- **Static IP address:** A static IP address is provided per AZ.
- **EIP support:** An Elastic IP address can be assigned for each AZ.
- **DNS fail-over:** If there are no healthy targets available, Route 53 directs traffic to load balancer nodes in other AZs.
- **Route 53 integration:** Route 53 can route traffic to an alternate NLB in another AWS region.
- **Zonal isolation:** The NLB can be enabled in a single AZ, supporting applications that require zonal isolation.

NLB Cheat Sheet

Key Topic

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of NLB:

- The NLB can load balance applications hosted at AWS and on premises using IPv4/IPv6 addresses.
- The NLB supports connections across peered VPCs in different AWS regions.
- The NLB supports long-running connections, which are ideal for WebSocket applications.
- The NLB supports failover across AWS regions, using Route 53 health checks.
- With the NLB, the source IP addresses of the clients that are connecting are preserved.
- Each NLB allows for extremely high throughput; an NLB can scale and handle millions of requests per second.
- The NLB flow-based algorithm is ideal for latency-sensitive TCP/UDP applications.
- The NLB provides “end-to-end security” with TLS termination performed by the NLB.

Multi-Region Failover

An NLB supports failover across AWS regions using Amazon Route 53 health checks, allowing organizations to create a highly available, globally distributed load balancing solution that can route traffic to the optimal region based on the health of the targets in each region.

An NLB must be created in each AWS region where traffic will be load balanced. Then create a target group in each region that contains the regional targets to which to route traffic. Enable cross-zone load balancing for each NLB so traffic is distributed across the targets in each AZ.

Next, create an Amazon Route 53 health check for each target group. You can use the default health check configuration or customize the health check settings to meet your specific requirements.

Finally, create a Route 53 record set that points to the NLB in each AWS region. Choices are a weighted record set, or a latency-based record set to specify the routing policy for the record set. With a weighted record set the proportion of traffic that should be routed to each region is controlled based on the weights assigned to the record set. With a latency-based record set, Route 53 routes traffic to the region that provides the lowest latency for the end user.

CloudWatch Metrics

CloudWatch metrics for ELB can be used to monitor and ensure that a workload is performing as expected. [Table 11-4](#) lists several metrics that provide operating details based on the sum of the totals.

Table 11-4 CloudWatch Metrics for ELB

ELB Metric	Description
ActiveConnectionCount	The number of concurrent TCP frontend and backend connections
ConsumedLCUs	The number of Load Balancer Capacity Units used
NewConnectionCount	The total number of TCP connections from clients to the load balancer to targets

ELB Metric	Description
ProcessedBytes	The total number of bytes processed by the load balancer
RequestCount	The number of requests processed with responses from a target
HealthyHostCount	The number of targets that are healthy
UnhealthyHostCount	The number of targets that are unhealthy
RequestCountPerTarget	The average number of requests received by each target in a group

AWS VPC Networking

The networking layer at AWS is called a ***virtual private cloud (VPC)***. Each customer's EC2 instances and containers and EBS storage must be deployed in a VPC (see [Figure 11-20](#)). Elastic Cloud Compute instances are always hosted within an AWS VPC. Software that runs on a Windows or Linux virtual server as web servers or application servers, databases, third-party virtual appliances, and AWS ECS or AWS EKS deployments also run on EC2 instances hosted in AWS VPCs.

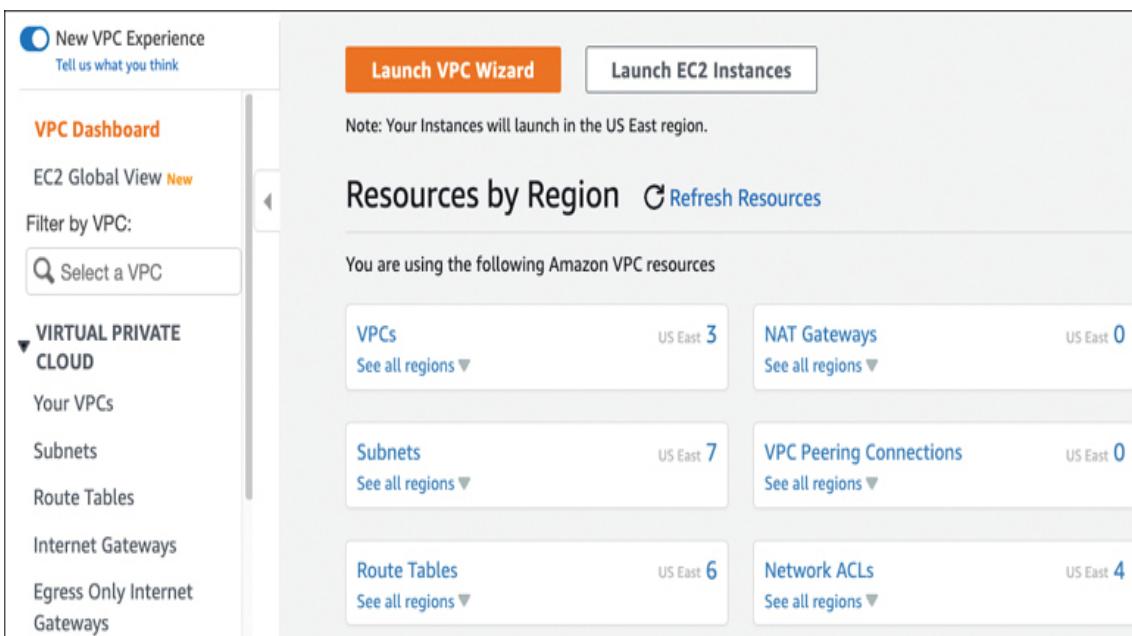


Figure 11-20 VPC Dashboard

When you create a VPC, AWS secures your VPC as a private, isolated software data center linked to your AWS account. AWS provisions, hosts, and secures each VPC; the remaining

configuration is your responsibility. Amazon is responsible for safeguarding and protecting all VPC networks; Amazon must ensure the continued separation of your VPC from those of all other AWS customers. As described in the next section, the shared responsibility model specifies that AWS is responsible for the security *of* the cloud; each organization is responsible for maintaining workload security *in* the cloud.

There are lots of moving parts and pieces at AWS, which I like to describe as a large toolbox containing a variety of tools and attachments that you can snap together in any way that suits your design needs. Within the VPC toolbox are many configurable options, including route tables, public and private subnets, VPN connections, gateways, and private endpoints. In addition, there are multiple security choices available at every network level, allowing you to fully protect your EC2 instances and containers; choices include the AWS Network Firewall, the DNS Firewall, security groups (SGs), ***network access control lists (NACLs)***, and more.

A VPC also has public and multiple private connectivity options, allowing you to connect your VPC to the Internet, to a private data center, or to other VPCs within or outside your region. Every cloud service that you order and deploy at AWS is hosted on the AWS network. It's up to you to plan where you want a

service to be deployed, keeping in mind the goal of creating reliable, highly available, and secure workloads.

The Shared Security Model



When you host your applications in the Amazon public cloud, you have implicitly agreed to work in partnership with AWS in what is typically defined as a *shared security model*, as shown in [Figure 11-21](#). AWS has responsibilities for building and securing its cloud infrastructure; this is typically referred to as *security of the cloud*. Your responsibility as an AWS customer is to design acceptable security provisions for your applications and data hosted on the AWS cloud. The level of acceptable security provisions is entirely your choice. Customers are therefore responsible for maintaining their *security in the cloud*. After AWS carries out the creation of a custom VPC, each customer makes the remaining design choices and security decisions.

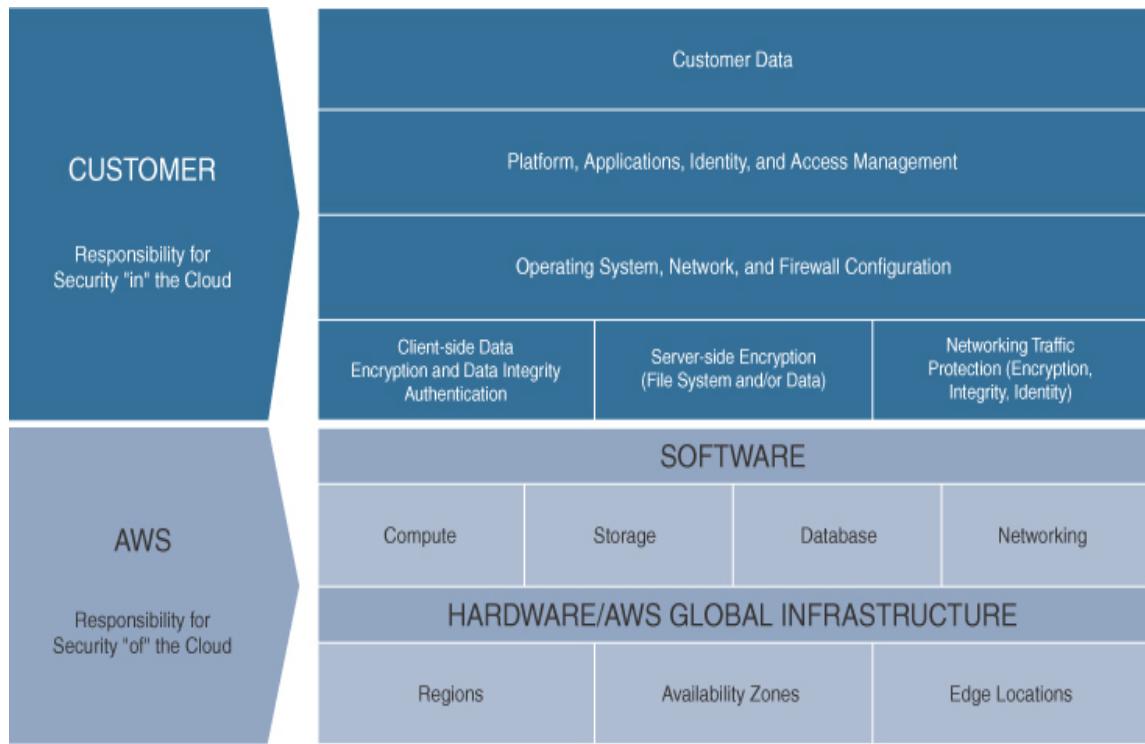


Figure 11-21 Shared Security Model

Within each VPC, EC2 compute instances are hosted on subnets that are created in selected AZs within the AWS region in which you have chosen to operate.

AWS Networking Terminology

Some of the AWS networking terms defined in this chapter may be new to you, whereas the names of other networking components and services that are used at AWS likely will sound familiar—for example, subnets, public and private IP addresses, and route tables. The networking services that are exposed to

each customer at AWS are not the same network hardware devices that are deployed in customer data centers. Hosting hundreds of thousands of customers in a massive, shared networking environment requires networking services that will be different from your on-premises networking services due to the size and scope of Amazon's overall operations.

The first major concept of networking at AWS is that within each VPC, the networking exposed to each customer is designed and managed at the subnet level—specifically, the Layer 3 subnet address space contained within each availability zone. That's as deep as we're going to get in the network stack at AWS.

Your on-premises networking environment is probably composed of virtual local area networks (VLANs), Layer 2 networks, and Multiprotocol Label Switching (MPLS) connections. Why does a customer's exposure to the AWS network then start and end at Layer 3? Because thousands of customers running on a massively shared network infrastructure at AWS is at a scale that far exceeds the scale utilized within your own data centers. As a result, the internal network design offered to each customer needs to be different as well.

AWS does not deploy VLANs on the internal private AWS network because they cannot scale to the number of customers that Amazon hosts. A VPC also doesn't use MPLS for communication; however, you may be utilizing MPLS connections when connecting to a VPC using an external AWS Direct Connect or AWS Transit Gateway connection from your on-premises network.

Each VPC is a software-defined network built with Amazon's own code and custom network hardware developed by AWS to match its required scale of network operations. The underlying physical network at AWS would be quite recognizable at the component level; however, AWS customers don't have access to the physical network—that's restricted to the folks at AWS and it's their job to maintain it.

EC2 instances run on a hypervisor installed on custom-designed bare-metal servers (see [Figure 11-22](#)). For a number of years, the standard hypervisor that AWS used was a customized version of Xen, but many changes have happened at AWS. All new EC2 instance types since late 2017 are deployed on the Nitro System, a customized version of the KVM hypervisor with a published benchmark of less than 1% differential when comparing virtual EC2 instance performance to bare-metal system performance. The Nitro System uses Nitro chipsets that

monitor the firmware and hardware at boot and support enhanced networking and EBS storage access and NVMe high-speed local SSD storage across a PCI bus.

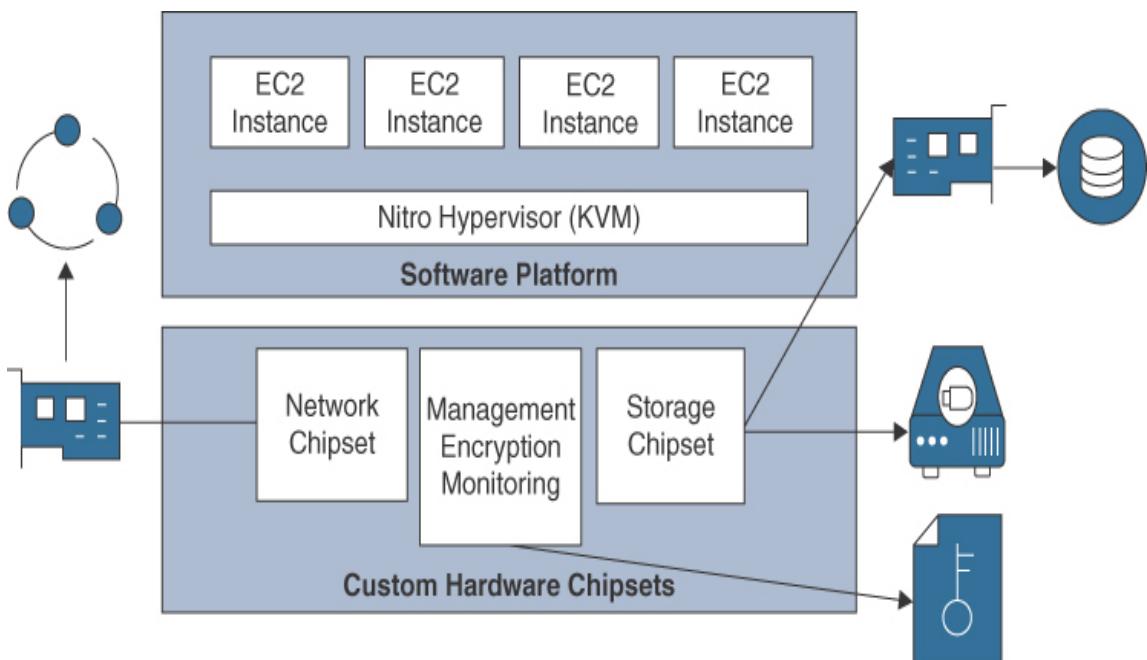


Figure 11-22 Instance Hosting by the Nitro Hypervisor

VPC Cheat Sheet

Key Topic

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of VPCs:

- You can create private IP addresses from any private address range that adheres to RFC 1918 address allocation rules.
- You can expand your VPC by adding additional CIDR ranges.
- You can create both private and public subnets.
- It is possible to control both inbound and outbound subnet traffic by using NACLs.
- You can access most AWS services privately from your VPC by using private VPC endpoints and gateway connections.
- You can privately connect to third-party services hosted at AWS using a PrivateLink connection.
- You can connect on-premises resources privately with a VPC with a site-to-site connection.
- You create VPN connections to your VPC only after a virtual private gateway (VPG) has been installed.
- You can use VPC flow logs to log network traffic information to CloudWatch logs or S3 buckets for further analysis.
- With a VPC, you can deploy both IPv4 and IPv6 addressing.
- You can connect a VPC with another VPC by using a peering connection.
- You can assign Elastic IP addresses to EC2 instances for public Internet access.
- You can assign multiple elastic network interfaces to EC2 instances.

- You can protect access to EC2 instances by using one or more security groups.
- The Network Firewall is a managed service that provides network intrusion protection for VPCs using stateful firewall rules.
- The DNS Firewall allows the filtering and protection of DNS queries to public domain names from EC2 instances hosted in a VPC.
- The VPC Reachability Analyzer performs connectivity testing between a source and destination resource hosted in a VPC providing hop-by-hop details and the blocking component.
- The Network Access Analyzer helps identify network configurations that have security issues through testing current ingress and egress paths.
- You can control subnet traffic flow by defining custom route tables and route table entries.
- You can customize DHCP options to suit your needs by using DHCP options sets.
- You can enable private subnets to get updates by using **NAT gateway services** or NAT EC2 instances.
- You can protect IPv6 EC2 instances from direct communication from the Internet by deploying an **egress-only Internet gateway (EOIG)**.

- You can connect thousands of VPCs, attachments, and gateway connections together in a custom network deployment by using transit gateways, transit gateway peering, and transit gateway route tables.
- You can route multicast traffic between attached VPCs by creating a transit gateway multicast domain.
- You can capture and mirror network traffic for select EC2 instances by using VPC traffic mirroring.

Creating a VPC

The initial creation of a VPC is either a very simple process or a slightly more complex process, depending on the options you choose to deploy using the Create VPC wizard from the VPC Dashboard. You can also use the Amazon command-line interface (CLI) and enter a simple command-line string to create a VPC.

Using the Create VPC Wizard

For this example, you begin by clicking the Your VPCs link on the VPC Dashboard. Here are the steps in creating a VPC by using the Create VPC wizard:

Step 1. In the AWS Management Console, click Services, and under Networking and Content Delivery, select VPC. The VPC

Dashboard appears.

Step 2. In the VPC Dashboard, under Virtual Private Cloud, select Your VPCs.

Step 3. Click the Create VPC button.

Step 4. The selected resources to create are VPC only.

Step 5. In the Name tag text box (see [Figure 11-23](#)), enter the name (tag) of your VPC.

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)

Figure 11-23 Using the Create VPC Wizard

Step 6. Select the IPv4 CIDR manual input option and manually enter the desired IPv4 CIDR range. For example, entering 192.168.0.0/16 would allow you to create subnets within the VPC that could total approximately 65,530 possible hosts. The valid CIDR ranges supported by AWS are /16 to /28. For further details, see <https://tools.ietf.org/html/rfc1519>.

Step 7. If required, in the IPv6 CIDR block section, select either IPAM-allocated IPv6 CIDR block, Amazon-provided IPv6 CIDR block, or IPv6 CIDR owned by me.

Step 8. Optionally, change the Tenancy setting from Default (for shared tenancy) to Dedicated. Shared tenancy allows you to host EC2 instances that use default shared tenancy. Choosing dedicated tenancy mandates that all EC2 instances hosted in this VPC are dedicated EC2 instances.

Step 9. Looking again at the top of [Figure 11-23](#), if you choose the VPC and More radio button under Resources to Create, you can select IPv4 and IPv6 CIDR blocks up to two public subnets, up to three private subnets, associated route tables, up to three AZs, NAT gateways, S3 gateway endpoints, and custom DNS options, as shown in [Figure 11-24](#). This figure is an excellent study guide for VPC network options.

Key Topic



Figure 11-24 VPC Starting Design Choices

Note

Hosting EC2 instances at AWS in a VPC means other AWS customers will also be sharing the underlying bare-metal server hardware and hypervisor with you. You will not know which customers you are sharing AWS resources with. Selecting dedicated tenancy allows you to run your EC2 instances within a VPC designated as dedicated on single-tenant hardware where you are the only tenant or customer. This might be an important consideration if your organization is bound by

strict governance rules that require it to operate with dedicated compute EC2 instances when operating in the AWS cloud.

Running dedicated EC2 instances at AWS is more expensive than multi-tenancy operations; a \$2 charge per hour is added when dedicated EC2 instances are leveraged.

Using the AWS CLI to Create a VPC

The Create VPC wizard provides a starting point for creating a VPC network. You can also choose to use AWS CLI commands to create a VPC, as shown in [Figure 11-25](#).

```
# Create a VPC with a /16 network and enable DNS hostnames
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --enable-dns-hostnames

# Create two subnets in each Availability Zone
aws ec2 create-subnet --vpc-id <vpc-id> --cidr-block 10.0.0.0/24 --
availability-zone us-east-1a
aws ec2 create-subnet --vpc-id <vpc-id> --cidr-block 10.0.1.0/24 --
availability-zone us-east-1a
aws ec2 create-subnet --vpc-id <vpc-id> --cidr-block 10.0.2.0/24 --
availability-zone us-east-1b
aws ec2 create-subnet --vpc-id <vpc-id> --cidr-block 10.0.3.0/24 --
availability-zone us-east-1b
```

Figure 11-25 Creating a VPC by Using the AWS CLI

How Many VPCs Does Your Organization Need?

Depending on the creation process you have used, your new custom VPC might contain the required IPv4 or IPv6 CIDR blocks, or you might have fleshed out your design by choosing public and private subnets and associated network connectivity services. Now is a good time to pause and think about how many VPCs are required for your workload in total. Your company may have many developers creating their own set of VPCs without regard for other developers who are also creating VPCs. How much growth will you need over the next 2 years or more?

For example, suppose that your company is creating a custom human resources application; therefore, one VPC is required for the production workload. Will a separate VPC be useful for development? Perhaps an additional VPC could test for quality control. What if your company decides to operate in multiple AWS regions? What if multiple developers with multiple AWS accounts work on the development of the human resources system in different countries and with separate AWS accounts? Hopefully, you appreciate how complicated network decisions can become.

There are initial AWS service quotas on the number of VPCs that you can create. The number of VPCs each AWS account can create is a maximum of five VPCs per AWS account per AWS

region. The Service Quotas utility is used for requesting the ability to create additional AWS resources including additional VPCs. Each AWS service has an initially defined quota limit assigned. Check the AWS documentation for each AWS service that you are planning to deploy to find the current and maximum quota limits and plan accordingly.

Note

Service quotas are per AWS account per region. With 30 regions available at AWS, a single AWS account could create 150 VPCs—5 per AWS region.

Consider these criteria for calculating the number of VPCs required:

- Your organization wants to extend, or burst, into the cloud, using resources in the corporate data center and cloud services when necessary at AWS. The primary need is to deploy additional compute resources at certain times of the month when additional performance is required. For this scenario, one VPC could be enough. A single VPC can host many subnets and EC2 instances with private connectivity back to a corporate data center.

- You are an independent developer creating a SaaS application that will be available across the Internet to users around the world. You have no corporate data center. You require a separate development, testing, and production workspace—three VPCs within a single region would be a good starting point.
- You are an administrator who has been tasked with leveraging cloud storage at AWS. You need unlimited storage, and you don't know the upper limit of your storage requirements. Your solution doesn't require a VPC. You need storage—perhaps S3 object storage or S3 Glacier archiving. The AWS S3 storage service does not reside within a VPC.
- You work for a large company that must follow specific compliance rules that dictate that workload resources must always remain separated. Separate VPCs for each workload must be created for development, testing, and production.

Creating the VPC CIDR Block

A VPC created using either an AWS CLI command or the Create VPC wizard is a blank slate except for the primary IPv4 Classless Inter-Domain Routing (CIDR) block and the local main routing table. Here are some CIDR details to be aware of:

- Both IPv4 and IPv6 subnets are supported within a VPC; however, a VPC or a subnet must have an initial IPv4 CIDR block defined first.
- IPv6 CIDR blocks can be associated with your VPC, but only after an initial IPv4 CIDR block has been created.
- Only IPv6 subnets can be created in a dual-stack (IPv4/IPv6 CIDR) VPC.
- CIDR blocks can't overlap with any existing CIDR blocks associated with another VPC connected with a peering connection. Overlapping CIDR blocks are to be avoided unless it's a deliberate decision to ensure that a VPC cannot connect to another VPC, regardless of the situation.
- The size of an existing CIDR block cannot be increased or decreased; it is locked after creation.

An Amazon-provided IPv6 CIDR block is a range of IPv6 addresses that can be used to create an IPv6 VPC and assign IPv6 addresses to your resources, such as EC2 instances and ELBs. When you create an IPv6 VPC, you can choose to use an Amazon-provided IPv6 CIDR block or specify your own IPv6 CIDR block. If you choose to use an Amazon-provided IPv6 CIDR block, AWS will automatically assign a range of IPv6 addresses to your VPC. Amazon-provided IPv6 CIDR blocks have the following characteristics:

- They are unique and globally routable, meaning they can be used to communicate with resources on the Internet.
- They are associated with your AWS account and are not transferable.
- They are automatically assigned when you create an IPv6 VPC, and you cannot specify the range of addresses that will be assigned.

Planning Your Primary VPC CIDR Block

There are many questions and many possible answers when planning IP addressing for your VPC. I can't stress it enough that if you are not the prime networking expert at your company, you should talk to your networking team and get advice on what IP address ranges you should use at AWS. Two or three years down the road, you might want to connect your network hosted at AWS to your corporate network, and you might find out that the IP address ranges selected were not the best choices. Meeting with your network team at the start of your cloud deployment will save you hours of future rework and prevent a serious meltdown. Without proper planning, your initial IP addressing choices could come back to haunt you.

Note

The primary IPv4 CIDR block and network mask that you choose for your VPC determines the number and size of IPv4 addresses that can be assigned to the subnets created within the VPC. Think of the CIDR block as a large bucket of IP addresses; the total number of addresses represents the number of EC2 instances, endpoints, AWS Lambda functions, and VPC connections that could be hosted on your VPC. The initial CIDR blocks that were added when you first created the VPC can't be changed; however, you have the option of adding additional four secondary CIDR blocks to an existing VPC.

Organizations can specify a range of IPv4 addresses for each VPC using a Classless Inter-Domain Routing (CIDR) block.

CIDR notation is a standard syntax for representing IP addresses and their associated routing prefix. It consists of an IP address and a prefix size, separated by a slash (/) character. The prefix size specifies the number of bits in the routing prefix, which determines the number of addresses in the range.

For example, the CIDR block 10.0.0.0/16 specifies a range of 256 IP addresses starting with 10.0.0.0 and ending with 10.0.255.255.

The /16 prefix size indicates that the first 16 bits of the address are used for the routing prefix, and the remaining bits are used for host addresses.

For a VPC's starting CIDR address, choosing 192.168.0.0 with a /16 network mask determines the number of possible hosts that can be contained on subnets within this single VPC (see [Figure 11-26](#).)

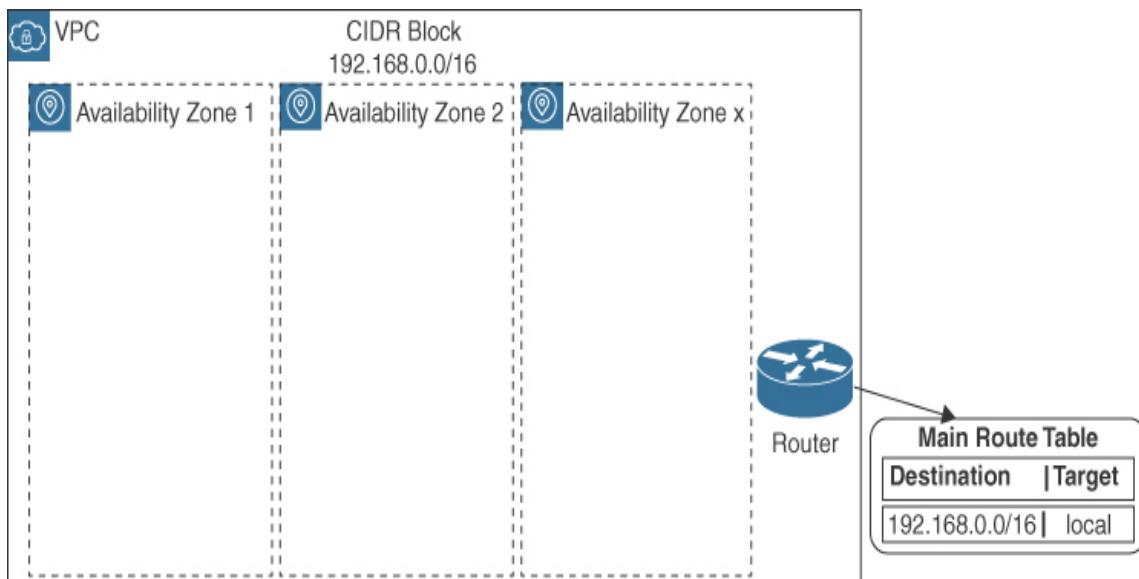


Figure 11-26 The Basic VPC Infrastructure Components

Note

Amazon supports netmask sizes from /16 to /28.

Table 11-5 lists sample address ranges from which you should be able to find an acceptable range of hosts and addresses to match your project. When in doubt, increase the subnet CIDR block size to allow the accommodation of additional hosts.

Table 11-5 VPC CIDR Block Examples

CIDR	Subnet Mask	Number of Hosts
/16	255.255.0.0	65,536
/17	255.255.128.0	32,768
/18	255.255.192.0	16,384
/19	255.255.224.0	8192
/20	255.255.240.0	4096
/21	255.255.248.0	2048
/22	255.255.252.0	1024
/23	255.255.254.0	512

CIDR	Subnet Mask	Number of Hosts
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32
/28	255.255.255.240	16

As discussed earlier, during the creation of a VPC, an IPv4 CIDR block must be assigned to each VPC, even if you're planning to use IPv6 addresses. VPCs can also operate with just IPv6 addressing or in a dual-stack mode, communicating over both IPv4 and IPv6 protocols. The subnet CIDR block for IPv6 addresses that are assigned by AWS is fixed at /64. During or after VPC creation, you can choose to associate an IPv6 CIDR block to your VPC.

Note

The first four IP addresses (0, 1, 2, and 3) and the last IP address (255) in each subnet's CIDR block are reserved for Amazon's use. Using /22 as a standard netmask for all subnets, the maximum number of hosts is 1,019. If you're creating a subnet for hosting thousands of clients using a VDI solution, you may need to pick a larger range for future expansion.

Adding a Secondary CIDR Block



Up to four secondary IPv4 CIDR blocks can be associated with an existing VPC. After you add an additional CIDR block, the new route is automatically added to the VPC's main route tables, enabling the additional local routes throughout the VPC. Routing table details are discussed later in this chapter.

Note

Keep in mind that the additional secondary CIDR block cannot be larger than the initial primary CIDR block. For example, if you associate a primary

CIDR block of 10.0.0.0/24, an additional CIDR block of the same range or larger is not allowed.

However, a CIDR block of 10.0.0.0/25 is allowed because it's a smaller range. The higher the CIDR number, the smaller the range of IP addresses available.

The primary advantage of being able to add additional secondary CIDR blocks to an existing VPC is for future expansion when necessary. If the initial primary CIDR block faces address space limitations over time, additional secondary CIDR blocks can be added in order to increase the number of IP addresses that can be assigned to subnets within the VPC. Each VPC can have up to 64,000 network access units (NAU) by default. You can request a service quota increase of up to 256,000.

The Default VPC

A default VPC is created in each AWS region, with each availability zone containing a public subnet. The default VPC is available within each AWS region and is created with the IPv4 CIDR block 172.30.0.0/16, which provides up to 65,531 private IPv4 addresses. In addition, an **Internet gateway (IG)** is

created and attached to the default VPC with a route table entry that sends all IP traffic intended for the Internet to the attached Internet gateway. A default security group and default network ACL are also associated with the default VPC. An EC2 instance placed on the default public subnet within the default VPC receives both a public and a private IPv4 address and public and private DNS hostnames. EC2 instances deployed into the default VPC automatically have Internet access.

AWS provides the prebuilt default networking VPC environment to enable you to start working with AWS quickly, even if you have limited network knowledge. The default VPC can be handy if you want to do a quick demo and don't want to bother setting up subnets and Internet connectivity or have to think about any CIDR decisions; these networking decisions have already been carried out for the default VPC.

Perhaps having a separate demo AWS account using the default VPC for demonstrations would be useful. However, the default VPC can easily cause deployment issues; the default VPC may be preselected, as shown in [Figure 11-27](#), and if you're not paying attention, using the default VPC can be trouble. For example, you might not want Internet access that has been defined for the public subnets of the default VPC by default. I recommend deleting the default VPC from every AWS region in your AWS

account. This means you must set up all your AWS networking from scratch. But perhaps in the longer term, you'll be happier knowing there's no default VPC with Internet access provided to unsuspecting developers and administrators.

vpc-c753f9a2 / default VPC				Actions ▾
Details Info				
VPC ID vpc-c753f9a2	State Available	DNS hostnames Disabled	DNS resolution Enabled	
Tenancy Default	DHCP options set dopt-52859730	Route table rtb-511fb734	Network ACL acl-5ad07c3f	
Default VPC No	IPv4 CIDR 172.30.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -	
ClassicLink DNS support Disabled	ClassicLink Disabled	Owner ID		

Figure 11-27 The Default VPC

Note

You cannot assign an existing VPC to become a default VPC, but you can delete the default VPC. If you want to re-create the default VPC, you can run an AWS-provided AWS CLI script.

Subnets

A **subnet** is a range of IPv4 or IPv6 addresses within a VPC that is associated with a specific availability zone. When you create a subnet, you specify the CIDR block for the subnet, which determines the range of IPv4 and IPv6 addresses that are available for use in the subnet.

You can create multiple subnets within a VPC, and each subnet can span one or more AZs. You can also specify a different CIDR block for each subnet.

When creating an IPv6 subnet, you will need to specify the IPv6 network range for the subnet and the number of IPv6 addresses that the subnet will contain. AWS uses a slash notation to specify the network range, similar to how IPv4 subnets are specified. For example, a subnet with a network range of 2001:db8:1234:5578::/64 would contain 64 IPv6 addresses.

Once an IPv6 subnet has been created, you can assign IPv6 addresses to resources within the subnet, such as EC2 instances, ELBs, and other resources. You can also create route tables and security groups to control inbound and outbound traffic for the IPv6 subnet.

There are several benefits to creating subnets within a VPC:

- **Network security:** Subnets can help you segment your network and isolate resources from each other, which can improve security and reduce the risk of unauthorized access.
- **Network management:** Subnets can help you organize and manage your network more efficiently, by allowing you to group resources based on their purpose or location.
- **Control traffic flow:** Subnets can help you control the flow of traffic between resources within the VPC and between the VPC and the Internet, by allowing you to specify different routing rules for different subnets.

The AZs that you select for subnet location are already available within the region where the VPC is created. It's usually stated that a VPC spans "all of the availability zones within its region," and certainly there is the potential to include all the AZs within a VPC if your design includes subnets for each AZ. However, AZs don't show up automatically in each VPC; instead, they are added during subnet creation when selecting each subnet's VPC and AZ location.

Each subnet that you create resides within its assigned AZ, as shown in [Figure 11-28](#). If you choose to design your applications for resiliency and high availability, you'll want to design your workload to operate across at least two AZs.

Subnet ID subnet-265f5f7c	Subnet ARN arn:aws:ec2:us-east-1:313858614000:subnet/subnet-265f5f7c	State Available	IPv4 CIDR 192.168.3.0/24
Available IPv4 addresses 250	IPv6 CIDR -	Availability Zone us-east-1b	Availability Zone ID use1-az6
Network border group us-east-1	VPC vpc-6d30d915 Dev VPC	Route table rtb-f6154889 private route dev vpc	Network ACL acl-dc9afca4 443 Traffic
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No

Figure 11-28 Physical Locations of Network Components

Every subnet you create begins life as a private subnet with no connectivity outside of the VPC where it is hosted. To create a subnet with Internet access, you must complete several steps:

Step 1. Order an Internet gateway (IGW).

Step 2. Associate the IGW with a VPC.

Step 3. Add a route table entry for the IGW to the subnet's route table that requires Internet access.

After you have completed these steps, you have created a public subnet.

Subnets are defined by the entries in the attached subnet route table:

- **Public subnet:** If a subnet's associated route table forwards traffic to the Internet through an Internet gateway, the subnet is defined as a public subnet.

- **Private subnet:** If a subnet's associated route table has no gateway or endpoint to direct traffic to, it is a private subnet, as traffic remains on the local subnet with no external connectivity. A subnet with no external gateway connectivity is a private subnet.
- **Protected subnet:** Protected subnets are often used to host resources that need to be isolated from the public Internet for security or compliance reasons, such as database servers or application servers. The subnet's route table allows inbound and outbound traffic only to and from resources within the subnet.

Most public-facing workloads or SaaS workloads will require the following subnet types:

- Public subnets for hosting public-facing load balancers and NAT services for private subnets
- Private subnets for web servers, application servers, or containers
- Protected subnets for database servers

Subnet Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of subnets:

- Subnets are contained within an AZ.
- IPv4 or IPv6 subnets can be created.
- IPv6-only subnets can be created.
- Subnets host EC2 instances.
- Public subnets have access to the Internet.
- Public subnets are for hosting infrastructure resources such as load balancers or NAT services.
- Private subnets are private, with no direct Internet access, although NAT services can provide indirect Internet access.
- A subnet cannot span multiple availability zones.
- If a subnet's traffic is routed to an Internet gateway, the subnet is a public subnet because there is a defined path to the Internet.
- If a subnet does not have a route to the Internet gateway, the subnet is a private subnet with no external destination defined.
- If a subnet has a route table entry that routes traffic to a virtual private gateway, the subnet is known as a VPN-only subnet, and it can be connected by using an external VPN connection.

IP Address Types

AWS supports both public and private IP address ranges that are assigned to public and private subnets and to the EC2 instances that are hosted on each subnet. Amazon, by default, handles the assigning of IP addresses using DHCP services. This section looks at the IP address options available at AWS, starting with private IPv4 addresses.

Private IPv4 Addresses

When a new EC2 instance is created and launched, by default, AWS assigns a primary private IP address to the default elastic network interface card (eth0) from the range of available IP subnet addresses. Network interfaces attached to EC2 instances are defined at AWS as elastic network interfaces (ENIs). Private IP addresses only communicate across the private network at AWS. A private DNS hostname that points to the associated private IPv4 address is also assigned to each EC2 instance. Private IP addresses are defined as addresses that are not routable over the Internet, regardless of whether they are used at AWS or on premises. If you choose to manually assign a primary private IP address, the private IP address chosen must be available in the subnet IP address range where the EC2 instance will reside. You can assign any private IP address in

the assigned subnet range to an EC2 instance if it is not currently in use or reserved by AWS for its communication needs.

Note

Once a primary private IP address is assigned, the EC2 instance retains the address for the lifetime of the EC2 instance.

Additional (secondary) private IP addresses can also be assigned to ENIs of an EC2 instance, and these addresses can be unassigned and moved to other EC2 instances that reside on the same subnet at any time.

Note

Cost is a factor here. Communication between EC2 instances residing on the same subnet using their private IP addresses is free of charge. However, EC2 instances using private IP addresses located on subnets in different AZs are charged an outbound data transfer fee for communicating with each other.

Private IPv4 Address Summary



A private IP address is assigned by default to every EC2 instance on both public and private subnets. Here are some other criteria to remember:

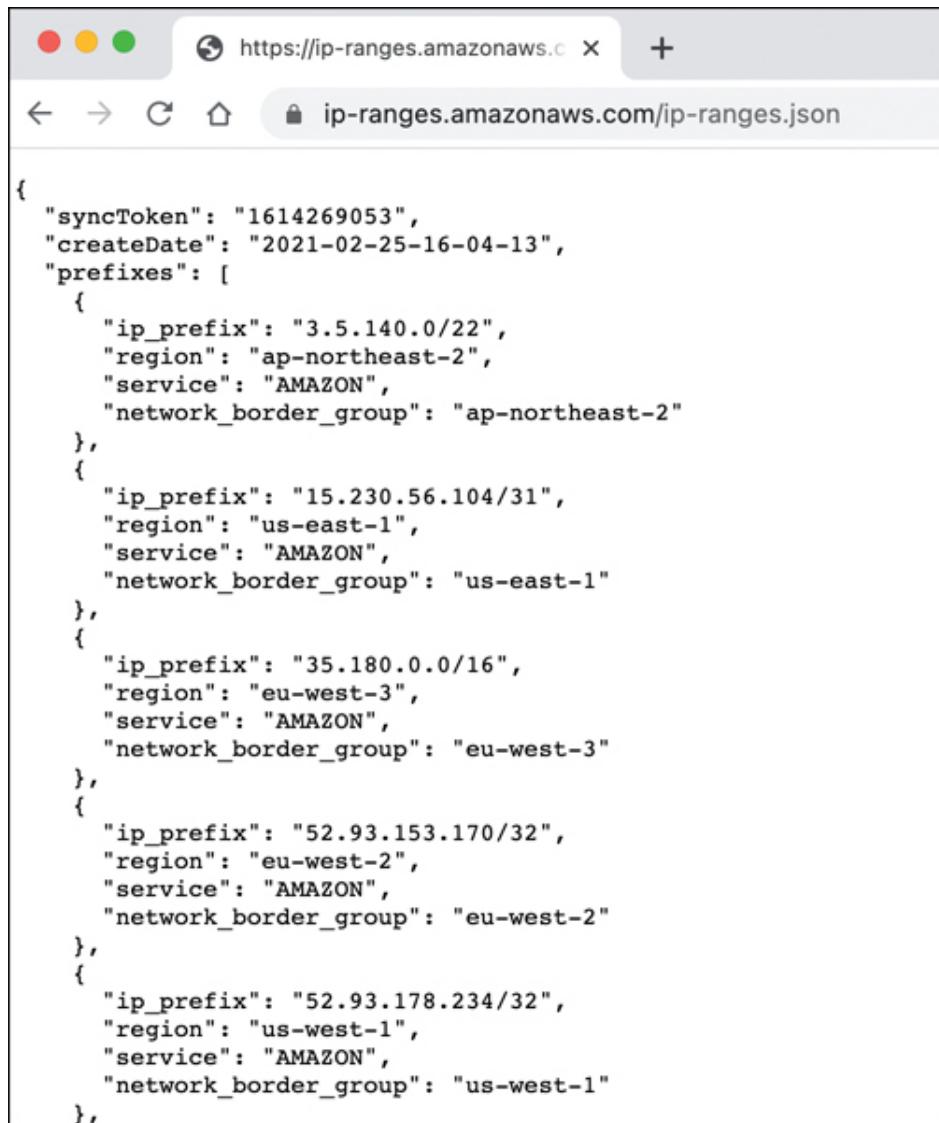
- EC2 instances don't need to use public IP addresses; however, an EC2 instance is always assigned a private IP address.
- A private IP address is assigned for the life of each EC2 instance. The IP address remains attached until the EC2 instance is terminated.

Public IPv4 Addresses

Public IP addresses are used to access resources that have been placed in a public subnet, which is a subnet that is configured to allow inbound and outbound traffic to and from the Internet. A public IP address is typically assigned from AWS's pool of public IP addresses, as shown in [Figure 11-29](#). Public IP addresses from AWS's own pool are managed and controlled by AWS and are therefore not permanently assigned to an EC2 instance.

Instances in a public subnet can be assigned a public IP address

automatically when they are launched. These public IP addresses are dynamic and are associated with the instance until it is stopped or terminated. Whether or not your EC2 instance receives a public IP address during creation is dependent on how the public IP addressing attribute has been defined on the subnet where the EC2 instance is to be hosted. At the subnet attribute level of any subnet you have created, the IPv4 public addressing attribute is initially set to false, which means no public IPv4 address will be assigned to any EC2 instance at creation.

A screenshot of a web browser window displaying a JSON response from the AWS IP ranges API. The URL in the address bar is https://ip-ranges.amazonaws.com/ip-ranges.json. The JSON object contains a single key-value pair where the value is an array of IP prefix objects. Each object has properties: ip_prefix, region, service, and network_border_group. The regions listed are ap-northeast-2, us-east-1, eu-west-3, eu-west-2, and us-west-1.

```
{
  "syncToken": "1614269053",
  "createDate": "2021-02-25-16-04-13",
  "prefixes": [
    {
      "ip_prefix": "3.5.140.0/22",
      "region": "ap-northeast-2",
      "service": "AMAZON",
      "network_border_group": "ap-northeast-2"
    },
    {
      "ip_prefix": "15.230.56.104/31",
      "region": "us-east-1",
      "service": "AMAZON",
      "network_border_group": "us-east-1"
    },
    {
      "ip_prefix": "35.180.0.0/16",
      "region": "eu-west-3",
      "service": "AMAZON",
      "network_border_group": "eu-west-3"
    },
    {
      "ip_prefix": "52.93.153.170/32",
      "region": "eu-west-2",
      "service": "AMAZON",
      "network_border_group": "eu-west-2"
    },
    {
      "ip_prefix": "52.93.178.234/32",
      "region": "us-west-1",
      "service": "AMAZON",
      "network_border_group": "us-west-1"
    }
  ]
}
```

Figure 11-29 The AWS Public IP Address Pool

You can enable the public IP addressing attribute during the creation of an EC2 instance; if you do, an AWS-controlled public IP address is assigned, overriding the default state of the subnet's public IP address attribute.

Elastic IP Addresses

An **Elastic IP (EIP) address** is a static public IPv4 address that is created and assigned to your AWS account and can be easily remapped to any EC2 instance or elastic network interface in your AWS account. They are useful for EC2 instances or services, such as NAT gateway services that need to be reachable from the Internet and require a consistent, static IP address. Requesting an EIP address is simple: Request an EIP address, as shown in [Figure 11-30](#), and it's added to your AWS account from the regional pool of available EIP addresses. EIPs are unassigned; you need to first assign an EIP to the desired VPC and then to the desired EC2 instance.

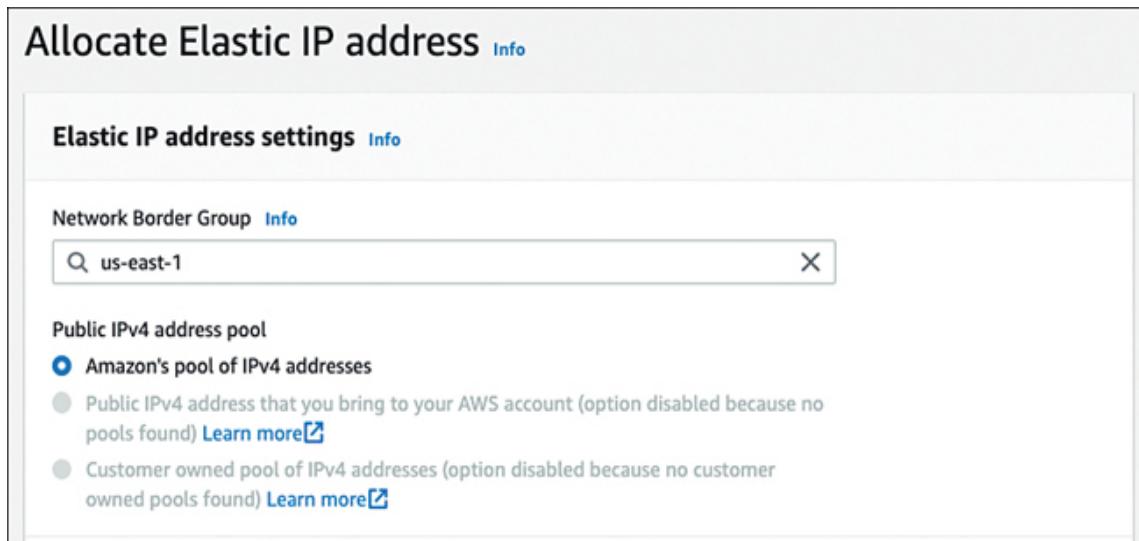


Figure 11-30 Elastic IP Addresses

AWS advertises each regional pool of its EIPs across the public Internet and to all other AWS regions. Because of this advertising, EIPs hosted at AWS can be located across the public Internet and within the public AWS address space.

Note

A public listing of all available AWS EIP addresses is available at

<https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html>.

When an EC2 instance has been assigned an EIP address, turning the EC2 instance off and back on is not an issue. The EIP address remains attached because it's assigned to your AWS account and to the EC2 instance. And there's no additional charge for ordering and assigning a single EIP address to an EC2 instance. However, if you order but don't assign an EIP address, AWS charges you because EIPs are in limited supply.

At AWS, there are four public pools of IP addresses to consider:

- **Dynamically assigned public IPv4 addresses:** Assigned to EC2 instances and returned to the common public pool of

AWS addresses when an EC2 instance shuts down, releasing its dynamically assigned public IPv4 address.

- **Elastic IP addresses:** Elastic IP addresses are static, public IP addresses that are allocated to your AWS account.
- **BYOIP public IPv4 and IPv6 addresses:** Detailed in the upcoming section “[Bring-Your-Own IP \(BYOIP\)](#).”
- **Global unicast IPv6 addresses:** These addresses are unique, globally routable addresses that are assigned to VPCs and subnets. They are used to communicate with resources on the Internet and can be assigned to instances, ELBs, and other resources in a VPC.

The public IPv4 address is displayed as an attribute of the network interface when viewed through the AWS EC2 dashboard, but the internal wiring is a little more complicated. On an EC2 instance with a public IP address, this address is internally mapped to the EC2 instance’s primary private IPv4 address using NAT services. When a public IP address is assigned to an EC2 instance, the inbound traffic is directed to your EC2 instance’s private internal IP address.

If your EC2 instance is directly accessible from the Internet, when someone wants to directly reach your EC2 instance, the inbound destination is the public IP address. When the EC2 instance needs to communicate outbound across the Internet,

the source address is its public IP address. Queries on the private network of the EC2 instance always use the private address of the EC2 instance. The takeaway from this example is that AWS attempts to use the private IP address, whenever possible, for network communication with an EC2 instance.

Each EC2 instance that receives a public IP address at AWS is also provided with an external DNS hostname. As a result, the external DNS hostname is resolved to the public IP address of the EC2 instance when queries are external to AWS, as shown in [**Figure 11-31**](#).

Instance summary for i-0ecfa43f140660754 Info		Instance state ▾
Updated less than a minute ago		C Connect
Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ecfa43f140660754	3.227.57.103 open address	172.30.0.198
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-3-227-57-103.compute-1.amazonaws.com open address	ip-172-30-0-198.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	3.227.57.103 [Public IP]	vpc-c753f9a2 (default VPC)
AWS Compute Optimizer finding	IAM Role	Subnet ID
Opt-in to AWS Compute Optimizer for recommendations. Learn more	-	subnet-7c6dd651 (Public Subnet)

Figure 11-31 Assigned IP Addresses and DNS Hostnames

Public IPv4 Address Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of public IPv4 addresses:

- You can use EIP addresses for NAT EC2 instances or custom public-facing appliances.
- You can use EIP addresses for public-facing load balancers.
- You can use EIP addresses for public-facing EC2 instances.
- You must use EIP addresses for the NAT gateway services.
- Public IP addresses are not necessary for EC2 instances.

Inbound and Outbound Traffic Charges

It's important to realize that you are billed differently for public traffic and private traffic at AWS. Your AWS outbound traffic costs can become extremely high if you don't pay attention. For example, you will be charged more for public traffic sent to a public IP address traveling across the Internet than for private IP address traffic. Private traffic traveling within AWS data centers is always cheaper than traffic on a public subnet (see [Figure 11-32](#)); therefore, whenever possible, AWS uses the private network for communication.

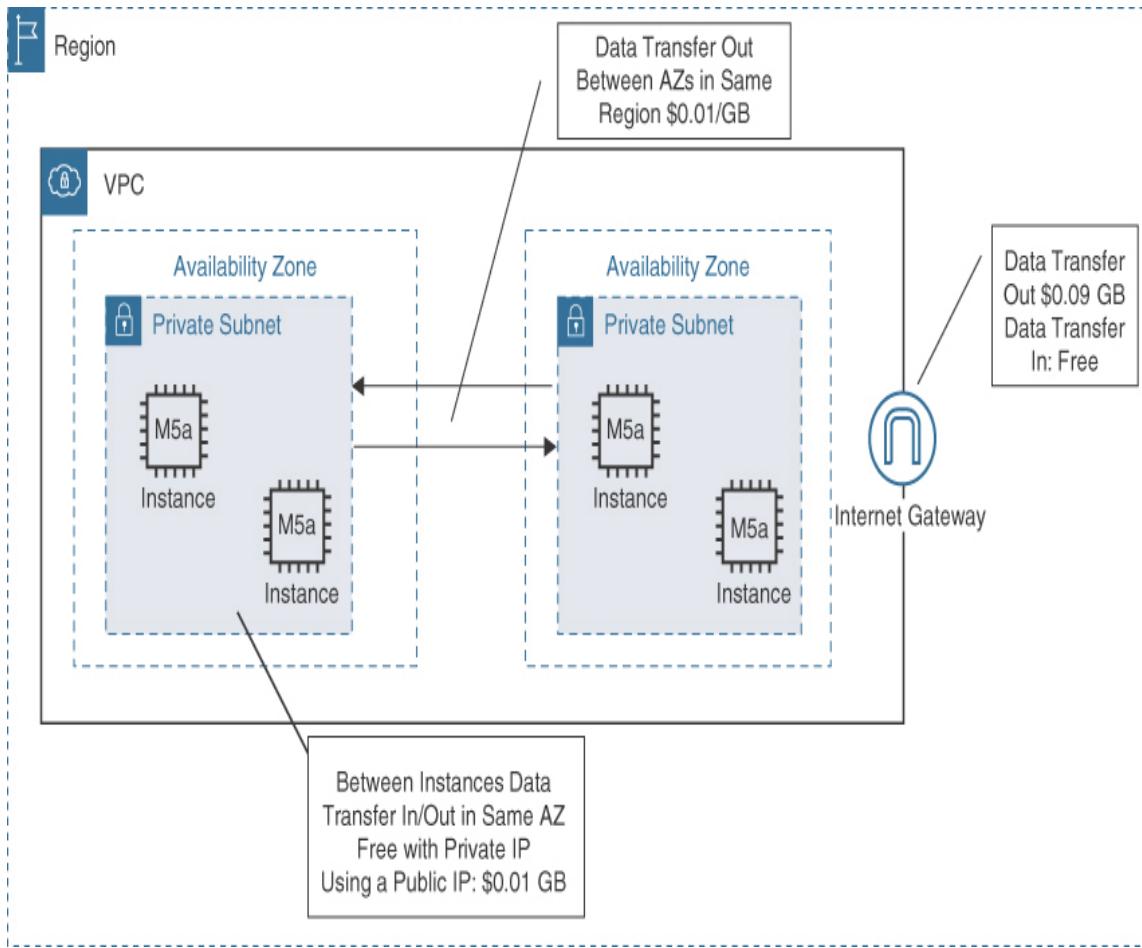


Figure 11-32 Traffic Charges at AWS

Private traffic that stays within a single subnet incurs no additional charges, whereas private traffic that travels across multiple private subnets that are hosted by different AZs incurs an egress charge of \$.01 per gigabyte. However, RDS database replication across separate availability zones is free for the synchronous replication of your data from the primary database EC2 instance to the secondary database EC2 instance. One task to carry out is a detailed cost optimization of your

production workload traffic patterns, including the costs for replication across AZs, load balancer traffic, and outbound traffic flow.

Note

All inbound communication traffic that an EC2 instance receives is free, regardless of whether it comes from inside AWS or from the public Internet. However, EC2 instance replication traffic across multiple AZs, with the exception of RDS deployments, is charged a data transfer fee.

Bring-Your-Own IP

Bring-Your-Own IP (BYOIP) is a feature offered by AWS that enables you to bring your own public IPv4 or IPv6 addresses to AWS and use them with your AWS resources. This can be useful if you want to use a specific range of IP addresses for your resources or if you want to migrate your existing IP addresses to AWS. If you own a publicly routable public IPv4 or IPv6 address range, you can move part or all of a public IP address from your on-premises network to AWS. Each organization still owns their public IP range; however, AWS hosts and advertises the public IP address range hosted at AWS across the Internet

and AWS regions for you. The public address range must be registered with your Regional Internet Registry (RIR)—for example, the American Registry for Internet Numbers (ARIN)—and must also be registered to a business or institution, not to an individual person. Bringing your own public IPv4 or IPv6 address space to AWS allows you to accomplish the following:

- Maintain your public IP address reputation.
- Avoid any changes to public IP addresses that have been whitelisted.
- Avoid changing IP addresses that legacy applications still use.
- Use a public IP address as a hot standby failover for on-premises resources.

The following are some examples of situations in which you might want to control your own public address space in the AWS cloud:

- You want to keep a recognizable public IP address but have the service assigned to that address hosted on AWS.
- You have 10,000 hard-coded lottery machines, and you want to change the hardware devices to virtual ones at AWS with your public IP addresses.
- You have 2,000 hard-coded public IP addresses within your data center, and you want to change the physical location of

your data center to AWS but keep the same public IP addresses.

- You have legacy workloads—or older applications that rely on specific fixed public IP addresses—and want to move these addresses to AWS.

Note

The specific prefix supported by BYOIP at AWS for IPv4 public addresses is /24. The specific prefix for IPv6 addresses is /48 for CIDRs that are publicly advertised, and /56 for CIDRs that are not publicly advertised.

The BYOIP Process

These are the basic steps for allocating BYOIP addresses to AWS:

Step 1. Import the public IP address, or the range of public IP addresses, into AWS. AWS creates a pool of these addresses and assigns the address pool to you.

After AWS has analyzed and accepted the range of public IP addresses, the state of the public address range to be hosted at AWS changes to “provisioned,” indicating that the IP address request has been accepted. At this point, you can use these

public IP addresses, but they have not yet been advertised across the Internet or to the peering partners of AWS.

Step 2. Advertise the public address range to all the peering partners of AWS. When the advertising process has been accepted and started at AWS, it's time to stop the advertising of the same public IP addresses to avoid any strange duplication routing conflicts.

Step 3. Allocate EIP addresses from your AWS-hosted pool of public IP addresses.

When using the hosted pool of addresses at AWS to allocate EIP addresses, you can select a random IP address from the hosted pool or select a specific IP address.

If in the future you decide you don't want AWS to advertise and host your pool of public IP addresses, you can execute a "withdraw" command to change the state of the public IP addresses from advertised back to an unadvertised state. At this point, AWS no longer advertises your public IP addresses. The last step is to run the deprovisioning command to remove the assigned EIP addresses.

IPv6 Addresses

**Key
Topic**

Even though IPv6 addresses are fully supported within a VPC, an IPv4 CIDR block must be created first. The allowable format for IPv6 addresses is 128 bits, with a fixed CIDR block size of /56. Amazon is in control of IPv6 addressing at AWS; you cannot select your own IPv6 CIDR range. At AWS, IPv6 addresses are globally unique addresses and can be configured to remain private or reachable across the Internet.

Amazon VPC has built-in support for address assignment via DHCP for both IPv4 and IPv6 addresses. If your EC2 instance is configured to receive an IPv6 address at launch, the address will be associated with the primary network interface (eth0). Assigned IPv6 addresses are also persistent; you can stop and start your EC2 instance, and the IPv6 addresses remain assigned. Access to the Internet using an IPv6 address can be controlled by using the egress-only Internet gateway (EOIG), route tables, and, optionally, network access controls. Here is a short summary of the steps for providing IPv6 Internet access:

Step 1. Associate the AWS-provided IPv6 CIDR block with your VPC.

Step 2. Create and attach an Internet gateway to the VPC and add a route table entry to the subnet that will communicate with the IGW.

Step 3. Create an egress-only Internet gateway. This allows your private subnet to enable outbound communications to the Internet using IPv6; the EOIG allows outbound communication and prevents any inbound communication.

Step 4. Update your route tables to route your IPv6 traffic to the EOIG:

- **For EC2 instances hosted on IPv6 public subnets:** Add a route that directs all IPv6 traffic from the subnet to the Internet gateway. Note that this is the regular Internet gateway and not the EOIG; the EOIG is controlling private outbound communication from the private subnet and stopping any inbound connections from the Internet.
- **For EC2 instances on IPv6 private subnets:** Create a route that directs all Internet-bound IPv6 traffic to the EOIG.

Step 5. Review and, if necessary, update your network access controls.

Note

EC2 instances launched in IPv6-only subnets and ENIs attached to them are assigned IPv6 addresses through the DHCPv6 options set from the IPv6 CIDR block of your subnet. When EC2 instances are launched in IPv6-only subnets, resource-based naming (RBN) is used automatically; the EC2 instance ID is included in the hostname of the instance.

VPC Flow Logs



Network traffic can be captured for analysis or to diagnose communication problems at the level of the elastic network interface, subnet, or entire VPC. AWS does not charge you for creating a flow log, but it will impose charges for data storage. When each flow log is created, you define the type of traffic that will be captured—either accepted traffic only, rejected traffic only, or all traffic, as shown in [Figure 11-33](#).

Flow log settings

Name - *optional*

Filter
The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).
 Accept
 Reject
 All

Maximum aggregation interval [Info](#)
The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.
 10 minutes
 1 minute

Destination
The destination to which to publish the flow log data.
 Send to CloudWatch Logs
 Send to an Amazon S3 bucket

Figure 11-33 Flow Log Storage Location Choices

Flow logs can be stored either in a CloudWatch log group or directly in an Amazon S3 bucket for storage, also shown in [Figure 11-33](#). If VPC flow logs are stored as a CloudWatch log group, IAM roles must be created that define the permissions for the CloudWatch monitoring service to publish the flow log data to the CloudWatch log group. Once the log group has been created, you can publish multiple flow logs to the same log group.

Creating a flow log for a subnet or a VPC, each network interface present in the VPC, or subnet is then monitored.

Launching additional EC2 instances into a subnet with an attached flow log results in new log streams for each new network interface and any network traffic flows.

Not all network traffic is logged in a flow log. Examples of traffic that is not logged in flow logs include AWS DNS server traffic, Windows license activation traffic, instant metadata requests, Amazon Time Sync Service traffic, reserved IP address traffic, DHCP traffic, and traffic across a PrivateLink interface.

Any AWS service that uses EC2 instances with network interfaces can take advantage of flow logs. Supporting services also include ELB, Amazon RDS, Amazon ElastiCache, Amazon Redshift, Amazon EMR, and Amazon WorkSpaces. Each of these services is hosted on an EC2 instance with network interfaces.

Connectivity Options

There are several methods of connecting resources in VPCs across the AWS private network, as summarized in [Table 11-6](#).

Key Topic

Table 11-6 VPC Private Connectivity Options

Option	Details
Peering VPC	Connect two VPCs together with a private network connection
Gateway endpoints	Connect a VPC to S3 buckets or a DynamoDB table across the AWS private network
Interface endpoints	Connect a VPC to most AWS services across the AWS private network
Transit gateway	The <i>transit gateway</i> is a network transit hub used to interconnect VPCs and on-premises networks privately. Traffic is encrypted automatically. Connect VPCs, Direct Connect gateways, VPN connections, and peering connections.

VPC Peering

VPC peering enables you to connect two VPCs in the same or different regions and communicate with each other as if they were on the same network, using private IP addresses. It's quite common to find that a single company has many AWS accounts and multiple VPCs. This can be a management nightmare, especially if separate AWS accounts and separate VPCs might need to be connected to share resources or common services, such as monitoring or authentication. Thankfully, you can create networking connections between VPCs through a process called *peering*, which enables you to route traffic between two VPCs that have been peered together. Route tables, security groups, and NACLs control which subnets or EC2 instances are able to connect using the peered connection with an AWS region.

A ***peering connection*** is not the same as a gateway connection or a VPN connection. Instead, peering is set up by first sending an invitation from one VPC to another VPC; the invitation must be accepted before the peering connection is established. Peering within the same AWS account involves using each VPC's ID for identification. Peering VPCs between different AWS accounts requires both AWS account IDs and the VPC IDs.

Peering occurs between a VPC in one AWS account or between a VPC in another AWS account. Peered VPCs can also reside in

completely different AWS regions. Data traffic between VPCs peered in different regions is encrypted using *AEAD encryption*, which uses the Authenticated Encryption with Associated Data protocol. With AEAD encryption, AWS manages the entire encryption process and supplies and rotates the encryption keys.

Establishing a Peering Connection

Key Topic

The VPC that starts the peering process is called the *requester VPC*; it defines the owner of the VPC that would like to establish a peering connection. The *accepter VPC* is the VPC and account owner that needs to accept the request to establish a peer (see [Figure 11-34](#)). Here are the basic steps involved in peering:

Step 1. The owner of the requester VPC sends a request to the owner of the accepter VPC.

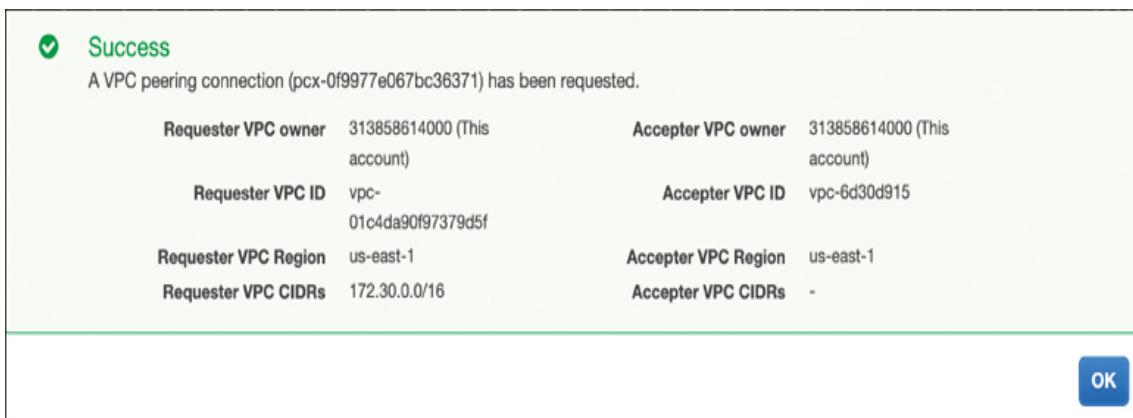


Figure 11-34 Peering Traffic Flow

Step 2. The owner of the accepter VPC accepts the VPC peering connection request.

Step 3. The peering connection is activated.

Step 4. Security group rules are updated within each VPC to ensure proper traffic routing to and from the VPCs that have been peered together.

Step 5. Route tables are updated with entries to allow the flow of traffic to enable communications between the two VPCs.

The accepter VPC might be a VPC that is in your AWS account and therefore one of your own, or it could be another AWS account's VPC. This relationship flexibility is important because single companies can have many developers with VPCs created within their AWS accounts. A VPC could also be from a third-

party service provider that has developed an application that's entirely hosted in a separate private VPC infrastructure, such as a monitoring service or a disaster recovery service.

The following are some additional considerations for VPC peering:

- VPC peering connections cannot be created between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks.
- More than one VPC peering connection between the same two VPCs is not allowed.

The following are some inter-region VPC peering limitations to be aware of:

- Public IPv4 DNS hostnames cannot be resolved from EC2 instances on one side of the peered connection to a private IPv4 address on the other side of the peered connection.
- VPC peering supports both IPv4 and IPv6 addresses, enabling you to connect VPCs that use either protocol.
- The maximum transmission unit (MTU) across the VPC peering connection is 1500 bytes; jumbo frames are not supported.
- Security group rules cannot reference a VPC security group across a peered connection; directing outbound traffic from

one side of a peered connection to a security group on the other side of the peered connection is not allowed. A VPN or a Direct Connect connection to a corporate network across a peered connection is not allowed.

- An Internet connection from a private subnet to a NAT device in a public subnet cannot travel across a peered connection to an AWS service such as DynamoDB or S3. For example, VPC A is connected to the corporate network using a Direct Connect connection. Users at the head office cannot connect through the Direct Connect connection to VPC A and across the peered connection to VPC B.
- A VPC peering connection is always a one-to-one relationship between two VPCs. Transitive peering relationships are not supported.
- A VPC can have multiple peering relationships with other VPCs, but each peering relationship is always a direct, one-to-one relationship.

VPC Endpoints



Endpoints enable you to create a private connection between your VPC and another AWS service without the need for an

Internet gateway, a NAT gateway, or a VPN connection. VPC endpoints are used to enable communication among resources in your VPC and other AWS services using private IP addresses, which can improve the security and performance of your applications. The majority of endpoint connections are interface VPC endpoints; however, there is a gateway endpoint for accessing Amazon DynamoDB.

VPC Gateway Endpoints

VPC Gateway endpoints are attached to your VPC and use a route in the associated subnet's route table to connect to the target service. They can be used to connect to Amazon S3 and DynamoDB, as shown in [Figure 11-35](#). VPC Gateway endpoints are not charged for creation and data transfer.

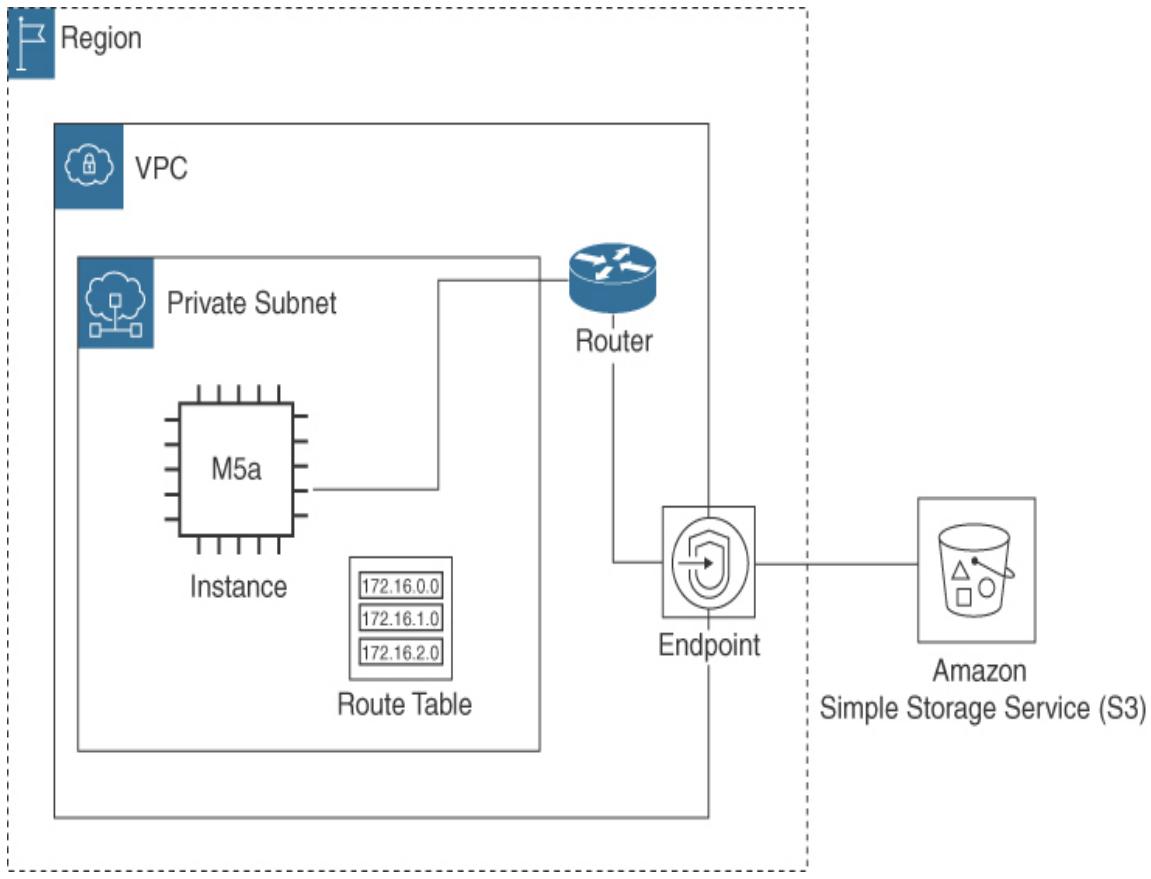


Figure 11-35 Gateway Endpoint Access to S3

To create a gateway endpoint, you need to follow these steps:

Step 1. From the VPC Dashboard, select the Endpoints tab on the left and click Create Endpoint.

Step 2. Select the gateway endpoint for DynamoDB.

Step 3. Select the VPC and subnets where access is required.

Step 4. Modify the default endpoint policy to match your security needs.

Step 5. Update the security groups and network ACLs as necessary.

Endpoint policies can be deployed to further define the endpoint access rules. The default policy allows full access to the service; this default policy should be evaluated and changed if necessary. Custom endpoint policies control access from the VPC through the gateway endpoint to the service from the EC2 instance.

VPC Interface Endpoints

The newest form of a VPC endpoint supported by AWS is an *interface endpoint* powered by a technology called PrivateLink. The “interface” is a network adapter designated with a private IP address. AWS services are not all accessible through interface endpoint connections; however, many AWS services are accessible through private interface endpoints.

AWS resources with an interface connection are accessed using a private IP address from the selected VPC subnet. If AWS resources are connected with multiple subnets, multiple private IP addresses will be used—one IP address per availability zone

subnet. If you are using AWS Direct Connect to link your on-premises data center with AWS resources, there is also a bonus: You can access AWS-hosted data records and AWS services from your on-premises network.

For example, a developer is sitting in an office, working on developing applications using the AWS Cloud9 IDE. The developer accesses the development portal privately across the high-speed fiber Direct Connect connection, and VPC interface connection shown in [Figure 11-36](#). When the application is finished and deployed in production, the application can continue to be accessed privately through the Direct Connect connection from the head office.

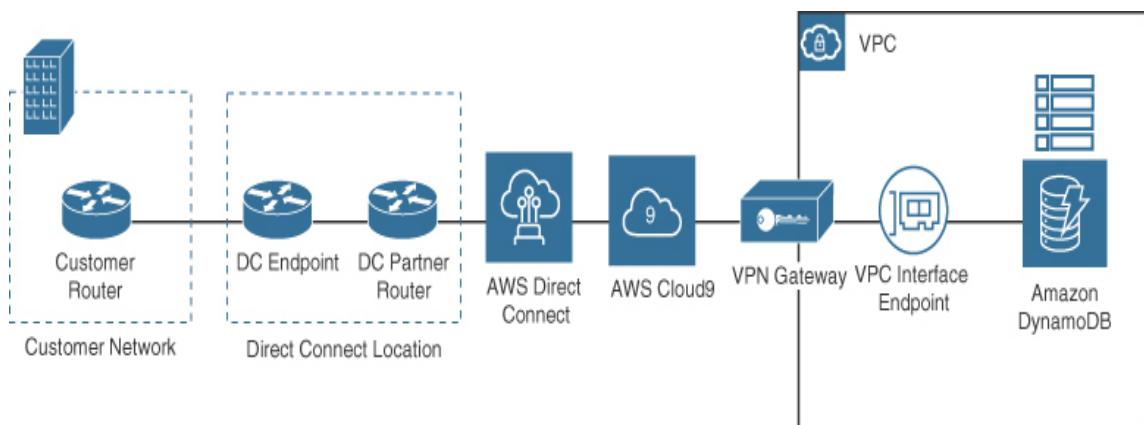


Figure 11-36 Using Interface VPC Endpoints

Many large corporations considering a move to the cloud remain cautious of having corporate data records stored in the

cloud. For these situations, endpoint connections combined with high-speed 100-Gbps Direct Connect connections deliver speed, security, and AWS services across a totally private environment.

With no public connectivity, the AWS services that are being accessed using an interface or gateway endpoint are fully protected from any Internet-facing attacks, including DDoS attacks, because the private interface endpoints simply cannot be reached from the Internet. When you create an endpoint inside your VPC, service names are protected; Route 53 DNS services send you to the private endpoint location and ignore any public routes that also may be advertised. Private endpoints also have regional zonal names designed for keeping traffic within the region, allowing customers to isolate traffic, if necessary, to a specific AZ. These zonal endpoints could also potentially save you additional data transfer charges and latency issues.

The hardware powering interface endpoints is publicly called PrivateLink, but internally, AWS calls this network hardware Hyperplane. Hyperplane is a massively scalable, fault-tolerant distributed system that is designed for managing VPC network connections. It resides in the fabric of the VPC networking layer, where AWS's software-defined networking is deployed; it

can make transactional decisions in microseconds. When a VPC interface endpoint is created, it is associated with several virtual Hyperplane nodes that intercept network communications at the VPC networking layer and quickly decide what to do with each request. If a request is made to a private endpoint, the transactional decision and the shared state are applied in milliseconds. Interface VPC endpoints only accept TCP traffic, and each endpoint supports a bandwidth of up to 10 Gbps per availability zone. It also automatically scales up to 100Gbps.

Endpoint Services



Using the PrivateLink technology, AWS hopes to help provide private SaaS services to corporations that are currently using AWS services, as shown in [Figure 11-37](#). The owner of a private SaaS service hosted at AWS is called a *service provider*, and the owner of an interface endpoint is called a *service consumer* because it is the consumer of the service. Private SaaS services could include monitoring services and disaster recovery services.

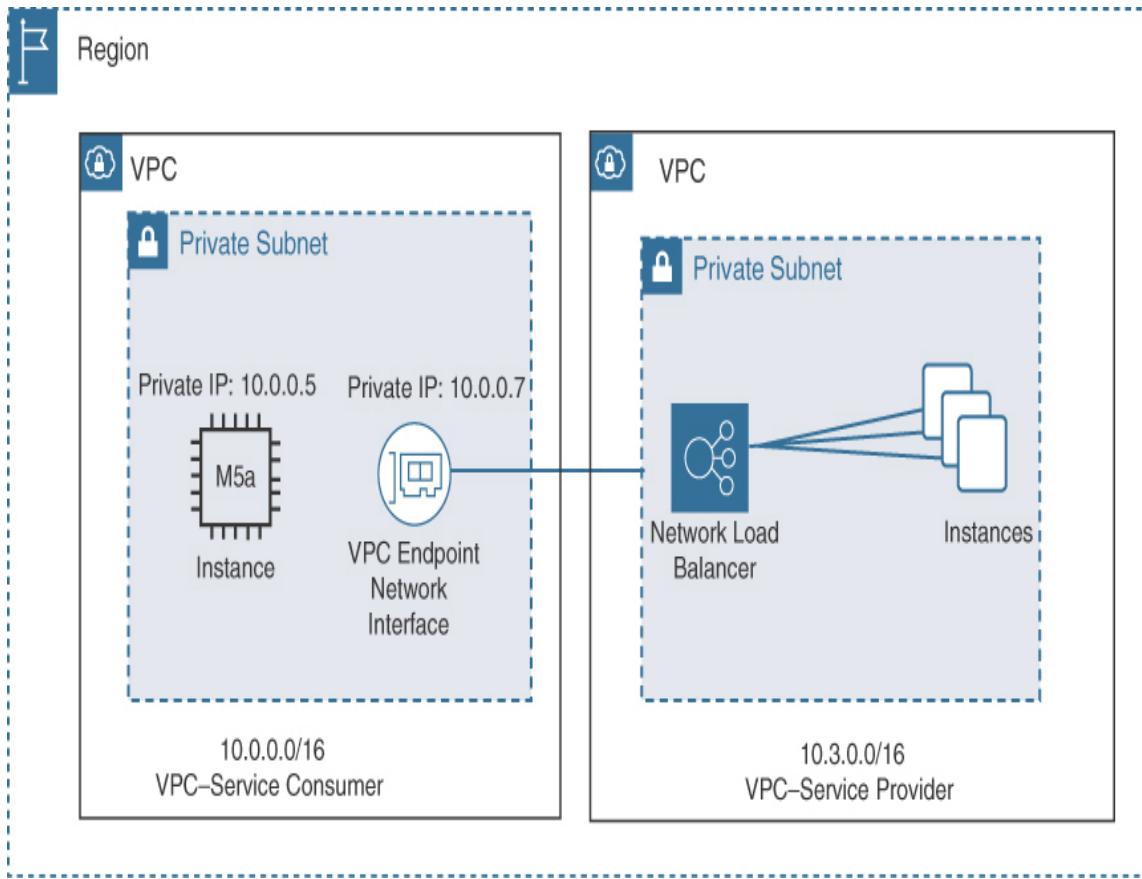


Figure 11-37 PrivateLink Endpoints

A customer who wants to access the third-party SaaS service creates a VPC gateway or interface endpoint connecting to the service provider's endpoint service. To handle network access to the subscribed service, behind the “interface” connection is a private NLB positioned at the entrance to the hosted SaaS service.

Third-party microservice architectures could also be hosted within the third party's private VPC. The service provider VPC

can follow the same best practices as recommended by AWS for creating fault-tolerant applications hosted in a VPC. Amazon uses this process to provide network load-balancing services to multiple customers within each region. Applications can be designed with availability targets located across multiple AZs.

Depending on the tenancy requirements of the customer, for a single-tenant mode of operation, a private NLB could be created for every client customer. Multi-tenant designs could allow multiple customers to use the same NLB service. There are several additional choices available to separate endpoint traffic from VPCs in a multi-tenant design:

- Use separate account/password security tokens at the application level.
- Use separate NLBs and different listener ports.
- Use the Proxy Protocol V2 preamble, which adds a header to each connection that lists the ID of the destination endpoint.

Note

The costs for PrivateLink are split between the provider and the customer. The provider side pays for the NLB costs. The client side pays for the PrivateLink endpoint costs.

The steps for creating a PrivateLink interface endpoint are as follows:

Step 1. From the VPC Dashboard, select Endpoint from the menu and click Create Endpoint.

Step 2. Select the PrivateLink Ready partner service.

Step 3. Select the VPC and subnets where access is required.

Step 4. Select Enable Private DNS Name, if required.

Step 5. Select Security Group.

Step 6. Update route tables, security groups, and network ACLs as necessary.

Exam Preparation Tasks

As mentioned in the section “[How to Use This Book](#)” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. [Table 11-7](#) lists these key topics and the page number on which each is found.



Table 11-7 [Chapter 11](#) Key Topics

Key Topic Element	Description	Page
Section	CloudFront Use Cases	529
Section	Serving Private Content	530
Section	Using an Origin Access Identifier	531
Section	CloudFront Origin Failover	532
Section	Edge Functions	534
Section	CloudFront Cheat Sheet	536

Key Topic Element	Description	Page
List	Standard and custom accelerators	537
Section	Rules, Conditions, and Actions	545
Section	Target Groups	547
Section	Target Group Attributes	550
Section	ALB Cheat Sheet	553
List	NLB features	554
Section	NLB Cheat Sheet	554
Section	The Shared Security Model	557
Section	VPC Cheat Sheet	560
<u>Figure 11-24</u>	VPC Starting Design Choices	563

Key Topic Element	Description	Page
Section	Adding a Secondary CIDR Block	568
Section	Subnet Cheat Sheet	572
Section	Private IPv4 Address Summary	574
Section	IPv6 Addresses	580
Section	VPC Flow Logs	581
<u>Table 11-6</u>	VPC Private Connectivity Options	583
Section	Establishing a Peering Connection	584
Section	VPC Endpoints	585

Key Topic Element	Description	Page
Section	Endpoint Services	588

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Amazon CloudFront

regional edge cache

origin access identity (OAI)

origin failover

Lambda@Edge

listener

connection draining

target group

health check

sticky session

virtual private cloud (VPC)

network access control list (NACL)

NAT gateway service

egress-only Internet gateway (EOIG)

Internet gateway (IG)

subnet

Elastic IP (EIP) address

peering connection

endpoint

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep Software Online.

- 1.** How does deploying CloudFront help globalize your data records?
- 2.** Can you manage third-party load balancers with an ELB service?
- 3.** What CIDR address range should you use for your VPC?
- 4.** Why would organizations require more than one VPC?
- 5.** Why should you avoid using public subnets for your web servers?
- 6.** How can you move your existing public IP addresses to AWS?
- 7.** What AWS networking services can replace existing hardware devices?
- 8.** What can network ACLs do that a security group cannot do?
- 9.** Why would you use Elastic IP addresses?
- 10.** How does deploying endpoints help secure VPC hosted workloads?

Chapter 12

Designing Cost-Optimized Storage Solutions

This chapter covers the following topics:

- [Calculating AWS Costs](#)
- [Cost Management Tools](#)
- [Storage Types and Costs](#)
- [AWS Backup](#)
- [Data Transfer Options](#)
- [AWS Storage Gateway](#)

This chapter covers content that's important to the following exam domain and task statement:

Domain 4: Design Cost-Optimized Architectures

Task Statement 1: Design cost-optimized storage solutions

Hosting applications at AWS is supposed to save you money, or so goes the common assumption. Moving to the cloud can help you save money, but you need to review AWS's pricing structure in depth to understand how AWS charges for using its cloud services. Pricing at AWS is complicated; someday, there

will probably be a university degree focusing on AWS pricing. The AWS Certified Solutions Architect – Associate (SAA-C03) exam has been revised to include an expanded knowledge domain on costs. You must understand how to manage storage costs.

Recall from [Chapter 2, “The AWS Well-Architected Framework,”](#) that Cost Optimization is one of the six pillars of the AWS Well-Architected Framework. The AWS document “Cost Optimization Pillar” (see

<https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/wellarchitected-cost-optimization-pillar.pdf>) provides additional details to help you understand how to manage costs at AWS. Additionally, there are cost-optimization labs at <https://wellarchitectedlabs.com/cost/> to assist you in developing hands-on skills in the management of AWS cloud costs. To help study for the SAA-C03 exam, make sure to sign up for an AWS Free Tier account; most of the Well-Architected Labs can be completed at little to no cost (<https://aws.amazon.com/free>).

“Do I Know This Already?”

The “Do I Know This Already?” quiz enables you to assess whether you should read this entire chapter thoroughly or

jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 12-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A](#), “[Answers to the ‘Do I Know This Already?’ Quizzes and Q&A Sections.](#)”

Table 12-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Calculating AWS Costs	1, 2
Cost Management Tools	3, 4
Storage Types and Costs	5, 6
AWS Backup	7, 8
Data Transfer Options	9, 10
AWS Storage Gateway	11, 12

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment.

Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. How are costs charged at AWS?

1. Per AZ
2. Per AWS region
3. Per report
4. Included in the price

2. How are Elastic Load Balancing (ELB) charges calculated?

1. Number of connections
2. Inbound traffic levels
3. Load balancer capacity units
4. Amount of outgoing data transferred

3. How do you access the cost management tools at AWS?

1. AWS Cost Explorer
2. AWS Billing Dashboard
3. AWS Budgets
4. AWS Cost and Usage Reports

4. How is the tracking of your AWS costs initiated?

1. AWS Budgets
2. AWS Cost and Usage Reports
3. Enable data collection with Cost Explorer
4. Enable rightsizing recommendations in Cost Explorer

5. How are Amazon S3 storage costs calculated ?

1. By gigabytes per month stored
2. Tiered pricing
3. By storage class
4. Based on AWS region

6. What are the most expensive Amazon EBS volume types?

1. SSD volumes
2. Throughput-optimized volumes
3. PIOPS volumes

4. Cold storage volumes

7. What two storage tiers are maintained by AWS Backup?

1. Hot and cold tiers
2. Standard and infrequent access tiers
3. On-demand and continuous tiers
4. Warm and cold storage tiers

8. What AWS Backup plan feature manages warm and cold storage tiers?

1. Backup window
2. Tags
3. Lifecycle rules
4. Regional copies

9. What service automates the movement of data from on-premises locations to either Amazon S3 buckets or EFS storage?

1. AWS SFTP
2. AWS FTP
3. AWS Snow Family
4. AWS DataSync

10. What is the smallest member of the AWS Snow Family?

1. Snowball
2. Snowmobile
3. Snowcone
4. Snowball Edge

11. Which AWS Storage Gateway storage option supports virtual tape backups?

1. Tape Gateway
2. File Gateway
3. Volume Gateway
4. Amazon S3 Glacier gateway

12. What type of storage is supported by AWS Storage Gateway?

1. Cloud storage
2. Amazon S3 storage
3. Amazon FSx for Windows File Server storage
4. Hybrid storage

Foundation Topics

Calculating AWS Costs

AWS costs are based on what is called a “consumption model,” where customers pay for the cloud services ordered per AWS

account by the second for compute resources and monthly for storage resources. The price of each AWS service is different in each AWS region. For example, choosing the Central AWS region means the AWS resources are located in Canada and, depending on the currency exchange rate of your country, pricing in Canada may be more or less than pricing in your home country. If you are restricted to operating in the European Union (EU) or South America (São Paulo), you can expect pricing to be a few hundred percent more than in Canada! Keep in mind, however, that you might not have the option to operate in a cheaper AWS region if compliance requirements dictate where your organization is allowed to operate.

It's just a fact that costs are higher in some areas of the world for all public cloud providers. The biggest and cheapest AWS region to operate in is us-east-1 (Northern Virginia), which also has the largest number of availability zones (AZs) and AWS cloud services. Other AWS regions with comparable pricing to Northern Virginia include Ohio, located in us-east-2, and Ireland (eu).

The purchase price of compute resources can be significantly reduced in many cases by purchasing Reserved Instances. For example, a human resources system deployed at AWS requires

an average of ten t2.medium EC2 compute instances placed behind an ELB Application Load Balancer. Although there are many variables when analyzing pricing, there are some general trends, such as widely differing regional costs. For example, as shown in [Figure 12-1](#), comparing the us-east Northern Virginia region costs to the São Paulo region costs for a load balancer results in an 80% price difference.

Estimate of Your Monthly Bill		
<input type="checkbox"/> Show First Month's Bill (Include all one-time fees, if any)		
 Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.		
Export to CSV		Save and Share
<input type="checkbox"/> Amazon EC2 Service (US East (N. Virginia))	Compute:	\$ 339.70
<input checked="" type="checkbox"/> Amazon EC2 Service (South America (Sao Paulo))		\$ 3935.30
<input type="checkbox"/> Amazon Elastic Load Balancing (US East (N. Virginia))	Application LBs:	\$ 16.47
<input type="checkbox"/> Amazon Elastic Load Balancing (South America (Sao Paulo))		\$ 24.89

Figure 12-1 Comparing Regional Prices

Note

Many costs are included in the price of AWS services. For example, there is no charge to spin up a VPC and add subnets.

Cloud Service Costs

Using a managed AWS cloud service for storage, database deployments, or monitoring removes the cost of managing, maintaining, updating, and administering a customer-built service. For example, when storing objects in S3 buckets, customers create buckets and manage their stored objects, but they don't have to maintain and update the S3 storage array. Overall, costs are reduced when you use AWS cloud services as part of your workload.

How AWS charges for cloud services that are used can be broken down by component, as shown in [Table 12-2](#). This breakdown of costs is the same in every AWS region.

Key Topic

Table 12-2 Cloud Service Charges at AWS

Service	Charges	Components	Examples
---------	---------	------------	----------

Service Management tools	Charges Number of times service performs selected tasks; tiered pricing after the minimum is reached	Components Compute, storage, notification, and automation charges	Examples AWS Config AWS Lambda AWS CloudWatch Amazon GuardDuty, Amazon Macie
Compute	By the hour or second	Compute hours plus <u>data transfer</u> charges	EC2 instances, ECS, RDS

Service Storage	Charges Monthly	Components Storage,	Examples Amazon EB
	<p>per gigabyte stored; tiered pricing after the minimum is reached</p>	<p>data transfer, and, optionally, encryption or notification and automation charges</p>	<p>S3 Glacier, EFS, FSx, EI snapshots, server logs</p>
Amazon CloudWatch	<p>By alarm, metrics, and analysis; tiered pricing after minimum is reached</p>	<p>Compute, storage, notification, and automation charges</p>	<p>Alerts/Alarms, CloudWatch logs, Events/Rules</p>

Service	Charges Per	Components	Examples
Amazon SNS	notification; tiered pricing after minimum quota is reached	Compute, storage, notification, and automation charges	Notification
Data transfer	Per gigabyte transferred	Data transfer out	Across AZs, regions, edge locations outgoing da



Tiered Pricing at AWS



Many AWS management services and data transfer costs offer **tiered pricing**. With tiered pricing, a customer uses a particular AWS service until their usage of the service exceeds a

defined default value; then, the customer's costs scale based on a sliding price point. The more usage, the smaller the charges. After usage increases to the next tiered level, AWS provides another price break until the next checkpoint. For example, charges for data transfer out from EC2 instances to the Internet start at \$0.09/gigabyte until you reach 10 TiB. Then, additional discounts apply, as shown in [Table 12-3](#). The charge for region-to-region traffic is currently set at \$0.02/gigabyte, but in the case of us-east-1 and us-east-2, it's \$0.01/gigabyte. The prices presented here may change by the time you read this, because AWS changes prices from time to time.

Table 12-3 Tiered Storage Costs for Data Transfer Out from EC2 Instance to the Internet

Storage Amount	Price per Month
Up to 100 GiB	Free
Up to 9.999 TiB	\$0.09 per gigabyte
Up to 40 TiB	\$0.085 per gigabyte
Up to 100 TiB	\$0.07 per gigabyte

Storage Amount	Price per Month
Greater than 150 TiB	\$0.05 per gigabyte
Greater than 500 TiB	Call AWS

A typical web application hosted at AWS will also most likely be using the following services. Each service has specific charges based on its operation.

- **Amazon CloudFront:** The first 1 TiB of data transfer, the first 10 million HTTP/S requests, and the first 2 million CloudFront Functions invocations are free.
- **Amazon CloudWatch:** Up to the first 10,000 custom metrics are charged \$0.30 per month. The next 240,000 metrics are charged at \$0.10 per month.
- **AWS CloudTrail:** The first copy of management events stored in S3 is free, then \$2.00 per 100,000 management events delivered.
- **EC2 instance traffic:** EC2 traffic between different AZs hosted within the region costs the same as region-to-region data transfer. ELB to EC2 traffic is free of charge within the AZ.

- **ELB:** ELB charges are calculated by **Load Balancer Capacity Units (LCUs)** for Application Load Balancer and Network Load Balancer. The regional location of your load balancer is also considered when calculating the LCU price. The LCU measures the *dimensions* of your traffic, which include new connections, active connections, the amount of bandwidth, and the number of rules processed by the load balancer. Whichever LCU dimension is highest per ALB/NLB-hour (or partial hour) is the one charged.
- **Amazon S3 Standard:** First 50 TiB/month \$0.023 per GiB; next 450 TiB/month \$0.022 per GiB.

Management Tool Pricing Example: AWS Config

As an example of management pricing, AWS Config enables customers to monitor the configuration of their IaaS resources, such as compute, networking, and monitoring resources, against a set of managed and customer-defined rules that capture any changes to the resource's configuration. For AWS Config to operate, it needs to execute its review of resources in the AWS account. AWS Config then stores the results in S3 storage.

AWS charges you for the number of configuration items being monitored by AWS Config in your AWS account for each AWS

region (see [Figure 12-2](#)). AWS Config managed rules are predefined rules created and managed by AWS and are organized into categories, such as compliance, security, and networking. An example of a Config managed rule is **encrypted-volumes**, which, once enabled, continually checks that all boot volumes attached to EC2 instances are encrypted. The management charges for AWS Config operation are as follows:

- AWS Config compares the **encrypted-volumes** rule against all boot volumes in your AWS account; the rule is defined as an “active rule” because the rule was processed by the Config management service. The first ten AWS Config managed rules are charged a flat rate of \$2.00 per AWS Config rule per region per month. There are additional charges for executing custom rules.
- The next 40 managed rules have a slightly lower price per month: \$1.50 per processed rule. When you reach 50 rules or more, the price drops further, to \$1.00 per processed rule. Actual prices might increase or decrease over time. Examples provided throughout this book are used to demonstrate the pricing models used by AWS; actual pricing rates change over time.

AWS Config rules evaluations	Price
First 100,000 rule evaluations	\$0.001 per rule evaluation per region
Next 400,000 rule evaluations (100,001-500,000)	\$0.0008 per rule evaluation per region
500,001 and more rule evaluations	\$0.0005 per rule evaluation per region

Figure 12-2 AWS Config Rules Pricing

AWS Config's data gathering is carried out using AWS Lambda functions. Lambda is an AWS-managed service that runs functions that can be created using several programming languages, including Python, C#, Node.js, and Java. When AWS Config rules are processed, additional AWS Lambda charges are applied for each AWS Config custom rule. Lambda charges are based on the amount of CPU, RAM, and time taken to complete the execution of each function.

AWS Config Results

The results gathered by AWS Config are stored in an Amazon S3 bucket. Charges for S3 storage depend on the size of the storage results. For example, S3 storage pricing is \$0.023 per gigabyte for the first 50 TiB per month. AWS Config is also integrated with the monitoring service Amazon CloudWatch and sends detailed information about all configuration changes and notifications to AWS Simple Notification Server events.

Using AWS Config, customers can be alerted if their infrastructure components are found to be out of compliance. For example, if a developer creates an unencrypted boot volume, AWS Config alerts could use the Simple Notification Service (SNS) to call an AWS Lambda function. The first 1 million SNS requests per month are free; after that, they are charged \$0.050 per 1 million SNS requests.

For this example, if a custom AWS Config rule discovers an unencrypted boot volume, an SNS event could trigger a custom AWS Lambda function to perform a snapshot of the unencrypted boot volume, and then delete the out-of-compliance EC2 instance and unencrypted volume.

Consider a scenario where you have 500 configuration items in your AWS account that are checked by AWS Config each month with 30 active AWS management rules in the us-east-1 (Northern Virginia) region. Your monthly charges would be as follows:

AWS Config costs: $500 \times \$0.003 = \1.50

AWS Config rules: \$2.00 per active rule for the first ten active config rules = \$20.00

\$1.50 for each of the next ten active Config rules = \$15.00

Total AWS Config monthly charges = \$36.50

If resource configurations were found not to be in compliance with a defined AWS Config rule, notifications and alerts would be generated by AWS Config, resulting in additional SNS and AWS Lambda charges.

AWS pricing usually involves more than a single charge. Calculating costs at AWS is complicated because there are many moving parts in each managed service.

Cost Management Tools

To access cost management tools for your AWS account, select My Billing Dashboard from the AWS Management Console, as shown in [Figure 12-3](#). The root account of your AWS account has full access to the Billing Dashboard. Using AWS IAM, access can be delegated to specific IAM users or IAM roles for accessing billing and cost management data for the AWS account by activating access to the Billing Dashboard and then attaching the desired IAM policy. The AWS Cost Explorer API is the engine powering the Cost Explorer; it can also be directly accessed by customers who wish to query cost and usage data programmatically or from the AWS CLI.

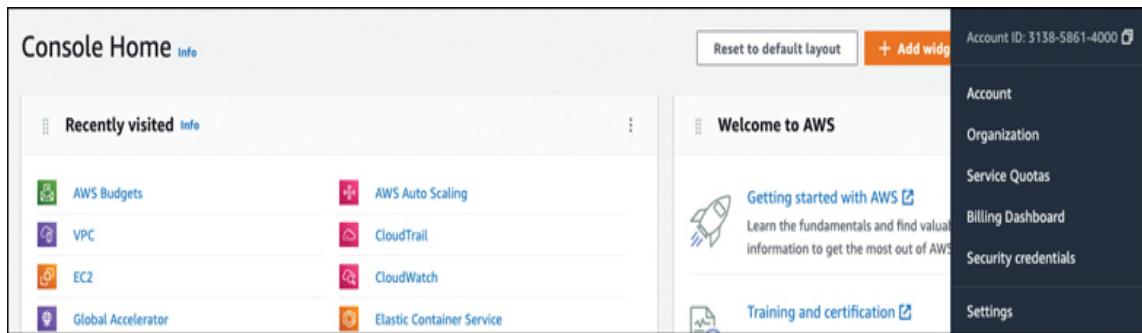


Figure 12-3 Accessing the Billing Dashboard and Cost Management

The cost management tools available at AWS for financial management are listed in [Table 12-4](#) by use case, by Cost Management feature, and Billing Dashboard feature, along with associated AWS services.

Key Topic

Table 12-4 Cost Management Tools by Use Case

Use Case	Details	Cost Management Feature

Use Case	Details	Cost Management Feature
Organize	Define a tagging policy	—
Report	Default and custom cost reports	Cost Explorer
Access	Track costs across AWS Organizations	—

Use Case	Details	Cost Management Feature
Control	—	Cost Anomaly Detection
Forecast	—	Cost Explorer Budgets reports
Budget	Budget threshold and alert notifications to control spend	Budgets Budgets actions
Purchase	Discounts for compute usage	Savings Plans Reserved Instances

Use Case	Details	Cost Management Feature
Rightsizing Recommendations	Match Reserved Instance allocation to current workload needs	Rightsizing recommendations
Inspect	Current resource deployment and costs	Cost Explorer



Note

AWS Resource Groups can be used to manage a collection of tagged AWS resources that reside in the same region. Supported AWS resources include Amazon S3 buckets, Amazon SNS, Service Quotas, AWS Secrets Manager, Amazon SageMaker,

Amazon Route 53, Amazon RDS, Amazon Redshift, AWS Organizations, Lambda functions, AWS IAM, AWS Config, Amazon DynamoDB, AWS CloudTrail, AWS CloudWatch, Amazon CloudFront, Amazon FSx for Windows File Server, ELB, EFS, ECS, and EC2 instances.

AWS Cost Explorer



Cost Explorer helps customers analyze AWS account costs and overall usage with free reports. Default reports include a breakdown of the AWS services that are incurring the most costs, including overall EC2 usage and Reserved Instance utilization. Optionally, an organization can carry out a deep-dive cost analysis, filtering with numerous dimensions; for example, by AWS service and region. AWS accounts that are members of an AWS Organization can take advantage of consolidated billing and review the charges generated by all member accounts. Using Cost Explorer (see [Figure 12-4](#)), you can filter AWS costs based on the following:

- **API operation:** Requests and tasks performed by each AWS service
- **Availability zone:** Charges per availability zone
- **All costs:** Costs per AWS account
- **Linked accounts:** Member accounts in an AWS organization
- **AWS region:** Where operations and AWS services operated
- **AWS service:** AWS services used
- **Tags:** Cost allocation tags assigned to the selected service
- **Tenancy:** Multi- or single-tenancy EC2 instances
- **Usage type:** Amounts of AWS service data (compute hours, data transfer in or out, CloudWatch metrics, I/O requests, and data storage)

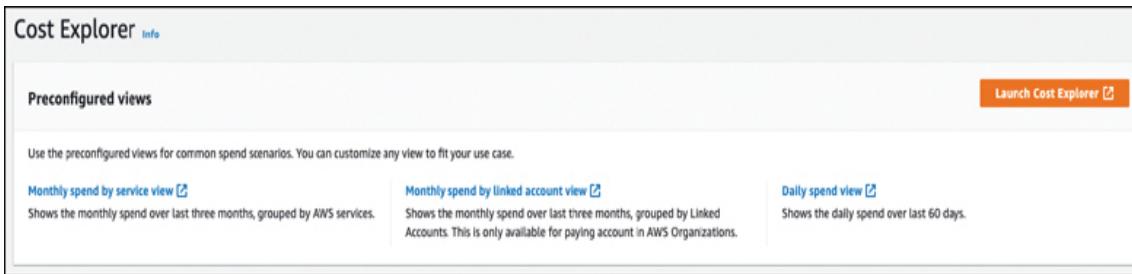


Figure 12-4 Cost Explorer

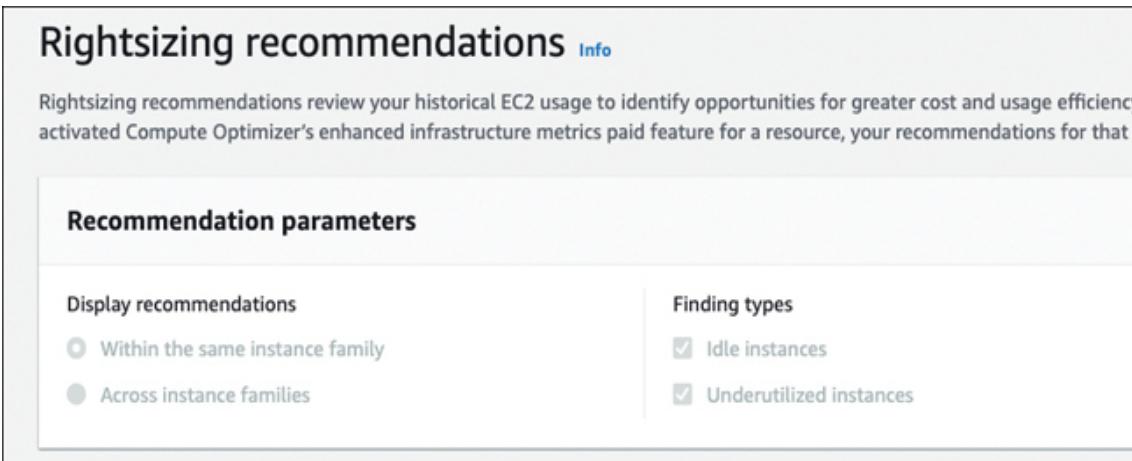
To enable tracking of your costs with Cost Explorer, you need to sign in to the AWS Management Console and enable data collection by opening the Billing Dashboard and Cost Management and launching the Cost Explorer. After the initialization of Cost Explorer, costs are displayed for the

current month and will be forecast for the next 12 months; updates will be made to the current spend every 24 hours. Cost Explorer can also display historical cost data, costs for the current month, and trends and forecasts. The following are the default Cost Explorer reports:

- **AWS Marketplace:** Costs for the products you've ordered through AWS Marketplace.
- **Daily Costs:** How much you spent over the past 6 months and projected costs over the next month.
- **Monthly Costs by Linked Account:** Costs for linked accounts in an AWS organization.
- **Monthly Costs by Service:** Costs over the past 6 months highlighted by the top five services' costs.
- **Monthly EC2 Running Hours Costs and Usage:** The amount spent on active reserved instances (RIs).
- **Reserved Instance Utilization Report:** Reserved instances used, including any overspending and net savings from using reserved instances.
- **Reserved Instance Coverage Report:** Details on how many EC2 instance hours have been covered by reserved instances, how much was spent on the on-demand instance, and how much could be saved by purchasing reserved instances.
Compute services that can use EC2 reserved instances include Amazon Redshift, Amazon RDS, Elasticsearch

clusters, and Amazon ElastiCache. Filters include the specific AZ, EC2 instance type, linked AWS account, operating system platform, AWS region, and compute tenancy. In addition, detailed information for each reservation can be downloaded as a CSV file.

- **Rightsizing Recommendations:** Cost Explorer recommendations for improving the use of reserved instances and AWS resources (see [Figure 12-5](#)).



The screenshot shows the 'Rightsizing recommendations' section of the AWS Cost Explorer. At the top, there's a heading 'Rightsizing recommendations' with an 'Info' link. Below it, a descriptive text states: 'Rightsizing recommendations review your historical EC2 usage to identify opportunities for greater cost and usage efficiency activated Compute Optimizer's enhanced infrastructure metrics paid feature for a resource, your recommendations for that r'. Underneath, there's a 'Recommendation parameters' section with two columns. The left column contains 'Display recommendations' with two options: 'Within the same instance family' (selected) and 'Across instance families'. The right column contains 'Finding types' with two options: 'Idle instances' (selected) and 'Underutilized instances'.

Figure 12-5 Cost Explorer Recommendations

Cost Explorer provides costing reports for daily and monthly charges based on the AWS services that you subscribe to (see [Figure 12-6](#)). The reports offered include information on the following:

- Monthly costs by service (view costs and usage over the last 12 months)
- Daily costs by service
- Monthly costs by linked account (view the monthly spend for paying accounts in an AWS organization)
- Services ordered by AWS Marketplace
- Monthly EC2 instance running hours cost and usage
- Reservation utilization report, and Reservation coverage help analyze purchases and savings
- Savings Plans utilization report and Savings Plans coverage help analyze purchases and savings

The screenshot shows the 'Reports' section within the AWS Cost Management console. At the top, there's a breadcrumb navigation: 'AWS Cost Management > Reports'. Below it, the title 'Reports' is followed by a small 'Info' link. A sub-header 'All reports (9)' is displayed above a search bar with a magnifying glass icon and the placeholder 'Search'. A table follows, listing nine reports with columns for Report name, Type, Time range, and Time granularity.

	Report name	Type	Time range	Time granularity
<input type="checkbox"/>	Monthly costs by service	Cost and usage	Last 6 cal. months	Monthly
<input type="checkbox"/>	Monthly costs by linked account	Cost and usage	Last 6 cal. months	Monthly
<input type="checkbox"/>	Monthly EC2 running hours costs and usage	Cost and usage	Last 6 cal. months	Monthly
<input type="checkbox"/>	Daily costs	Cost and usage	Last 6 cal. months + Today to cal. month-end	Daily
<input type="checkbox"/>	AWS Marketplace	Cost and usage	Last 12 cal. months	Monthly
<input type="checkbox"/>	RI Utilization	Reservation utilization	Last 3 cal. months	Daily
<input type="checkbox"/>	RI Coverage	Reservation coverage	Last 3 cal. months	Daily
<input type="checkbox"/>	Utilization report	Savings Plans utilization	Last 3 cal. months	Daily
<input type="checkbox"/>	Coverage report	Savings Plans coverage	Last 3 cal. months	Daily

Figure 12-6 Cost Explorer Details

Customers can review the past 13 months of operation and forecast the potential bill for the next 3 months based on the current costs. Forecasts can be created for future AWS charges based on a custom time frame, filtering and grouping costs using several parameters, including:

- **Availability zone:** Where resources are located
- **Instance type:** The type of reserved instance used to launch EC2 and RDS instances

- **Purchase option:** On-demand, reserved, spot, or scheduled reserved pricing
- **AWS services ordered and used:** AWS service usage
- **Tenancy:** Dedicated or multi-tenant reserved EC2 instances
- **Cost allocation tags:** Generate hourly and monthly cost allocation reports based on tagged resources

Additional features of the Billing Dashboard and Cost Management that can be useful include analyzing current and future costs with Cost Explorer graphs, and receiving email notifications when charges reach a defined cost threshold by enabling billing alerts (see [Figure 12-7](#)). Companies using Reserved Instances can create a budget to track the current spend and expected spend. Each budget defines a start and end date and a budgeted amount to track costs against. Budgets can also include AWS costs related to specific AWS services, associated tags, purchase options, instance types, region, and AZ locations. When a budget forecast hits the defined threshold (a percentage of a budgeted amount or a dollar figure), alerts and notifications can be sent to email accounts and an Amazon SNS topic.

Configure alerts

You can send budget alerts via email and/or Amazon Simple Notification Service (Amazon SNS) topic ARN.

Budgeted amount [Edit](#)
300 GB

Alert 1

Send alert based on:

Actual Usage
 Forecast Usage

Alert threshold
80 [Usage Amount](#)

Notify the following contacts when **Actual Costs** is **Greater than -- (--)**.

Email contacts
markb@costts.prg

Figure 12-7 Enabling Billing Alerts

Note

Cost Explorer can review EC2 instance memory utilization if the CloudWatch agent has been enabled. To enable the agent for Linux instances, use **mem_used_percent**; for Windows instances, use **% Committed Bytes In Use**.

AWS Budgets

Key Topic

AWS Budgets tracks AWS costs and can use billing alerts (refer to [Figure 12-7](#)) to alert organizations when costs are outside defined budget guidelines. Budget information is updated every 8 hours. Budgets can also be created to monitor the overall utilization and coverage of your existing reserved instances or savings plans. Alert notifications can be sent to an Amazon SNS topic and up to ten email addresses.

The following types of budgets can be created using Budgets (see [Figure 12-8](#)):

- **Cost budgets:** Define how much to spend on a particular AWS service (for example, EC2 instances).
- **Usage budgets:** Define how much to spend on one or more AWS services.
- **RI utilization budgets:** Define the expected usage level of purchased Reserved Instances and get alerted if RI usage falls below the defined usage threshold.
- **RI coverage budgets:** Define the expected coverage level of purchased Reserved Instances and get alerted if RI coverage of EC2 instances falls below the defined threshold number of hours.

- **Savings Plans utilization budgets:** Define the expected usage level of EC2 instances, Fargate, and AWS Lambda functions and get alerted when your savings plan falls below the defined threshold.
- **Savings Plans coverage budgets:** Define the expected coverage level of EC2 instances, Fargate, and AWS Lambda functions and get alerted when your savings plan falls below the defined threshold.

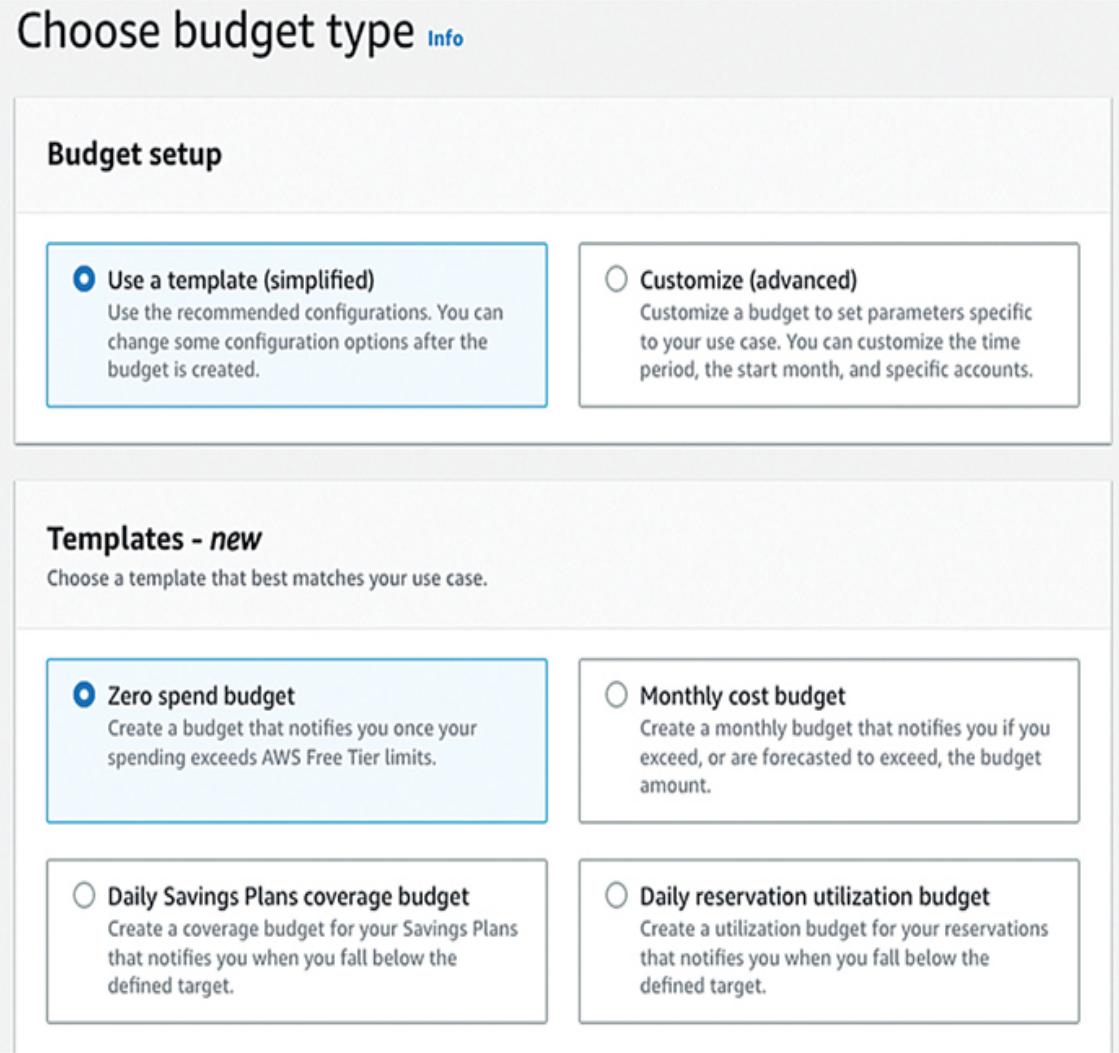


Figure 12-8 Budget Choices

You can also create custom budgets with predefined templates such as:

- Define a zero spend budget that notifies you when spending exceeds the AWS Free Tier limits.

- Define a budget with monthly costs, with fixed targets tracking all costs associated with an AWS account. Alerts could be defined for both actual and forecasted spending.
 - Define a budget with escalating monthly spending costs, with notifications alerting when funds are spent in excess of the allowable increases.
 - Define a budget with a fixed usage amount, with notifications alerting when the budget spend is close to being exceeded.
-

Note

Budget notifications use Amazon SNS to send alerts. Alerts are sent to defined SNS topics, and automated actions can be performed using AWS Lambda functions. Notifications can also alert via email or text message.

AWS Cost and Usage Reports



Cost and Usage Reports (CUR) provide a comprehensive overview of the monthly costs and usage of AWS services per AWS account or AWS organization (see [Figure 12-9](#)), showing

hourly, daily, or monthly expenses based on products and resources used or based on resource tags that have already been defined. From the Billing Dashboard and Cost Management, select Cost and Usage Reports, and click Create Report. Resource IDs can be used to create individual line items for each AWS resource used. An Amazon S3 bucket must be chosen as the location for storing requested reports.

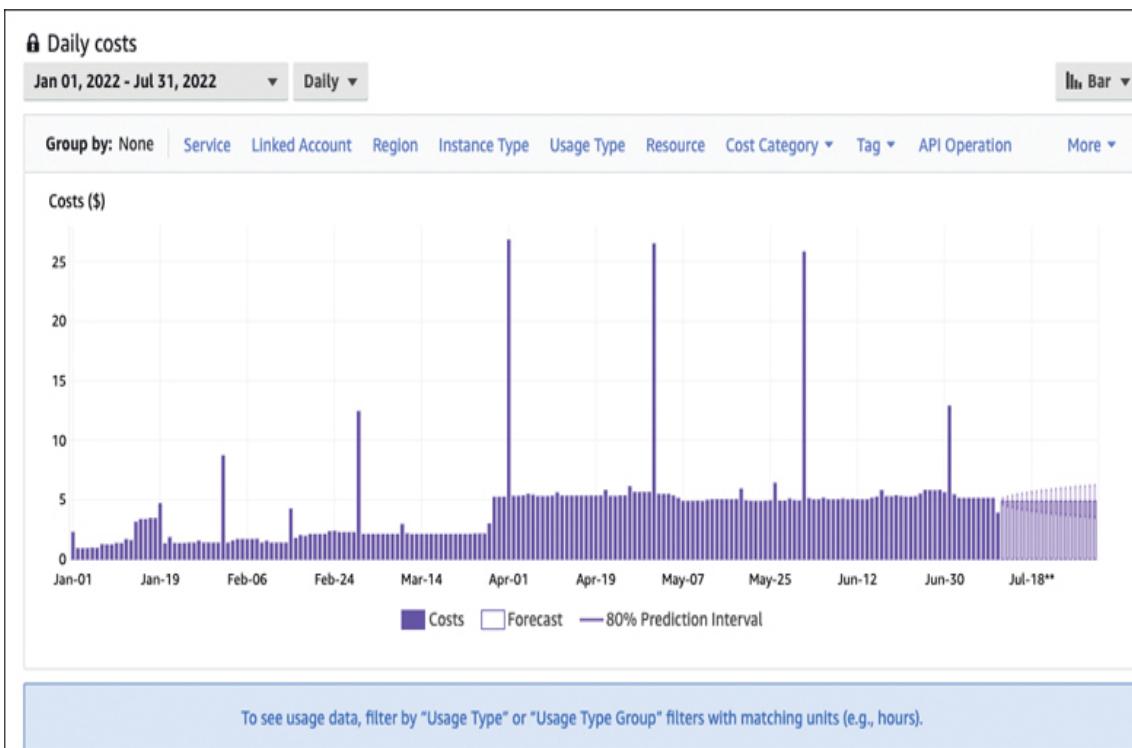


Figure 12-9 Cost and Usage Report

Reports can also be viewed directly from the Cost and Usage Reports console; a CSV usage report can also be downloaded. Optionally, you can use Amazon Athena with standard SQL

queries, or load CUR data to an Amazon Redshift deployment, or use Amazon QuickSight for additional analysis.

Managing Costs Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of managing costs:

- Track your costs and usage against defined budgets by using AWS Budgets.
- Use AWS Cost Explorer to review current costs and usage by creating a Cost and Usage Report.
- Create tags for your AWS resources and then activate the tags to add information to a Cost and Usage Report.
- Use AWS Cost Explorer to perform forecasting for your costs and overall usage for the upcoming month or year.
- Use AWS Organizations to take advantage of consolidated Billing Dashboard, with one bill for multiple accounts.
- Combine resource usage across all AWS accounts contained in an AWS organization, resulting in the sharing of volume pricing discounts.

- Use AWS Budgets to define a monthly budget for your AWS costs and usage and discounts for savings plans or reserved instances.
- Use AWS Config to review the inventory of your AWS resources and configuration within your AWS account or across your AWS organization.
- Use AWS Systems Manager to get a detailed inventory of your EC2 assets.
- Use tag policies in AWS Organizations defining rules for how tags must be used on AWS resources across your AWS accounts in the organization.

Tagging AWS Resources

One essential task to consider is the creation of tags when you create AWS resources. Each tag is a key/value pair, and 50 individual tags can typically be created for each AWS resource. Tags can be used for automation, analysis, reporting, compliance checks, and with the Billing Dashboard. Custom tags can be used for many reasons, including the following examples:

- **Tags on creation:** Tags can be created for EBS volumes at the time of creation.

- **Cost allocation tags:** These tags allow you to have visibility into your actual snapshot storage costs. In the Billing Dashboard and Cost Management, you can select Cost Allocation Tags and then select and activate tags that can be used for the Billing Dashboard process (see [Figure 12-10](#)).
- **Enforced tags:** AWS IAM security policies can enforce the use of specific tags on EBS volumes and control who can create tags.

Cost allocation tags <small>Info</small>	
Cost allocation tags activated: 9	
User-defined cost allocation tags	AWS generated cost allocation tags
User-defined cost allocation tags (16) <small>Info</small>	
<input type="text"/> Search for a tag key	All statuses ▾
<input type="checkbox"/> Tag key	▲ Status
<input type="checkbox"/> apache	✗ Inactive
<input type="checkbox"/> aws-control-tower	✗ Inactive
<input type="checkbox"/> Default	✓ Active
<input type="checkbox"/> Description	✗ Inactive
<input type="checkbox"/> elasticbeanstalk	✗ Inactive
<input type="checkbox"/> elasticbeanstalk:environment-id	✓ Active
<input type="checkbox"/> elasticbeanstalk:environment-name	✓ Active
<input type="checkbox"/> graphic dept	✗ Inactive
<input type="checkbox"/> Interface	✓ Active

Figure 12-10 Activate Cost Allocation Tags

Using Cost Allocation Tags

Key Topic

Cost allocation tags can be created and deployed to track your AWS costs via cost allocation reports, which make it easier for you to track and categorize your existing AWS costs. There are two types of ***cost allocation tags***: AWS-generated tags and user-defined tags. Each tag contains a key and a linked value. For example, user-defined tags might be **Mark=Developer** and **Costs** (see [Figure 12-11](#)). An example of an AWS-generated tag is the **createdBy** tag, which tracks who created the resource. The name of user-generated tags could include Cost Management and Stack.



Figure 12-11 Cost Allocation Tags Example

AWS has defined the **createdBy** tag for use with selected AWS resources for cost allocation tracking. AWS-generated tags must be activated in the Billing Dashboard and Cost Management before they can be used. The **createdBy** tag is supported by the following AWS services: AWS CloudFormation, Amazon

Redshift, Amazon Route 53, Amazon S3 storage, AWS Storage Gateway, and Amazon EC2 instances and networking components, including VPCs, security groups, snapshots, subnets, and Internet gateways.

As mentioned, once tags have been created and applied to an AWS resource, you can activate the tags in the Billing Dashboard and Cost Management, which then generates a cost allocation report in a CSV file that has your usage and costs grouped by the assigned active tags. AWS Cost Explorer and AWS Cost and Usage Reports can break down AWS costs by tags.

Note

Tag policies can be created to standardize tags across AWS Organizations, allowing customers to label resources using key/value pairs. Tag policies help Cost Explorer to identify resources by dimensions such as owner, cost center, or environment, helping identify and break out the cost of AWS.

Storage Types and Costs

Storage costs at AWS depend on the storage service being used—whether it's EBS storage volumes, shared file storage using Amazon EFS or Amazon FSx for Windows File Server, Amazon S3 object storage, or archival storage using Amazon S3 Glacier. The following list describes these AWS storage options and storage and data transfer costs:

- **Amazon S3 buckets:** An S3 bucket has monthly storage and retrieval costs based on the storage class and location of the request. Other costs are based on the frequency of operation. **PUT, COPY, POST, and LIST** requests per 1,000 requests for S3 Standard is \$0.005. **GET, SELECT, and all other requests** per 1,000 requests for S3 Standard is \$0.004. S3 lifecycle and data transfer requests are charged per 1,000 requests. There are no data transfer charges for storing and retrieving objects from an S3 bucket located in the same region where the EC2 instance is located.

Optional Amazon S3 features such as S3 Lifecycle transitions, data transfer (outbound directly to the Internet or to Amazon CloudFront), S3 Transfer acceleration, and Cross-Region replication to another S3 bucket all have separate and additional costs. Amazon S3 bucket replication within or across AWS regions also has specific bundled costs:

- **S3 Same-Region Replication (SRR):** Amazon S3 charges for storage in the selected destination Amazon S3 storage

class, the primary copy, replication **PUT** request, and if applicable, an infrequent access storage retrieval charge.

- **S3 Cross-Region Replication (CRR):** S3 charges for storage in the selected destination S3 storage class, the primary copy, replication **PUT** requests, and if applicable, an infrequent access storage retrieval charge, and inter-region data transfer out to the selected region.

As an example, storing 500 TiB of standard storage in the us-east-1 (Northern Virginia) region would cost you roughly \$12,407.14 per month (see [Figure 12-12](#)). It would include 5,000 **PUT/COPY/POST/LIST** requests and 30,000 **GET** requests.

Services	Estimate of your Monthly Bill (\$ 12407.14)
Choose region: <input type="button" value="US East (N. Virginia)"/>	
Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers	
A newer version of the S3 calculator is available	
S3 Standard Storage & Requests:	
Storage:	<input type="text" value="500"/> TB
PUT/COPY/POST/LIST Requests:	<input type="text" value="0"/> Requests
GET/SELECT and Other Requests:	<input type="text" value="0"/> Requests
Data Returned by S3 Select	<input type="text" value="0"/> GB
Data Scanned by S3 Select	<input type="text" value="0"/> GB

Figure 12-12 S3 Storage Pricing

- **Amazon S3 Glacier:** S3 Glacier archive storage ranges from \$0.04 to under \$0.01 for archival storage in S3 Glacier Deep Archive:
 - S3 Glacier Instant Retrieval storage costs \$0.004 per GiB when accessed every 90 days
 - Amazon S3 Glacier Flexible Retrieval storage with retrieval from 1 minute to 12 hours (\$0.0036 per GiB)
 - Amazon S3 Glacier Deep Archive storage that is accessed once or twice a year and is restored within 12 hours (\$0.004 per GiB)

Amazon S3 Glacier storage is also subject to storage and retrieval pricing that is based on the speed of the data retrieval required. In addition, it is subject to outbound data transfer pricing, which varies based on the destination (for example, outbound directly to the Internet or to CloudFront). A recommended practice is to archive infrequently used data in S3 Glacier Flexible Retrieval and move long-term archived data to Glacier Deep Archive. Storing 100 TiB of archived records in the US-East (Northern Virginia) region with data retrieval of 10 GiB per month and an average of 20 requests per month would cost roughly \$411.92.

- **EBS volumes:** Virtual hard drives can be ordered in several flavors: SSDs, SSDs with provisioned IOPS, throughput-

optimized drives, or cold HDDs (infrequently accessed hard drive storage). You are also charged for snapshot storage in Amazon S3 for EBS volume snapshots.

For example, a single general-purpose SSD sized at 16,384 GiB hosted in the US-East-1 (Northern Virginia) region would cost you roughly \$1,798 per month. A provisioned IOPS SSD io1 volume sized at 8,000 GiB with 16,000 IOPS hosted in the US-East-1 (Northern Virginia) region would cost you \$2,244.50 per month, as shown in [Figure 12-13](#). Note that all prices quoted are subject to change over time.

Services		Estimate of your Monthly Bill (\$ 2244.50)																					
Choose region: US East (N. Virginia) ▾		Inbound Data Transfer is Free and Outbound Data																					
 Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud, making it easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances.																							
Compute: Amazon EC2 Instances: <table border="1"> <thead> <tr> <th>Description</th> <th>Instances</th> <th>Usage</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>(+ Add New Row)</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>						Description	Instances	Usage	Type	(+ Add New Row)													
Description	Instances	Usage	Type																				
(+ Add New Row)																							
Compute: Amazon EC2 Dedicated Hosts: <table border="1"> <thead> <tr> <th>Description</th> <th>Number of Hosts</th> <th>Usage</th> <th>Type</th> <th>Billing Option</th> </tr> </thead> <tbody> <tr> <td>(+ Add New Row)</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>						Description	Number of Hosts	Usage	Type	Billing Option	(+ Add New Row)												
Description	Number of Hosts	Usage	Type	Billing Option																			
(+ Add New Row)																							
Storage: Amazon EBS Volumes: <table border="1"> <thead> <tr> <th>Description</th> <th>Volumes</th> <th>Volume Type</th> <th>Storage</th> <th>IOPS</th> <th>Baseline Throughput</th> </tr> </thead> <tbody> <tr> <td>(- Remove)</td> <td>1</td> <td>Provisioned IOPS SSD (io1) ▾</td> <td>8000 GB</td> <td>16000</td> <td>500 MBs/sec</td> </tr> <tr> <td>(+ Add New Row)</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>						Description	Volumes	Volume Type	Storage	IOPS	Baseline Throughput	(- Remove)	1	Provisioned IOPS SSD (io1) ▾	8000 GB	16000	500 MBs/sec	(+ Add New Row)					
Description	Volumes	Volume Type	Storage	IOPS	Baseline Throughput																		
(- Remove)	1	Provisioned IOPS SSD (io1) ▾	8000 GB	16000	500 MBs/sec																		
(+ Add New Row)																							

Figure 12-13 EBS Price Calculations

- **Snapshot storage:** Snapshot storage costs can be extremely high if snapshots that are no longer required are not deleted. Amazon Data Lifecycle Manager, which is found in the EBS section of the EC2 console, allows you to schedule and manage the creation and deletion of EBS snapshots.
- **Shared storage (EFS/FSx for Windows File Server):** Amazon EFS and Amazon FSx for Windows File Server are shared file storage services. At a minimum, you pay for the

total amount of storage used per month. EFS Infrequent Access storage is priced based on the amount of storage used and the amount of data accessed. You can also optionally pay for faster-provisioned throughput in megabytes per month, depending on your performance requirements. FSx for Windows File Server usage is prorated by the hour, and customers are billed for the average usage each month, paying for the storage and throughput capacity specified and for any backups performed. FSx for Windows File Server customers pay for data transferred across availability zones or peering connections in the same region and for data transferred out to other AWS regions.

As an example for EFS, suppose a file system hosted in the US-East-1 (Northern Virginia) region uses 300 GiB of storage for 20 days for a single month. The charges would be as follows: total usage (GiB-hours) = $300 \text{ GiB} \times 20 \text{ days} \times (24 \text{ hours/day}) = 144,000 \text{ GiB-hours}$. The total charge equates to \$43.20 per GiB-month. Moving your files to the EFS Infrequent Access storage tier would reduce your EFS storage costs by up to 92%.

[Table 12-5](#) provides a comparison of S3, EBS, EFS, and FSx storage services, including costs of storage, how to reduce storage costs, and the associated backup tools for each service.

Key Topic**Table 12-5** Amazon S3, EBS, EFS, and FSx for Windows File Server Comparison

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS)	Elastic File System (EFS)
Costs of storage	Scaled cost based on the first 50 TiB of storage used and the number of requests made (POST, GET) Data transfer per GiB out of S3	General-purpose SSD: \$0.8 per GiB per month Provisioned IOPS SSD: \$0.125 per GiB per month; \$0.065 per provisioned IOPS per month	Standard storage: per GiB per month Infrequent access storage: \$0.045 per GiB per month Access requests \$0.01 per

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS) Throughput-optimized	Elastic File System (EFS)
Storage size	No limit	Maximum storage size 65 GiB	Petabyte
Transfer costs	HDD: \$0.045 per GiB per month Cold HDD: \$0.015 per GiB per month	Optimized for throughput	Optimized for latency

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS)	Elastic File System (EFS)
Storage classes	Standard, Intelligent-Tiering, Standard IA, One Zone IA, Glacier Instant/Flexible Retrieval/Deep Archive	General-purpose SSD, Provisioned IOPS SSD io1, io2, Throughput optimized HDD volumes, Cold HDD volumes	or Infrequent Access or One Zone EFS One or One Zone EFS One Infrequent Access
File size	5 TiB	64 TiB maximum volume size	47.9 TiB single file volume size

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS)	Elastic File System (EFS)
How to reduce storage costs	Intelligent-Tiering, One Zone-Infrequent Access	Reduce volume size and type, reduce IOPS	Provisioned throughput, EFS Standard or EFS On-Demand, Zone-IA
Backup Tools	Cross-Region and Same-Region Replication	Snapshots, Data Lifecycle Manager	EFS Lifecycle Manager, Intelligent Tiering

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS)	Elastic File System (EFS)
Associated AWS service	AWS Backup, AWS Snowball, AWS Lambda	AWS Lambda	AWS Backup, AWS Lambda
Data location	Data stays within the region or requested AZ	Data stays within the same AZ	Data stored within a specific region

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS)	Elastic File System (EFS)
Data access options	Public (HTTP, HTTPS) or private network endpoints (Gateway)	Private AWS network from an EC2 instance	Private network from multiple instances from on-premises locations
Encryption	SSE: Amazon S3, AWS-KMS, SSE-C	AWS and KMS: managed (CMK) with AES 256-bit encryption	AWS and KMS: managed CMK with AES 256-bit encryption

Feature	Simple Storage Service (S3)	Elastic Block Store (EBS)	Elastic File System (EFS)
Availability	Four 9s; can survive the loss of two facilities	EBS volumes	Stored across multiple AZs
Use Case	Static files	Boot drives, database instances	Big data analytics
		SQL, NoSQL	media workflow (media editing, studio production or home director)

AWS Backup

Key Topic

AWS Backup is a centralized backup service for managing data backups across multiple AWS regions for AWS compute, storage services, and database services (see [Figure 12-14](#)). Backups can be on-demand, scheduled, or continuous. A continuous backup includes a continuous backup of Amazon RDS database instances and continuous backup of the transaction logs.

Continuous backups can restore RDS deployments with a point-in-time recovery (PITR) within 5 minutes of activity within a defined 35-day time period. Amazon S3 buckets can be restored within 15 minutes of recent activity. Backups can also be automated per EC2 instance with crash-consistent backups of attached EBS volumes. AWS Backup also integrates with AWS Organizations.

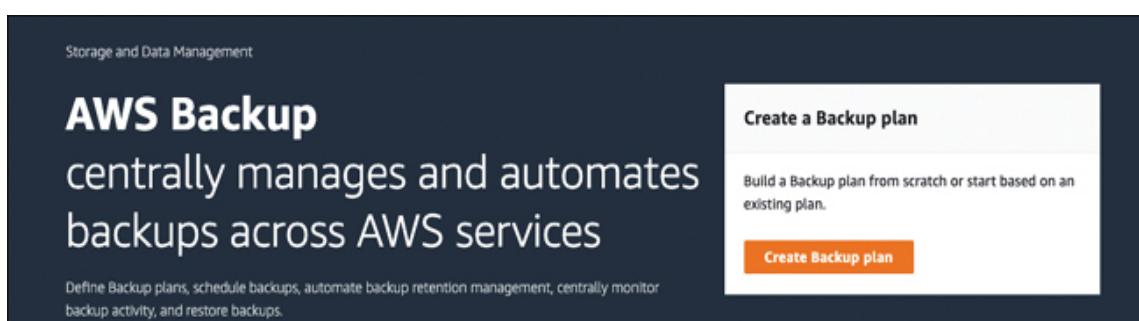


Figure 12-14 AWS Backup

The following AWS services can be backed up with AWS Backup:

- EBS volumes
- EC2 instances and Windows applications (including Windows Server, Microsoft SQL Server, and Microsoft Exchange Server)
- Amazon RDS databases (including Amazon Aurora clusters)
- Amazon DynamoDB tables
- Amazon Elastic File System file systems
- Amazon FSx for Windows File Server file systems
- Amazon FSx for Lustre, ONTAP, and OpenZFS file systems
- Neptune and DocumentDB clusters
- AWS Storage Gateway – Volume Gateway
- Amazon S3 buckets, objects, tags, and custom metadata
- Amazon Outposts, VMware Cloud on AWS, and on-premises VMware virtual machines (require AWS Backup gateway software to be installed on each VMware VM)

You can select templates when creating a backup plan with AWS Backup, or create a new backup plan (see [Figure 12-15](#)). When you assign a storage resource to a backup plan, the selected resource is backed up automatically on a defined schedule. A backup plan requires the following information:

- **Backup schedule:** Every hour (cron expression), 12 hours, daily, weekly, monthly
- **Backup window:** Starting time and duration
- **Lifecycle rules:** When a backup is transitioned to cold storage and when the backup expires
- **A backup vault:** For storing encrypted backups with KMS encryption keys
- **Regional copies:** Backup copies in another AWS region
- **Tags:** Associating multiple resources with tag-based backup policies

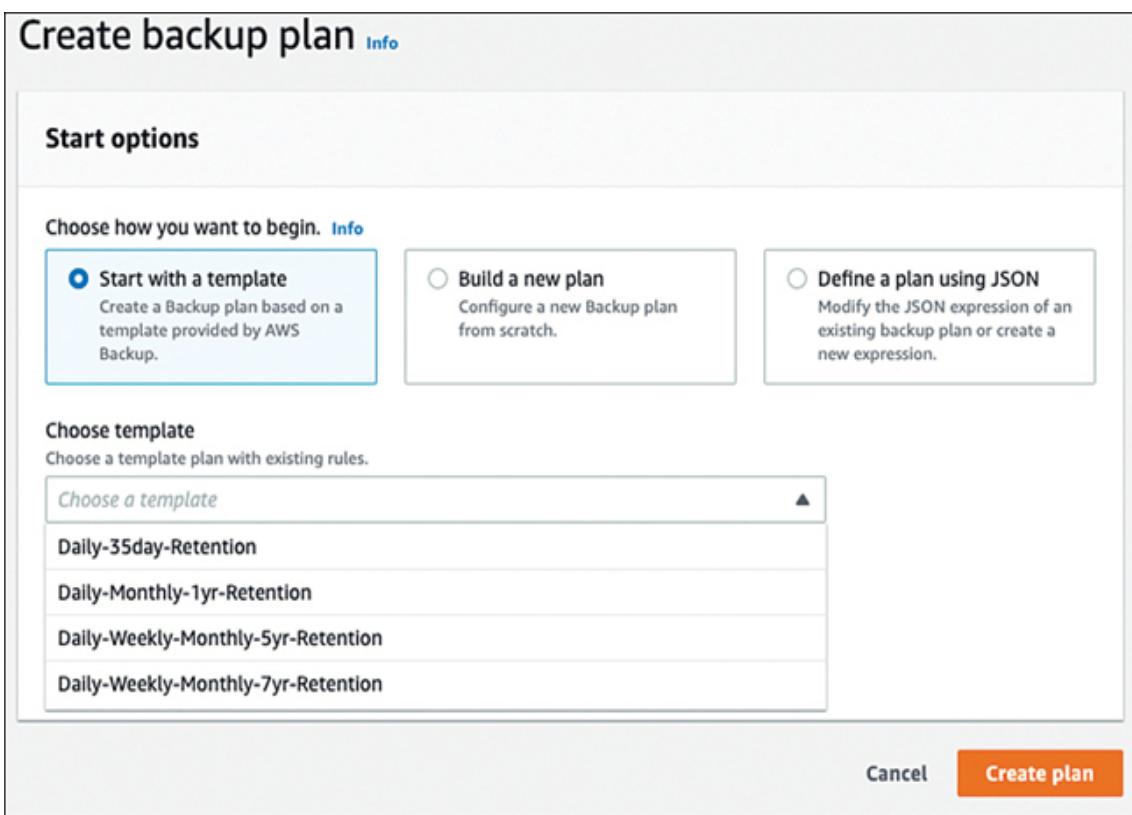


Figure 12-15 AWS Backup Plan

Lifecycle Rules

Key Topic

AWS Backup can be stored in either a warm or cold storage tier. ***Lifecycle rules*** allow customers to transition backups that are stored in warm storage to cheaper cold storage. In us-east-1, warm storage is \$0.05 per GiB; cold storage is \$0.01 per GiB. The defined lifecycle in each backup plan defines when a backup will transition into cold storage. Each backup stored in cold storage is a full backup. Backups that have transitioned to cold storage must remain in cold storage for 90 days. In [Figure 12-16](#), transition rules have been set to transition the monthly backup to cold storage after 8 days and retain the backup for 1 year.

Backup rule name

Monthly

Backup rule name is case sensitive. Must contain from 1 to 50 alphanumeric or '-' characters.

Backup vault [Info](#)

Default [Create new Backup vault](#)

Backup frequency [Info](#)

Monthly

on Day 1

Enable continuous backups for point-in-time recovery (PITR) [Info](#)
Available for RDS and S3 resources.

Backup window

Use backup window defaults - recommended [Info](#)
5 AM UTC, starts within 8 hours.

Customize backup window

Transition to cold storage [Info](#)

Days 8

Retention period [Info](#)

Years 1

Copy to destination [Info](#)

Choose a Region

▶ **Tags added to recovery points**
AWS Backup copies tags from the protected resource to the recovery point upon creation. You can specify additional tags to add to the recovery point.

Figure 12-16 Lifecycle Settings

AWS Backup Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of

AWS Backup:

- A backup of an EC2 instance includes snapshots of all volumes and launch configuration.
- A continuous backup allows you to restore RDS deployments any point in time within 35 days within 5 minutes of activity
- Periodic backups retain data for the specified duration.
- On-demand backups back up the selected resource type at once.
- Backup plans create incremental backups.
- Incremental backups are lower cost than an on-demand or periodic backup.
- The first backup is always a full backup; subsequent backups are incremental.
- When an EFS file system is created, automatic backups with AWS Backup are turned on.
- AWS Backups are stored in vaults.
- AWS Backup vaults are encrypted with KMS encryption keys.
- AWS Backup Vault Lock enforces a write-once, read-many (WORM) setting for all backups stored in a backup vault.
- AWS Backup Audit Manager audits the compliance of your AWS Backup policies.
- Amazon S3 backups require versioning to be enabled.
- AWS Backup charges by the GiB-month depending on the amount of resource type stored and restored per month.

- The AWS Backup lifecycle feature automatically transitions your recovery points from a warm storage tier to a lower-cost cold storage tier for backups of Amazon EFS file systems, Amazon DynamoDB tables, and VMware virtual machines.
- Individual files can also be restored without having to restore the entire file system.

Data Transfer Costs

There is no charge for inbound data transfer into AWS from the Internet, from an edge location, or Direct Connect connection.

- When data is transferred to the Internet from an AWS service, data transfer charges apply based on the service and the AWS region where the service is located.
- Data transfers across the Internet are billed at AWS region-specific and tiered data transfer rates.
- Data transferred into and out from Amazon EC2, Amazon RDS, Amazon Redshift, Amazon DynamoDB, Amazon ElastiCache instances, an Elastic Network Adapter, or VPC peering connections across availability zones in the same AWS region is charged at \$0.01/GiB in each direction.
- Data transferred across regional endpoints between Amazon S3, Amazon S3 Glacier, Amazon DynamoDB, Amazon Simple Queue Service (SQS), Amazon Kinesis, Amazon Elastic

Container Registry (ECR), Amazon SNS, and Amazon EC2 instances in the same AWS region is free of charge. However, if data is transferred across a PrivateLink connection, VPC endpoint, AWS NAT Gateway Service, or AWS Transit Gateway, data transfer charges will apply.

The AWS Certified Solutions Architect – Associate (SAA-C03) exam will require an understanding of the available solutions for transferring data records into AWS. Regardless of the location of your data, an ever-increasing number of tools and services are available to move your data from on-premises locations into the AWS cloud. [Tables 12-6](#), [12-7](#), [12-8](#), and [12-9](#) pose questions and details for data transfer options that are covered on the SAA-C03 exam.

Table 12-6 What Type of Data Do You Need to Transfer from On Premises to AWS?

Data Type	Transfer Option	Costs
Text	File sharing	Low

Data Type	Transfer Option	Costs
Virtual server images	AWS Application Migration Service, AWS Server Migration Service (SMS)	Free for the first 90 days for each server migrated. EC2 and EBS charges.
Database	AWS Database Migration Service (DMS)	Data transfer into AWS DMS is free. Data transferred between DMS and databases in RDS and EC2 instances in the same AZ is free.

Data Type	Transfer Option	Costs
Bulk storage files	AWS Transfer Family (SFTP, FTPS, and FTP)	\$0.30 per hour for enabled service. \$0.04 per gigabyte for the amount of data uploaded/downloaded.

Table 12-7 Where Will On-Premises Data Be Stored at AWS?

Data Usage	Storage Options
Daily use at AWS	Amazon S3, Amazon EFS, or FSx for Windows File Server
Archived storage	Amazon S3 Glacier
Stored long-term	Amazon S3 Glacier Deep Archive

Table 12-8 How Much Data Needs to Be Transferred?

Data Size	Data Transfer Option
Gigabytes	AWS Transfer Family
Terabytes	AWS Snowball, AWS Snowcone, or AWS Snowball Edge
Exabytes	AWS Snowmobile

Table 12-9 What Data Transfer Method and Hybrid Solution Could You Choose?

Private Network Connection to AWS	AWS Direct Connect
Edge location transfer	S3 Transfer Acceleration
Internet transfer	AWS DataSync or AWS Transfer for SFTP

Private Network
Connection to AWS

AWS Direct Connect

Offline data
transfer

AWS Snowball, AWS Snowball
Edge, or AWS Snowmobile

Hybrid storage

AWS Storage Gateway

Options for moving data records from on-premises locations into the AWS cloud are as follows:

**Key
Topic**

- **AWS Direct Connect:** AWS Direct Connect allows you to create a private single-mode fiber connection from your on-premises data center or a co-location into AWS; a connection can be partitioned into up to 50 private virtual interfaces connecting to public and VPC resources at AWS.
- **AWS DataSync:** AWS DataSync can automate the movement of large amounts of data from on-premises locations to either Amazon S3 buckets or Amazon EFS storage across the Internet, or with an AWS Direct Connect or AWS VPN

connection. Both one-time and continuous data transfers are supported using the NFSv4 protocol. Parallel processing creates fast data transfers using an AWS DataSync virtual machine agent downloaded and installed on your network. The first step is to create a data transfer task from your on-premises data source (NAS or file system) to the selected AWS destination, and then start the transfer. Data integrity verification is continually checked during the data transfer; data records are encrypted using Transport Layer Security (TLS). AWS DataSync supports both PCI DSS and HIPPA data transfers.

- **The AWS Snow Family:** The Snow family includes AWS Snowcone, AWS Snowball, and AWS Snowball Edge network-attached devices, or an AWS Snowmobile truck with a 40-foot storage container. Configuration involves logging in to the Snowball dashboard to create a job, selecting the parameters of the ***Snow device*** you wish to order, and select the S3 bucket that will store the locations of the Snow device once it is shipped back to AWS. When data has been moved to the selected S3 bucket and verified, the Snow device is securely erased and sanitized, removing all customer information. AWS Snow pricing is based on data transfer job fees, the commitment period, data transfer, and storage and shipping

fees. Data transfer into Amazon S3 from an external location is free. The following Snow Family options are available:

- **AWS Snowcone:** This is the smallest member of the Snow Family, with two vCPUs, 4 GiB of memory, and 8 TiB of object or block storage. It also has wired network access and USB-C power.
- **AWS Snowball:** Petabyte data transfer is possible using multiple Snowball devices; each device can hold either 42 TiB or 80 TiB of object or block storage. After you create a job request, as shown in [Figure 12-17](#), a Snowball device will be shipped to you via UPS. When you receive the device, hook it up to your network using an RJ-45 connection. The Snowball client software must be installed, and predefined security information must be entered before data transfer begins. After the data is transferred into the Snowball device, the device is shipped back to AWS and the device's data is deposited into an S3 bucket. This process can also be reversed, transferring object data from AWS back to your on-premises location. All data that is transferred to Snowball is encrypted with 256-bit encryption keys defined using AWS Key Management Service (KMS). The following use case options are available for Snowball devices:

- **Compute-optimized Snowball:** 42-TiB GPU option for machine learning or advanced video analysis use cases (\$1,200 to \$1,600 per job)
- **Storage-optimized Snowball:** Large data transfers and local storage (\$300 to \$500 per job)

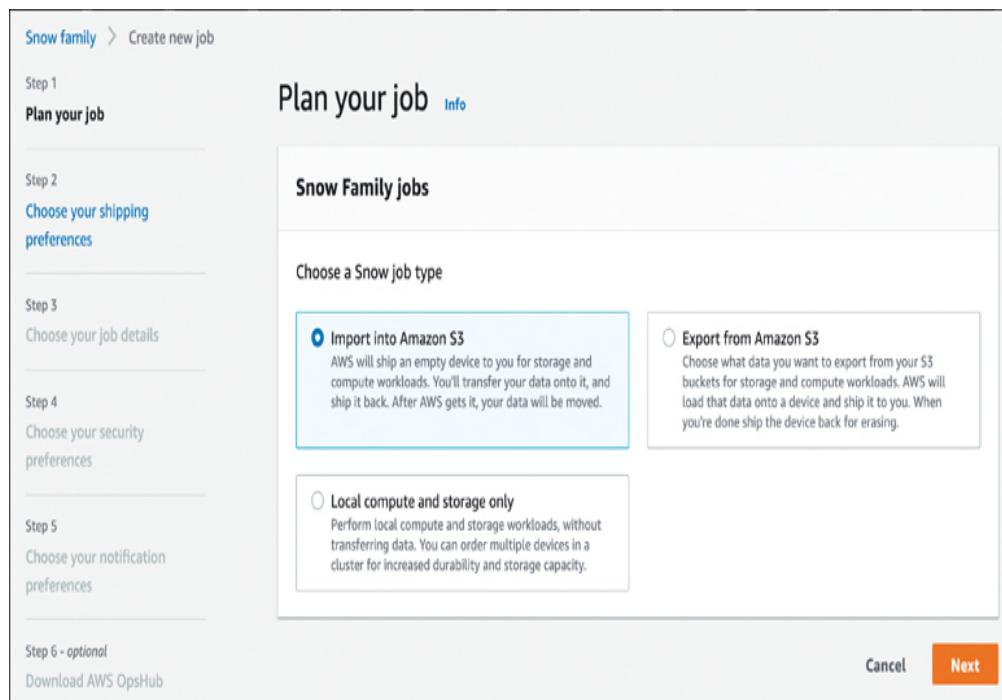


Figure 12-17 Creating a Snowball Job

- **AWS Snowball Edge:** The Snowball Edge device supports the installation of a local instance to carry out the local processing duties that can be built from your AMIs. Snowball Edge compute options are designed for

local data processing within the device with storage for processing or analysis before being stored back at AWS.

- **AWS Snowmobile:** Move up to 100 PB of data with an AWS Snowmobile truck. AWS employees show up with a transport truck containing a 45-foot shipping container and attach it to your data center. After the shipping container is filled with data, it is carefully driven back to AWS accompanied by an escort vehicle for safety, and the data is uploaded into S3 storage.
- **AWS Transfer Family:** Transfer files into and out of S3 buckets using the SSH File Transfer Protocol (SFTP). Connect existing SFTP software to the SFTP endpoint at AWS, set up user authentication, select an S3 bucket, assign IAM access roles, and transfer data records to AWS.

AWS Storage Gateway



AWS Storage Gateway is a hybrid storage solution that allows you to integrate your on-premises network with AWS storage and allows your on-premises applications and utilities to seamlessly store data records to Amazon S3, Amazon S3 Glacier, and FSx or Windows Fire Server storage. AWS Storage Gateway

can be used for backing up and archiving documents, storage migration, and storing on-premises tiered storage at AWS as a background process. The actual AWS Storage Gateway gateway device can be a hardware device such as a Dell EMC PowerEdge server with Storage Gateway preloaded, or a virtual machine image that can be downloaded and installed in VMware or Hyper-V environments. There are four configuration choices available for deploying AWS Storage Gateway:

- **Amazon S3 File Gateway:** File Gateway interfaces directly into Amazon S3 storage and allows you to store and retrieve files using either NFS or SMB, as shown in [Figure 12-18](#). Access S3 storage from EC2 instances or from on premises.
- **File Gateway—Amazon FSx for Windows File Server:** Begin Windows file-based storage migration to AWS for data that is frequently accessed. Supports the SMB protocol.

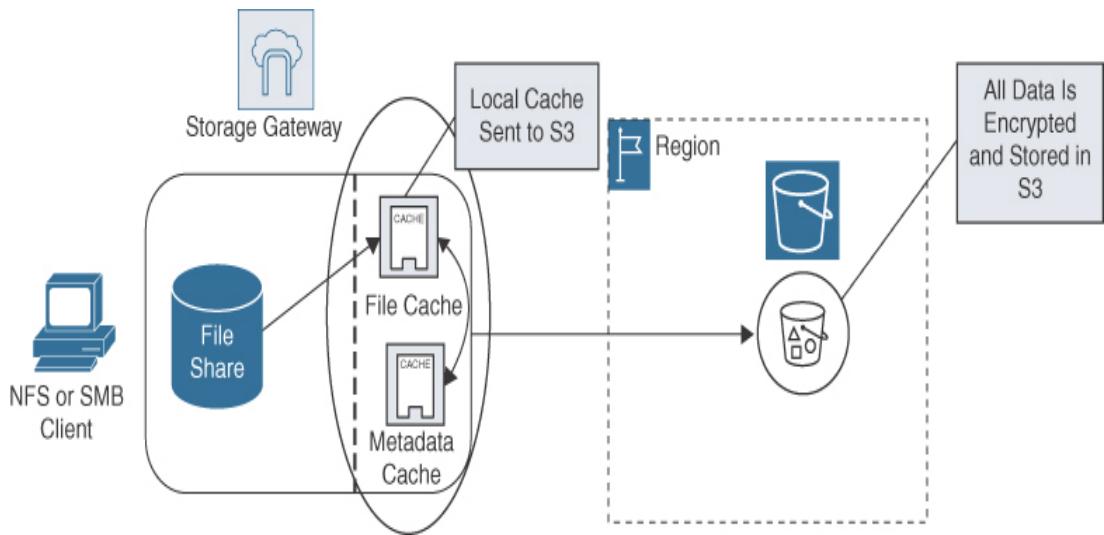


Figure 12-18 Storage Gateway: File Gateway Architecture

- **Volume Gateway:** Volume Gateway provides Amazon S3 cloud storage that can be mounted as an on-premises iSCSI device. Data is stored in Amazon S3 with a copy of frequently accessed data cached locally with the iSCSI volumes asynchronously backed up to Amazon S3 using incremental snapshots, as shown in [Figure 12-19](#).
- **Tape Gateway:** Tape Gateway is a virtual tape drive that supports a wide variety of third-party backup applications and allows you to store and archive virtual tapes in Amazon S3 storage using the iSCSI protocol. Virtual tape backups can also be moved to Amazon S3 Glacier using lifecycle rules.

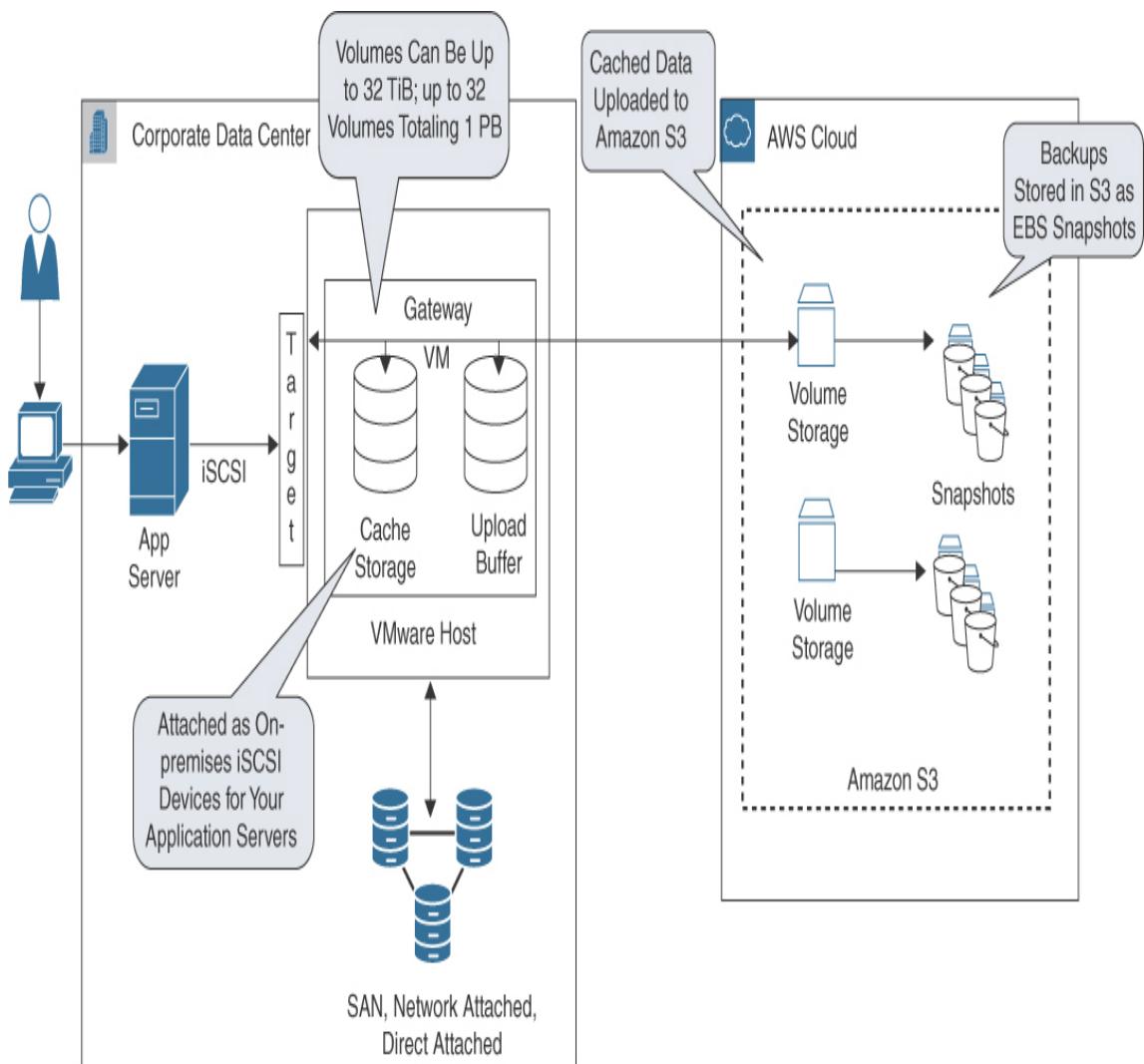


Figure 12-19 Storage Gateway: Volume Gateway Architecture

AWS Storage Gateway Cheat Sheet

Key Topic

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of AWS Storage Gateway:

- AWS Storage Gateway provides hybrid storage between on-premises environments and AWS.
- AWS Storage Gateway stores frequently accessed content on-premises while storing data securely and durably in S3 storage.
- AWS Storage Gateway is useful for on-premises disaster recovery solutions.
- AWS Storage Gateway is useful for cloud migrations of data records.
- AWS Storage Gateway supports three storage interfaces: File Gateway, Volume Gateway, and Tape Gateway.
- AWS File Gateway allows on-premises servers to store content in S3 buckets using NFSv4 or SMB mount points.
- AWS File Gateway allows on-premises servers to store content in FSx for Windows File Server.
- AWS Volume Gateway Stored mode provides asynchronous replication of on-premises data to Amazon S3.
- AWS Volume Gateway Cached mode stores your primary data in Amazon S3; frequently used data is cached locally.
- Tape Gateway allows you to use your existing tape software and store backups in Amazon S3 storage.

- With AWS Storage Gateway, data transfer is encrypted with SSL/TLS.
- With AWS Storage Gateway, data storage is encrypted with server-side encryption keys (SSE-S3).
- Storage Gateway pricing includes request, data transfer, and storage charges.

Exam Preparation Tasks

As mentioned in the section “[How to Use This Book](#)” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. [Table 12-10](#) lists these key topics and the page number on which each is found.



Table 12-10 [Chapter 12](#) Key Topics

Key Topic Element	Description	Page Number
<u>Table 12-2</u>	Management Service Charges at AWS	598
Section	Tiered Pricing at AWS	599
<u>Table 12-4</u>	Cost Management Tools by Use Case	602
Section	AWS Cost Explorer	604
Section	AWS Budgets	607
Section	AWS Cost and Usage Reports	609
Section	Managing Costs Cheat Sheet	610
Section	Using Cost Allocation Tags	612

Key Topic Element	Description	Page Number
<u>Table 12-5</u>	Amazon S3, EBS, EFS, and FSx for Windows File Server Comparison	616
Section	AWS Backup	618
Section	Lifecycle Rules	619
Section	AWS Backup Cheat Sheet	620
<u>Table 12-6</u>	What Type of Data Do You Need to Transfer from On Premises to AWS?	622
List	Options for moving data from on-premises locations into the AWS cloud	623
Section	AWS Storage Gateway	625

Key Topic Element	Description	Page Number
Section	AWS Storage Gateway Cheat Sheet	627

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

[data transfer](#)

[tiered pricing](#)

[Load Balancer Capacity Unit \(LCU\)](#)

[Cost and Usage Report \(CUR\)](#)

[cost allocation tags](#)

[lifecycle rules](#)

[Snow device](#)

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep Software Online.

- 1.** What are the two main components of calculating management service costs at AWS that are applied to every service cost?
- 2.** How are data transfer costs incurred at AWS?
- 3.** What type of pricing at AWS is calculated based on the usage of the service or resource?
- 4.** What are additional components of storage charges other than data transfer charges?
- 5.** What is the purpose of creating and enabling cost allocation tags?
- 6.** What is the difference between an S3 lifecycle rule and an AWS Backup lifecycle management policy?
- 7.** Where can AWS backups be copied to?

8. What is the difference between an AWS Storage Gateway Volume Gateway and File Gateway?

Chapter 13

Designing Cost-Effective Compute Solutions

This chapter covers the following topics:

- [EC2 Instance Types](#)
- [EC2 Instance Purchasing Options](#)
- [Strategies for Optimizing Compute](#)

This chapter covers content that's important to the following exam domain and task statement:

Domain 4: Design Cost-Optimized Architectures

Task Statement 2: Design cost-optimized compute solutions

There are hundreds of Amazon Elastic Compute Cloud (EC2) instances to consider deploying for a wide variety of workloads. EC2 instances also have a variety of pricing options to consider when deploying compute resources at AWS. EC2 instances can be deployed at AWS and on premises for hybrid deployments.

Recall from [Chapter 2, “The AWS Well-Architected Framework,”](#) that Cost Optimization is one of the six pillars of the AWS Well-Architected Framework. It's an excellent idea to download the

AWS document “Cost Optimization Pillar” (see <https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/wellarchitected-cost-optimization-pillar.pdf>) and read it thoroughly; doing so will help you greatly in understanding how to manage costs at AWS.

“Do I Know This Already?”

The “Do I Know This Already?” quiz enables you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you doubt your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 13-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A](#), “[Answers to the ‘Do I Know This Already?’ Quizzes and Q&A Sections.](#)”

Table 13-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
---------------------------	-----------

EC2 Instance Types	1, 2
--------------------	------

Foundation Topics Section	Questions
EC2 Instance Purchasing Options	3, 4
Strategies for Optimizing Compute	5, 6

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment.

Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What type of EC2 instance provides single tenant protection?
 1. C instances
 2. Dedicated instances
 3. Bare Metal instance
 4. Micro instances

2. How can network performance be improved for an EC2 instance?

1. Add a second network adapter
2. Change to an EC2 instance type that supports enhanced networking
3. Install enhanced networking drivers
4. Change to a general-purpose instance

3. What Reserved instance pricing is the least expensive?

1. Convertible reservation with upfront 3-year payment
2. Standard reservation with upfront payment for 3 years
3. Spot instance pricing
4. Capacity reservation

4. What type of compute instance pricing is the lowest cost?

1. Micro instances
2. Spot instances
3. Reserved instances
4. On-demand

5. What type of zone supports on-premises deployments?

1. Local Zone

2. Availability zone
3. Wavelength Zone
4. AWS Outposts

6. For companies requiring stringent data residency requirements, what is the best deployment choice?

1. Multi-AZ deployment
2. Multi AWS region deployment
3. AWS Outposts deployment
4. Local Zone deployment

Foundation Topics

EC2 Instance Types

EC2 instances are members of several *compute families* grouped and defined using a name and generation designation. In each instance's name, the first letter indicates the family that the instance belongs to (see [Figure 13-1](#)); the family dictates the resources allocated to the instance and the workloads that the instance is best suited for. The letter *c* stands for compute, *r* for RAM, and *i* for input/output operations per second (IOPS).

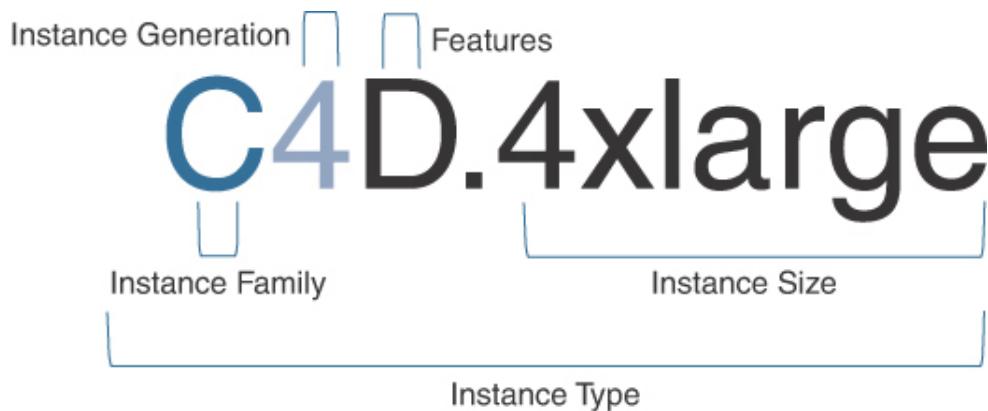


Figure 13-1 Decoding an Instance’s Name

The next number in the instance name is the generation number. This number is very much like a software version number, so a c5 instance is newer than a c4 instance, and so on. (And, interestingly, a newer c5 instance is cheaper than the older c4 instance.)

The next letter, if present, indicates additional features that define the special characteristics of the instance. For example, in c4d, the “d” denotes solid-state drives (SSDs), for instance storage. The last component of the instance’s name deals with the size of the instance; this is sometimes called a *T-shirt size*. Sizes range from small up to 32 times larger than the smallest size. (The size of an instance is based on the number of vCPU cores, the amount of RAM, and the amount of allocated network bandwidth.) For example, c4.8xlarge is eight times larger than c4.Large in terms of vCPU cores, RAM, and network bandwidth.

Note that this example does not have an additional number or letter that would indicate additional features.

When you run a smaller instance at AWS, a smaller portion of the physical server's resources are allocated to the EC2 instance. When you run an x32-sized instance, you could possibly have all the resources assigned. Regardless of the instance type ordered, the allotted memory, vCPU cores, storage, and network bandwidth are isolated for each AWS instance. Customers are virtually isolated from each other, and this isolation is a key element of cloud security.

What Is a vCPU?

AWS defines the amount of CPU power assigned to each instance as a virtual CPU (vCPU). A vCPU is a part of a physical CPU core. A process called *hyperthreading* associates two virtual threads to each physical core—an *a* thread and a *b* thread working in a multitasking mode (see [Figure 13-2](#)). You can think of each physical core as a brain that can be split into two logical brains; a thread is a communication channel that links each instance to a specific amount of processing power. Linux and Windows process these virtual threads differently: The Linux operating system enumerates the first group of “*a*” threads before the second group of “*b*” threads. The Windows operating

system interleaves the threads, selecting the “a” thread and then the “b” thread. Dividing the vCPU count shows the actual physical core count, which might be important if the licensing for your software requires a physical core count (for example, an Oracle database).

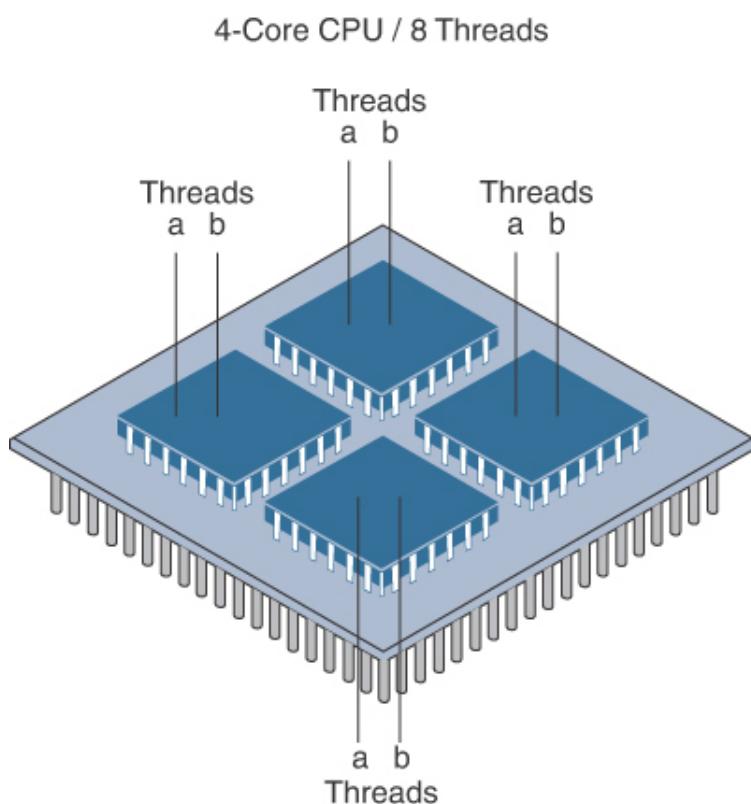


Figure 13-2 Virtual Cores

Note

For further details on core counts, visit
<https://aws.amazon.com/ec2/virtualcores>.

EC2 Instance Choices

The EC2 Dashboard has many instance types to choose from. There are at least 300 types grouped into an ever-increasing number of EC2 instance families, from general-purpose instances to EC2 instance families designed for compute, storage, and memory-optimized workloads (see [Figure 13-3](#)). There are even bare-metal instances available to order. (You do not have to memorize the EC2 instance types and families for the AWS Certified Solutions Architect – Associate [SAA-C03] exam.) When selecting an EC2 instance for a given workload, select the instance family that matches the required vCPU, memory and networking needs at the lowest cost.

Compare instance types								
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and network. You have the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.								
Currently selected: t2.micro								
Instance types (1/606)								
<input type="text"/> Filter instance types								
Instance type	vCPUs	Architecture	Memor...	Storag...	Stor...	Network ...	On-Demand Linux ...	On-Demand Windows pricing
u-12tb1.112xl...	448	x86_64	12288	-	-	100 Gigabit	109.2 USD per Hour	129.808 USD per Hour
u-18tb1.112xl...	448	x86_64	18432	-	-	100 Gigabit	163.8 USD per Hour	184.408 USD per Hour
u-24tb1.112xl...	448	x86_64	24576	-	-	100 Gigabit	218.4 USD per Hour	239.008 USD per Hour
u-6tb1.112xlar...	448	x86_64	6144	-	-	100 Gigabit	54.6 USD per Hour	75.208 USD per Hour
u-9tb1.112xlar...	448	x86_64	9216	-	-	100 Gigabit	81.9 USD per Hour	102.508 USD per Hour
u-3tb1.56xlarge	224	x86_64	3072	-	-	50 Gigabit	27.3 USD per Hour	37.604 USD per Hour
u-6tb1.56xlarge	224	x86_64	6144	-	-	100 Gigabit	46.40391 USD per Hour	56.70791 USD per Hour

Figure 13-3 AWS Instance Choices

When you use the EC2 Dashboard to choose an instance, the initially available choices are defined as the “current generation” instance types. Organizations can still order the original m1 instance that AWS offered in 2006, but it’s not recommended to do so as there are many more powerful and cost-effective options available.

Note

The type of image (AMI) that you use at AWS to deploy your instance is also important to consider. Linux instance types defined as current generation do not support the older paravirtual (PV) images. If you require older PV images, at AWS, you are limited to a smaller number of EC2 instances (c1, c3, hs1, m1, m2, m3, and t1) and a limited number of regions that support PV AMIs, including Tokyo, Singapore, Sydney, Frankfurt, Ireland, São Paulo, North Virginia, Northern California, and Oregon. Windows AMIs support only hardware virtual machine (HVM) images.

An abundance of detailed documentation is available at AWS for EC2 instances at <https://aws.amazon.com/ec2/instance-types>. Here are some common EC2 instance types:

- **General purpose:** General purpose instance types are well suited for a wide range of workloads, including web and application servers, development, and test environments, and small to medium-sized databases. Examples include the m4, m5, and t3 instance types.
- **Compute optimized:** Compute optimized instance types are designed for compute-intensive workloads, such as batch processing, scientific simulations, and high-performance computing (HPC) applications. Examples include the c5, c6g, and c7g instance types.
- **Memory optimized:** Memory optimized instance types are designed for workloads that require high memory-to-vCPU ratios, such as in-memory databases and real-time processing of large data sets. Examples include the r5 and x1e instance types.
- **Storage optimized:** Storage optimized instance types are designed for workloads that require high I/O performance or large amounts of local storage, such as data warehousing, Hadoop, and NoSQL databases. Examples include the d2 and h1 instance types.
- **GPU instances:** GPU instances are designed for workloads that require graphics processing units (GPUs) for tasks such as video transcoding, machine learning, and scientific simulations. Examples include the p2 and g4 instance types.

- **Bare-Metal instances:** For developers who like to host databases on bare-metal servers for maximum performance, a bare-metal server might be an acceptable option to consider. Bare-metal instances were first created for VMware to be able to host ESXi deployments at AWS. Examples include the m5.metal and zlb.metal.

Note

The selected EC2 instance size directly affects your overall network throughput. The larger the EC2 instance, the larger the associated EBS storage and network bandwidth.

Dedicated Host

A Dedicated Host is a physical server with Amazon EC2 instance capacity dedicated to a single customer. A Dedicated Host enables you to use your own existing software licenses—for example, Windows Server or Microsoft SQL Server—and to meet compliance requirements. A Dedicated Host also allows you to control the *affinity*, or placement of your EC2 instances, on the Dedicated Host. Dedicated Hosts support per-socket, per-core, or per-VM software licenses. Here are some benefits to deploying Dedicated Hosts:

- **Cost savings:** Dedicated Hosts can be a cost-effective option for organizations that have many EC2 instances and can take advantage of volume pricing discounts.
- **License compliance:** Dedicated Hosts can help meet licensing requirements for software that requires a specific underlying hardware configuration.
- **Improved security:** Dedicated Hosts can provide an additional layer of security by isolating your instances on physical hardware that is dedicated to your use.

There are some AWS limitations and restrictions when ordering and using Dedicated Hosts:

- The instance size and type of instance placed on a Dedicated Host must be the same type.
- To run RHEL, SUSE Linux, and Microsoft SQL Server on Dedicated Hosts, AMIs must be provided by each customer. RHEL, SUSE Linux, and SQL Server AMIs provided by AWS on AWS Marketplace can't be used with Dedicated Hosts.
- EC2 instances hosted on a Dedicated Host must be launched in a VPC with single tenancy enabled.
- Amazon Relational Database Service (RDS), placement groups, and EC2 Auto Scaling groups are not supported.
- Billing charges are just the hourly charge for each active, dedicated server host; you're not billed for the hosted

instances on the dedicated host. Pricing is based on the on-demand dedicated host price or Reserved instance pricing.

Note

A dedicated host is not the same as a bare-metal server; there is a hypervisor installed on a dedicated host.

Dedicated Hosts Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of dedicated hosts:

- Dedicated hosts are physical servers dedicated completely to your usage and targeting of instances.
- Dedicated hosts are useful for server licenses that require per-core, per-socket, or per-VM metrics.
- Each dedicated host can run one EC2 instance type.
- Billing is per dedicated host.

Dedicated Instances

Organizations may choose to use a dedicated instance if compliance rules and regulations require complete compute

instance isolation for a single virtual server. Each dedicated instance runs in a VPC on hardware resources dedicated to the customer. Dedicated instances have the same performance and security as instances hosted on a dedicated host but also have some limitations to be aware of, including the following:

- No access or control of the sockets and physical cores of the physical host is allowed.
- EBS volumes that are attached to a dedicated instance are standard EBS volumes.

Placement Groups

Amazon EC2 placement groups are logical groupings of EC2 instances within a single AZ. Placement groups are used to ensure that instances are physically isolated from each other within the same AZ.

There are three types of placement groups:

- **Cluster placement groups:** Cluster placement groups group instances that require low network latency and high network throughput. Cluster placement groups are recommended for applications such as HPC, big data, and other applications that require high-performance networking.

- **Spread placement groups:** Spread placement groups are used to distribute instances evenly across distinct hardware. Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other, such as database masters.
- **Partition placement groups:** Partition placement groups are used to group instances across logical partitions so groups of instances in one partition do not share the underlying hardware with groups of instances located in different partitions. Recommended for large, distributed workloads, such as Kafka and Cassandra.

EC2 Instance Purchasing Options

Compute costs (think EC2 instances) are, by default, defined as on-demand or pay-as-you-go (PAYG) and are charged based on the instance's size and AWS region. Compute pricing can be significantly reduced by prepaying compute costs by ordering **Reserved instances**. There are also pricing considerations for dedicated or single-tenancy instances. EC2 instances can also be reserved based on a recurring schedule. There are several other pricing considerations for EC2 instances. For example, there are data transfer pricing differences for using public versus private IP addresses for communication. There are also pricing

differences between instances located within the same AZ on the same subnet and communicating across different AZs.

Selecting an EC2 instance for your workload requirements at the lowest possible cost is the overall goal. Let's consider an example:

A new AWS customer has started initial development and testing in the AWS cloud, matching the EC2 instance size at AWS as closely as possible to the virtual machine size used on premises. However, the virtual machine used on premises is quite large, with many vCPU cores and gigabytes of RAM. The following considerations are important to consider for this scenario:

- Moving to the AWS cloud, the single EC2 instance size at AWS can be smaller because multiple EC2 instances can be deployed on subnets located across multiple AZs hosted behind an ELB load balancer, matching the required compute power with multiple instances and providing high availability and failover.
- Preliminary testing confirms the overall performance of the application stack under a steady-state load with a constant number of users accessing the application hosted on multiple EC2 instances.

- During the initial testing period, compute resources will not be scaled up and down.
- During initial testing and in production, the compute workload environment can be turned off after-hours, or scaled in, when it's not being heavily utilized, reducing compute and data transfer charges.
- Once the application moves from testing to production and requires higher utilization, Reserved instances pricing or Savings Plans could help reduce the compute price up to 70%.
- Once the application is running as a production application, the application load will change from a small number of users to a much larger number of users.
- Include elasticity in the design of the application stack by deploying EC2 Auto Scaling. Automatically scale out or in compute instances based on current user demand. Auto scaled EC2 instances match the end user requirements at any given time, providing the lowest compute costs and the required performance.
- Amazon RDS, and optionally, Amazon Aurora Serverless, has the ability to pause database operation after a defined period of inactivity.

Operating in the AWS cloud and running all resources 24/7 is not cost-effective. Therefore, each customer needs to decide

what services should remain on 24/7 and what services should be turned off or reduced in operating size when not under high demand. [**Table 13-2**](#) outlines options for what services need to always be online and what can possibly be turned off for additional cost savings.

Table 13-2 Service Uptime

AWS Service	On-Premises Operation	At AWS	Cost Savings
DNS servers	24/7	Use Amazon Route 53	Use managed DNS service no servers to manage and administer

AWS Service	On-Premises Operation	At AWS	Cost Savings
Development/testing environments	24/7	Turn off when not being used	14 hour day cost savings
Applications	24/7	EC2 Auto Scaling or Amazon Auto Scaling	Minimize computational resources used

AWS Service	On-Premises Operation 24/7	At AWS	Cost Savings
Databases	Amazon Aurora, Amazon DynamoDB	Amazon Aurora, Amazon Auto Scaling; Amazon Aurora Serverless	14 hour day cost savings

AWS Service	On-Premises Operation	At AWS	Cost Savings
Storage arrays	24/7	Amazon EFS, FSx for Windows File Server, Amazon S3, Amazon EBS	No storage arrays required for administration and management

EC2 Pricing—On-demand

When you first start with AWS, you will use on-demand pricing for your instances. Over time, as you move into production, you will consider a variety of compute pricing options. On-demand pricing involves no long-term contract and requires no upfront payments, but it can be the most expensive pricing option if your EC2 instances are always running. Each EC2 instance also has a specific billing cycle:

- An EC2 instance that is turned on and assigned to your account is billed a compute charge while it is powered on.
- When an EC2 instance is turned off, the billing cycle finishes, and there is no further compute charge. The only additional charges that will be associated with an EC2 instance are for the attached EBS storage volumes and any snapshots or AMIs that have been created. Storage charges at AWS are per month per gigabyte of EBS or S3 storage.

Note

There is no separate charge for EC2 instances with ephemeral storage. The cost for temporary local block storage is included in the price of the EC2 instance.

With on-demand pricing, you pay a flat rate for using resources, and there is no long-term commitment. This model is charged based on an hourly rate, but the increments might be as low as 1 second (for example, for RDS deployments or Linux EC2 instances). For testing purposes or for very short-term usage—perhaps for a 4-hour training class—or for customers first starting out in the AWS cloud, on-demand pricing is fine. The following are other pricing options to consider at AWS:

- On-demand pricing might be best for workloads only running during business hours.
- If you require compute power for applications under constant usage and the application will be running for at least a year, then Reserved instances are a better option than on-demand instances because you will save up to 72%.
- If your application can run any time, spot instances might be a consideration.
- It is also possible to configure a Spot Fleet that uses a combination of on-demand, Spot requests, and Reserved instance pricing. Reserved instances and Spot instances are covered later in this chapter.

On-demand Instance Service Quotas

Once customers have signed up for AWS, they typically think they can spin up as many EC2 instances as desired; however, for every AWS service, there is a default service quota. On-demand EC2 instance quotas are based on the number of vCPUs that on-demand instances have deployed. There are several on-demand instance default service quotas outlined in [Table 13-3](#).

Key Topic

Table 13-3 On-demand Limits Based on vCPUs

On-demand EC2 Instance Type	Default Quota
Running on-demand all standard (a, c, d, h, i, m, r, t, z) instances	1152 vCPUs
Running on-demand all f instances	128 vCPUs
Running on-demand all g instances	128 vCPUs
Running on-demand all inf instances	128 vCPUs
Running on-demand all p instances	128 vCPUs
Running on-demand all x instances	128 vCPUs

At first, the EC2 instance model might seem complicated, but it's rather simple; the default quota is the amount of compute power (vCPU) you are using. Instead of planning limits based on the instance types, you can plan your EC2 instance limits based on the total number of vCPUs used in your workload and AWS account.

For example, with the standard instance quota of 256 vCPUs, you could launch 16 c5.4xlarge instances or any combination of standard instance types and sizes that adds up to 256 vCPUs. It's important to note that current quotas can usually be increased using the Service Quotas utility from the AWS Management Console (see [Figure 13-4](#)).

Request quota increase: All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests X

Quota name
All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests

Description
The maximum number of vCPUs for all running or requested Standard (A, C, D, H, I, M, R, T, Z) Spot Instances per Region

Utilization
0

Applied quota value
512

AWS default quota value
5

Region
US East (N. Virginia) us-east-1

Change quota value:
Enter in the total amount that you want the quota to be. [Learn more](#) 



Must be a number greater than your current quota value

Figure 13-4 Requesting a Quota Change

If you have never communicated with AWS support, how are they going to know what resources you require in the AWS cloud? If you call your on-premises data center staff and request 100 virtual machines, the answer might be, “We can’t right now; we don’t have the capacity.” The same difficulty will arise at AWS: They might not have the capacity or the types of instances that are required. Amazon has a handy calculator called the Limits Calculator that can help you figure out the

number of vCPU views you need (see [Figure 13-5](#)). Open the EC2 dashboard, select an AWS region, and from the menu on the left select Limits. You can enter the following information in the Limits Calculator:

- **Instance type:** The instance type details
- **Instance count:** The number of instances you need



Limits Calculator

Use this tool to calculate how many vCPUs you need to launch your On-Demand Instances

Select the instance type and the number of instances you require. The calculator will display the number of vCPUs assigned to the selected instances. Use the New Limit value as a guide for requesting a limit increase.

Instance type	Instance count	vCPU count	Current limit	New limit
a1.2xlarge <input type="button" value="X"/>	1	8 vCPUs	1,920 vCPUs	1,928 vCPUs <input type="button" value="X"/>
c3.8xlarge <input type="button" value="X"/>	1	32 vCPUs	1,920 vCPUs	1,952 vCPUs <input type="button" value="X"/>

[Add instance type](#)

Limits calculation

Instance limit name	Current limit	vCPUs needed	New limit	Options
All Standard (A, C, D, H, I, M, R, T, Z) instances	1,920 vCPUs	40 vCPUs	1,960 vCPUs	Request on-demand limit increase <input type="button" value="↗"/> Request spot limit increase <input type="button" value="↗"/>

Figure 13-5 The Limits Calculator

The vCPU count column of the Limits Calculator shows the number of CPUs that correspond to the instance count entered in the Limits Calculator.

After calculations are finished, use the links at the bottom right of the Limits Calculator to request a limit increase. If running production workloads need to scale up at a moment's notice,

you should guarantee that the EC2 instances you need are available when required.

Reserved Instances

Reserved instances (RI) are a cost-saving offering that enables you to reserve capacity for your Amazon EC2 instances in exchange for a discounted hourly rate. Reserved instances are automatically applied to running on-demand instances provided that the specifications match.

Once a Reserved instance is ordered, you will be charged the discounted hourly rate, which can be significantly lower than the on-demand rate. With Reserved instances, you pay for the entire term regardless of actual usage. You will be billed for the reserved term whether you run an instance that matches your reservation or not.

For EC2 instances or specific compute-related AWS services that are constantly in use, Reserved instance pricing will save a great deal of money. Organizations need to consider several variables when ordering Reserved instance pricing; for example, the AWS region they are operating in and the specific availability zone location. Note that a c5a.8xlarge EC2 instance is not available in each AZ in the Northern Virginia region,

which has six AZs (see [Figure 13-6](#)). Reserved instance pricing can be ordered for standard 1-year or 3-year durations. A Reserved instance reservation provides a billing discount that applies to EC2 instances hosted in a specific AZ or region. The billing discount could be as high as 72% compared to the standard on-demand hourly rate. Each RI is defined by the following attributes:

- **Instance type:** The instance family and the size of the instance
- **Scope:** The AWS region or availability zone location of the Reserved instance
- **Regional:** The AWS region location of the Reserved instance
- **Zonal:** The AWS availability zone location of the Reserved instance
- **Tenancy:** Shared default hardware or single-tenant, dedicated hardware
- **Platform:** Windows or Linux

Details			
Instance type <input checked="" type="radio"/> c5a.8xlarge	Instance family <input checked="" type="radio"/> c5a	Instance size <input checked="" type="radio"/> 8xlarge	Hypervisor <input checked="" type="radio"/> nitro
Auto Recovery support <input checked="" type="radio"/> true	Supported root device types <input checked="" type="radio"/> ebs	Dedicated Host support -	On-Demand Hibernation support -
Availability zones <input checked="" type="radio"/> us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1f	EBS optimization support <input checked="" type="radio"/> default	Network performance <input checked="" type="radio"/> 10 Gigabit	ENI support <input checked="" type="radio"/> required
Maximum number of network interfaces <input checked="" type="radio"/> 8	IPv4 addresses per interface <input checked="" type="radio"/> 30	IPv6 addresses per interface <input checked="" type="radio"/> 30	IPv6 support <input checked="" type="radio"/> true
Supported placement group strategies <input checked="" type="radio"/> cluster, partition, spread	ENI Express support <input checked="" type="radio"/> false		

Figure 13-6 Availability Zone Availability

Once a purchased RI matches the attributes of a running EC2 instance in your AWS account, the RI is applied immediately. To reiterate: An RI is a billing discount; it is not an EC2 instance; rather, it is a billing discount that you have purchased for a type of EC2 instance.

For applications or web servers that are online and operational 24/7, RI pricing is essential. For example, selecting a c5a.8xlarge instance, the RI discount shown in [Figure 13-7](#) could be as high as 75% when compared to the on-demand instance price.

Purchase Reserved Instances											
Platform		Availability Zone		Tenancy		Offering class					
Linux/UNIX		Any		Default		Standard					
Instance type		Term		Payment option							
c5ad.12xlarge		1 month to 12 months		All upfront						Search	
Seller	Term	Effective rate	Upfront price	Hourly rate	Availability Zone	Payment option	Offering class	Quantity available	Desired quantity		
AWS	12 months	\$1.214	\$10,631.00	\$0.000	us-east-1f	All upfront	Standard	Unlimited	1		Add to cart
AWS	12 months	\$1.214	\$10,631.00	\$0.000	us-east-1d	All upfront	Standard	Unlimited	1		Add to cart

Figure 13-7 Reserved Instance Pricing

Term Commitment

A Reserved instance can be purchased for a 1-year or 3-year commitment; the 3-year commitment provides a larger discount.

Payment Options

Reserved instance pricing has several options to consider. Paying all monies upfront results in the biggest discount (refer to [Figure 13-7](#)).

- **All upfront:** Full payment at the start of the term; no other costs or charges will be incurred for the term.
- **Partial upfront:** A portion of the cost must be paid upfront, and the remaining hours in the term are billed at a discounted hourly rate—*regardless of whether the Reserved instance is being used.*

- **No upfront:** A discounted hourly rate is billed for every hour within the term—*regardless of whether the Reserved instances are being used.*

EC2 Reserved Instance Types



There are two flavors of Reserved instances:

- **Standard Reserved instance:** A standard Reserved instance gives you the biggest discount and can be purchased as repeatable 1-year terms or as a 3-year term. After you've purchased a standard Reserved instance, you can make some changes to your reservation: You can change the AZ where the instance will be hosted, the instance size, and the networking type. What happens if your needs don't match the reservation that was purchased? You can register and try to sell your standard Reserved instance reservation through the Reserved Instance Marketplace.
- **Convertible Reserved instance:** If you may have to change instance types, operating systems, or switch from multi-tenancy to single-tenancy compute operation—then you should consider a convertible Reserved instance reservation.

The convertible reserved discount could be over 50%, and the term is a 1- or a 3-year term. A convertible Reserved instance reservation has more flexibility than a standard Reserved instance reservation because of the additional changes that can be made during the convertible Reserved instances term. However, you cannot sell a convertible reservation in the Reserved Instance Marketplace.

Note

Reserved instance pricing reservations, once expired, do not automatically renew. Billing alerts can be created in the Billing Dashboard to warn when any pricing reservations are due to expire.

Scheduled Reserved EC2 Instances

A scheduled RI reservation allows you to buy capacity reservations for a daily, weekly, or monthly term. The specific length of reservation time that can be requested is a maximum of 1 year. Once instances have been reserved as scheduled, you pay for the reserved compute time, regardless of whether the instances are used. You also can't modify or resell a scheduled instance reservation.

Note

Scheduled instances are supported by c3, c4, c5, m4, and r3 instance types.

Regional and Zonal Reserved Instances

Scope is the important caveat related to the purchase of Reserved instances: The *scope* of the Reserved instance request is regional or zonal.

A Reserved instance for a region is a regional reservation that can be used anywhere in the region.

A zonal Reserved instance involves a discount for a specific AZ within an AWS region. A zonal reservation is also a *capacity reservation* for the selected AZ, in addition to the discounted RI price. Therefore, by purchasing zonal Reserved instances, the capacity—that is, the number of instances you wish to run in a specific AZ is defined.

The Reserved instance price is based on the AWS region in which the instances will be hosted.

- A zonal reservation provides you with a capacity guarantee per AZ as well as a discounted price.

- A regional reservation does not provide you with a capacity reservation; however, it provides flexibility to use the EC2 instances in any AZ.

Table 13-4 lists key differences between regional and zonal Reserved instances.



Table 13-4 Regional Versus Zonal Reserved Instance Reservations

Factor	Regional RI	Zonal RI
Availability zone flexibility	A discount applies to instance usage in any AZ in the region.	A discount applies to instance usage in the specified AZ only.
Reserve capacity	A regional RI does not reserve capacity.	A zonal RI reserves capacity in the specified AZ.

Factor	Regional RI	Zonal RI
Instance size flexibility	A discount applies to any instance within the selected instance family, regardless of size, for Linux instances using default shared tenancy.	A discount applies to instance usage for the specified instance type and size only.
Queuing purchases	Regional RIs can be ordered for a future date and time to ensure that RI coverage continues at the regional level.	RIs cannot be pre-purchased for zonal reservations; zonal reservations apply immediately after purchase.

You can also view your organization's current RI and Savings Plans charges by opening the Billing Dashboard from the AWS

Management Console. To review your current and estimated monthly total, select Bill Details by Service, expand the Elastic Compute Cloud section, and select the AWS region to review current service charges about instances for your AWS account or AWS organization. Costs can also be reviewed by viewing the AWS Cost and Usage Report, and optionally downloading its information in CSV file format.

When purchasing EC2 instances, you need to consider the following factors:

- What AWS region are you going to be operating in?
- How many AZs are you going to use?
- How many EC2 instances do you want to run in each AZ?
- What size of EC2 instance are you planning to run?
- How many EC2 instances need to be running 24/7?
- What are your AWS account limits for each on-demand EC2 instance type required per AWS region?
- Do you need to request a service quota increase for each EC2 instance type to match your needs?
- Do you require a Reserved Instance Standard or Convertible reservation?

Reserved instance pricing provides pricing discounts for many AWS Services that use on-demand instances by default at AWS.

Table 13-5 shows the compute choices where RI can be applied.

Key Topic

Table 13-5 Reserved Pricing Choices with AWS

Reserved Instance Pricing Option	Details
Amazon RDS	Managed database instances
Amazon EMR	Hadoop cluster instances
Amazon ElastiCache	Memcached or Redis clusters
Amazon Redshift	Data warehouse clusters
EC2 instances	On-demand instances

Note

You can also request a capacity reservation for reserving EC2 capacity if you need to guarantee that on-demand instances are always available for use in a specific AZ. This option is not a Reserved instance discount; it is another pricing option. It's important to remember that after you've created a capacity reservation, you will be charged for the capacity reservation whether you actually use the instances or not. However, there are no long-term commitments for a capacity reservation, and the limits can be modified or canceled at any time.

Savings Plans



Savings Plans are a cost savings option that provides discounts on Amazon EC2, AWS Fargate, and AWS Lambda usage in exchange for a commitment to a consistent amount of usage (measured in dollars per hour) for a one- or three-year term. You will be charged the discounted Savings plan price for your

use of resources up to your defined commitment. For example, if you've committed to \$50 of compute usage per hour, the savings plan price for that usage will be charged the commitment amount every hour; any computer usage beyond the defined commitment will be charged the current on-demand rate.

Three types of Savings Plans are available:

- **Compute:** Compute Savings Plans provide discounts on EC2 instance usage across all instance families, sizes, and regions, and on Fargate usage for all regions and AWS compute platforms.
- **EC2 Instance:** EC2 Instance Savings Plans provides savings up to 72% in exchange for a 1- to 3-year commitment to usage of EC2 instance families in a specific AWS region, regardless of availability zone, EC2 instance size, operating system, or tenancy. Customers can change instance sizes if staying within the selected EC2 instance family. EC2 instance usage will be automatically charged at the discounted price; compute usage beyond the per hour commitment will be charged at the current on-demand instance rate. Payment options are all upfront (which provides the best price break), partial upfront, and no upfront. A savings plan also works

with AWS Organizations; benefits are applicable to all AWS accounts within an AWS organization.

- **SageMaker:** SageMaker Savings Plans helps you reduce SageMaker costs by up to 64% regardless of instance family, size, or AWS region.

Note

With consolidated billing, AWS treats all AWS Organization accounts as one account with regard to consolidated pricing. Usage data is combined from all AWS accounts belonging to the AWS organization, applying the relevant volume pricing tier providing the lowest total price on the consolidated resources.

Spot Instances



A *spot instance* is spare compute capacity that AWS is not currently using that is available for much less than Reserved instance pricing. Organizations can potentially save up to 90% of their compute purchase price. However, if and when AWS

takes your spot instance back, it only provides a 2-minute warning, and then—poof—your spot instance is gone. Spot instance pricing is based on EC2 availability, and as just mentioned, a spot instance is available until AWS reclaims it. Spot instances are not guaranteed to always be available; however, they are useful in these use cases:

- **Batch processing:** Spot instances can be used to run batch processing workloads, such as data analysis, machine learning, and video rendering. These types of workloads can be easily interrupted and are often time-sensitive, making spot instances a good choice.
- **Test and development environments:** Spot instances can be used to create test and development environments, where you can test new applications or perform experimentation.
- **High-performance computing (HPC) workloads:** Spot instances can be used to run HPC workloads, such as simulations and modeling, that require a large number of compute resources for a short period of time.
- **Web servers and application hosting:** Spot instances can be used to host web servers and applications, as long as the workload can tolerate the potential for interruption.

Note

Spot instances can be used with EC2 Auto Scaling groups, Elastic Map-Reduce instances (EMR), the Elastic Container Service (ECS), and AWS Batch.

Several terms are used when requesting spot instances:

- **Spot instance pool:** The EC2 instances of the same instance type, operating system, and AZ location that are currently unused.
- **Spot price:** The current per-hour price of a spot instance.
- **Spot instance request:** Request for a spot instance, includes the maximum price you're willing to pay. If you don't specify a maximum price, the default maximum price is the on-demand price. When your maximum spot price is higher than Amazon's current spot price, as long as capacity is available, your spot request will be fulfilled. You can request a spot instance request as a one-time purchase, or as a persistent request; when a spot instance is terminated, Amazon EC2 automatically resubmits a persistent spot instance request, which will remain queued until spot capacity becomes available once again.
- **Spot instances:** The Spot Fleet service evaluates your spot instances request and selects a number of spot instance pools, using available instance types that meet or exceed

your needs and launching enough spot instances to meet the desired target capacity (see [Figure 13-8](#)). Spot Fleets maintain the requested target capacity by default by launching replacement instances after spot instances in the current Spot Fleets are terminated. Note that a Spot Fleet can also include on-demand instances if requested; if your requested criteria cannot be met, on-demand instances should be launched to reach the desired target capacity. If on-demand instances used in the Spot Fleet deployment match a current RI billing discount, the discount is applied to the on-demand instances when they are running.

Required instance attributes

Enter your vCPU and memory compute requirements per instance.

vCPUs

Enter the minimum and maximum number of vCPUs per instance.

8	minimum	12	maximum
---	---------	----	---------

No minimum No maximum

Memory (GiB)

Enter the minimum and maximum GiBs of memory per instance.

12	minimum	16	maximum
----	---------	----	---------

No minimum No maximum

Additional instance attribute - optional

Add additional instance attributes to express your compute requirements in more detail.

Hibernate support	<input type="button" value="▼"/>	<input type="button" value="Add attribute"/>
-------------------	----------------------------------	--

▼ Preview matching instance types (12)

This list includes all the instance types that match your compute requirement. Amazon EC2 may provision capacity from any of these instance types used to fulfill your Fleet request will depend on the allocation strategy you use and available capacity.

<input type="checkbox"/> Instance type	<input type="button" value="▲"/>	vCPUs	<input type="button" value="▼"/>	Memory (GiB)
<input type="checkbox"/> c3.2xlarge		8		15.00
<input type="checkbox"/> c4.2xlarge		8		15.00

Figure 13-8 Selecting Spot Instance Attributes

- **Spot Fleet:** A Spot Fleet is a group of EC2 instances, created from a single request, that share a common set of options. To use a Spot Fleet, you specify the number and type of

instances you want, as well as the maximum price you are willing to pay for each instance type. The Spot Fleet then uses this information to launch the optimal mix of instances to meet your capacity needs at the lowest possible cost. You can also use a Spot Fleet to specify the number of instances you want to maintain in each AZ, enabling you to distribute your workloads across multiple AZs for increased fault tolerance. A Spot Fleet could be helpful if you want to launch a certain number of instances for a distributed application, a long-running batch-processing job, or a Hadoop cluster.

- **Spot Fleet request:** When making a Spot Fleet request, first define the desired total target capacity of your desired fleet and whether you want to use a combination of on-demand and spot instances, or just spot instances. Using on-demand instances provides protection for your workload and ensures that you always have a set amount of capacity available. In [Figure 13-9](#), the Spot Fleet request has, by default, a fleet allocation strategy of maintain target capacity.

Target capacity

Total target capacity
Set your total target capacity (number of instances or vCPUs) to launch. If you specified a launch template, you can allocate part of the target capacity as On-Demand. The number of On-Demand Instances always persists, while Spot Instances can be scaled.

instances ▾

Include On-Demand base capacity
Allocate part of target capacity as On-Demand instances

instances

Maintain target capacity
Automatically replace interrupted Spot Instances

Interruption behavior

Capacity rebalance
When a rebalance notification is sent to a Spot Instance, Spot Fleet automatically attempts to replace the instance before it is interrupted. [Learn More](#)

Instance replacement strategy

Fleet only launches a replacement instance and will not terminate the instance that receives the rebalance recommendation. You can t... ▾

Figure 13-9 Spot Fleet Target Capacity

You can also include multiple launch specifications in the launch template and can further define a number of variables, including the EC2 instance type, AMI, AZ, and subnet to be used. The Spot Fleet service then attempts to select a variety of available spot instances to fulfill your overall capacity request based on your specifications.

Note

The number of spot instances that you can request depends on your defined account spot service

quota limit for the AWS region in which you are operating.

Spot Fleet Optimization Strategies



To optimize the costs of using spot instances, you can deploy several allocation strategies:

- **Lowest price:** This strategy involves deploying the least expensive combination of instance types and availability zones based on the current spot price. This is the default Spot Fleet optimization strategy.
- **Diversified:** This strategy involves distributing spot instances across all available spot pools.
- **Capacity optimized:** This strategy involves provisioning from the most available spot instance pools.
- **Capacity rebalancing:** This strategy involves allowing the Spot Fleet service to replace spot instances that are at risk of interruption with new spot instances.
- **Instance pools to use:** This strategy involves distributing spot instances across the spot pools that you specify.

For Spot Fleets that run for a short period of time, you probably want to choose the lowest price strategy. For Spot Fleets that run for an extended period, you likely want to distribute spot instance services across multiple spot pools. For example, if your Spot Fleet request specifies five pools and a target capacity of 50 instances, the Spot Fleet service launches ten spot instances in each pool. If the spot price for one of the spot pools exceeds your maximum price for this pool, only 20% of your entire fleet is affected.

Note

Spot instances can also be provisioned for other AWS services, including EC2 Auto Scaling and EMR, as well as through the use of CloudFormation templates.

Spot Capacity Pools

To design resiliency with spot instances, you can create spot capacity pools, as shown in [Figure 13-10](#). Each pool is a set of unused EC2 instances that has the same instance type, operating system, and network platform.

Your fleet request at a glance			
Total target capacity 200 instances	Instance configuration template1, v.1 2 vCPUs, 3 GiB (min) 2 Availability Zones	Fleet strength Strong 12 instance pools	Estimated price ~\$24.427/hr at target capacity 36% savings compared to On-Demand

Figure 13-10 Spot Capacity Pools

To ensure that you always have the desired capacity available, even if some of your spot instances are suddenly removed, you can direct the Spot Fleet service to maintain your desired compute capacity by using on-demand instances if there are not enough spot instances available that match your launch specifications. The Spot Fleet service attempts to save you money by launching the lowest-priced instance type it can find —either a spot instance or an on-demand instance. Therefore, your spot capacity pools could contain both spot and on-demand instances, depending on what spot instances are available at the time of your request.

After your fleet is launched, the Spot Fleet service can maintain the desired target compute capacity when there are changes in the spot price or available capacity. The allocation strategy for your defined spot instances is based on Capacity Optimized. Other choices include Price Capacity Optimized or Lowest Price.

You can also choose to distribute the available spot instances across the spot instance pools by selecting Diversified Across All Pools.

Each spot capacity pool can also have a different price point. The built-in automation engine helps you find the most cost-effective capacity across multiple spot capacity pools when requesting a Spot Fleet. Both Linux and Windows operating system instances are available as spot instances. Remember that Spot Fleets operate within the defined service quota limits of your AWS account, which include the number of Spot Fleets per region, the number of launch specifications per fleet, and Spot Fleet target capacity.

Although spot instances can be terminated after a 2-minute warning, according to Amazon's analysis, most spot instance interruptions are due to customers terminating their spot instances when work is completed.

Note

A Spot Fleet cannot span different subnets within the same AZ.

You can choose to have a spot instance hibernated or stopped when it is interrupted instead of just having it terminated.

When your spot instances are hibernated, the data held in RAM is stored on the root EBS drive of the hibernated instance, and your private IP address is held. Spot hibernation is not supported for all instance types and AMIs, so make sure to check the current support levels for hibernated spot instances.

EC2 Pricing Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of EC2 pricing:

- On-demand instances require no long-term commitments but have the highest price.
- On-demand capacity reservations allow you to guarantee that compute capacity is available when you need it. However, you pay for the reservation 24/7 whether you use it or not.
- Reserved instances offer up to 75% savings because you prepay for capacity.

- Zonal Reserved instances have capacity guarantees.
- Regional Reserved instances do not have capacity guarantees.
- A Savings Plan enables you to set a baseline hourly price that you are willing to pay.
- Savings Plans used for EC2 instances have increased flexibility and reduced operational costs.
- Spot instances requests run on spare compute capacity in AWS data centers and can save you up to 80%.
- To obtain a spot instance, you create a spot instance request.
- Spot Fleets can be created specifying the desired number of spot instances to launch to fulfill the capacity request.
- Spot requests can be one-time or persistent requests.
- Spot Fleets attempt to maintain the desired compute instance capacity.
- A Spot Fleet is a collection of different spot instance types and, optionally, on-demand instances.

Compute Tools and Utilities



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following AWS tools and

utilities for assisting in evaluating compute costs and EC2 instance usage:

- **AWS CloudWatch:** Continuous monitoring of EC2 instances using CPU utilization, network throughput, and disk I/O metrics allows customers to observe peak values of each metric to help select the most efficient and cheapest instance type.
- **AWS Cost Explorer:** EC2 Usage Reports are updated several times each day, providing in-depth usage details for all your running EC2 instances.
- **AWS Operations Conductor:** Use recommendations from Cost Explorer to automatically resize EC2 instance.
- **AWS Trusted Advisor:** Inspect and identify underutilized EC2 instances.
- **AWS Compute Optimizer:** AWS Compute Optimizer uses machine learning to recommend optimal AWS resources for your workloads to reduce costs and improve performance. Compute Optimizer helps you choose optimal configurations for EC2 instance types, EBS volumes, and Lambda functions.
- Compute-optimized instances with CPU usage and memory usage less than 40% usage over a one-month period should be rightsized to reduce operating costs.
- Storage-optimized instances IOPS settings should be monitored to make sure EC2 instances are not

overprovisioned IOPS-wise.

- Amazon RDS instance performance baselines should be created and monitored using the RDS metrics Average CPU utilization, Maximum CPU utilization, Minimum available RAM, and Average number of bytes written and read to and from disk per second.
- Steady-state workloads that operate at a constant level over time should be switched to Savings Plans.
- Temporary workloads with flexible start and stop times should be deployed using spot instances instead of On-demand instances.
- Use spot instances for workloads that don't require high reliability.
- Schedule EC2 instances to ensure they run only during business hours using the AWS Instance Scheduler.

Strategies for Optimizing Compute

AWS has greatly increased its hybrid compute options to include AWS Local Zones, Wavelength Zones, and AWS Outposts (see [Table 13-6](#)) to allow customers to run AWS infrastructure and services anywhere.

Table 13-6 Distributed Compute Strategies Processing at the Edge

	CloudFront	Wavelength Zones	AWS Local Zones
Location	Edge cache for static and dynamic data	Hosted 5G applications in third-party data centers	AWS compute, storage, database, and services closer to customers
Latency	Edge location close to the customer	Single-digit ms	Single-digit ms

Use Case	CloudFront Web servers, S3 static data	Wavelength Zones 5G gaming, video streaming from the telco data center	AWS Local Zones High- bandwidth and secure connections between local workloads and AWS
Performance	Better	Faster	Fast

Customers that still remain cautious about moving to the cloud due to latency concerns or compliance regulations may find that AWS Local Zones, AWS Wavelength Zones, or AWS Outposts matches their requirements:

- **AWS Local Zones:** AWS infrastructure including compute, storage, and database services closer to customers in a single data center that can be linked to an existing VPC within an AWS region. Currently, EC2, VPC, EBS, Amazon FSx, Elastic Load Balancing, Amazon EMR, and RDS services can be

deployed in a Local Zone, allowing local applications running in on-premises data centers to have high-speed connections into the AWS cloud.

- **AWS Wavelength Zones:** AWS compute and storage services infrastructure deployed into third-party telecommunication providers' data centers located at the edge of the 5G network (see [Figure 13-11](#)). Applications deployed in a Wavelength Zone data center can locally connect to application servers without leaving the Wavelength Zone. Use cases include gaming, live video streaming, and machine learning.

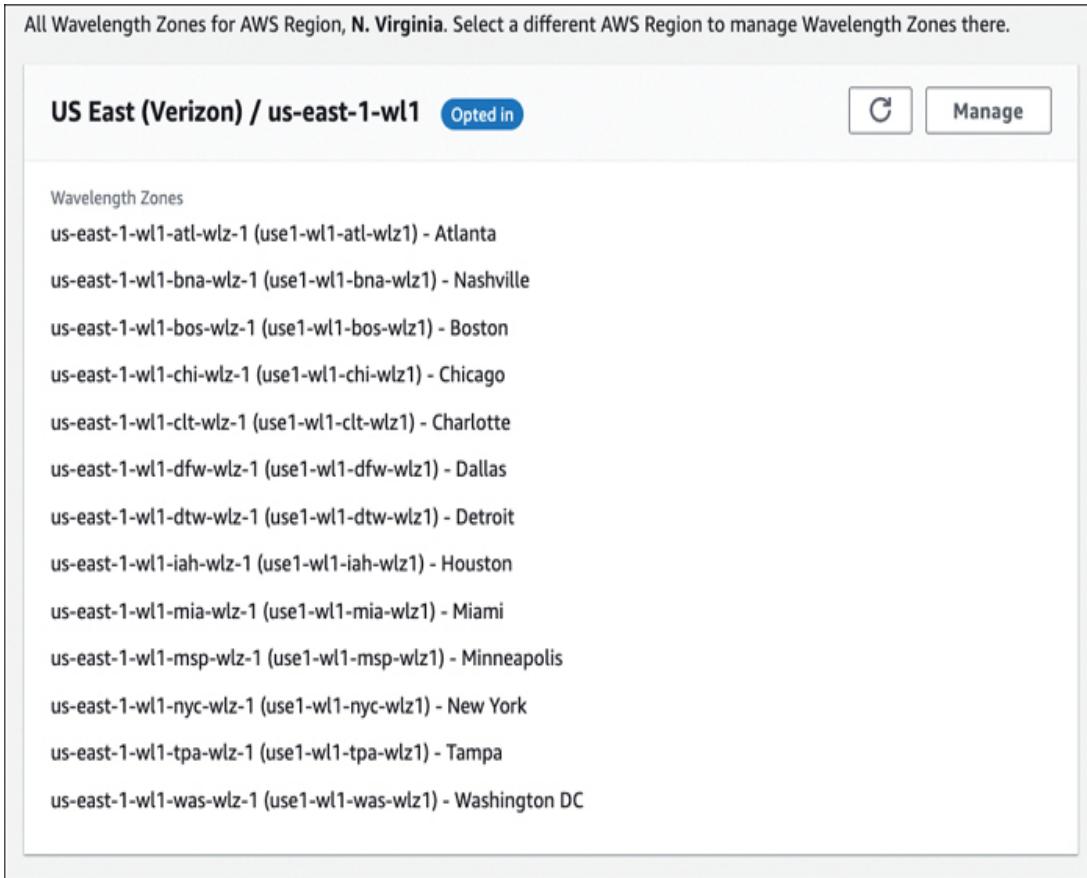


Figure 13-11 Wavelength Zones in US-East-1 (Northern Virginia)

- **AWS Outposts:** AWS Outposts allows companies to run AWS infrastructure services on premises or at co-locations. Available server form factors are 1U/2U Outpost servers on 42U Outpost racks (see [Figure 13-12](#)). A custom VPC can be extended to include an on-premises AWS Outposts location running AWS services locally. Customers can run workloads

on Outpost racks or Outpost servers on premises and connect to any required cloud services hosted at AWS.

- **AWS Outpost racks support the following AWS services locally:** Amazon Elastic Compute Cloud (EC2), Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), Amazon Elastic Block Store (EBS), Amazon EBS Snapshots, Amazon Simple Storage Service (S3), Amazon Relational Database Service (RDS), Amazon ElastiCache, Amazon EMR, Application Load Balancer (ALB), CloudEndure, and VMware Cloud.
- **AWS Outpost servers support the following AWS services locally:** Amazon EC2, Amazon ECS, AWS IoT Greengrass, or Amazon SageMaker Edge Manager.

Outposts catalog

Select a configuration that meets the needs of your application. For pricing information, see the [Outposts rack pricing page](#) and [Outposts server pricing page](#). You can also contact the AWS Outposts team to [request a custom configuration](#).

 Outposts server orders will begin shipping in the second quarter of 2022.

Supported hardware type

Servers

An Outpost form factor that is an industry-standard 1U or 2U server, which can be installed in a standard EIA-310D 19-inch compliant 4-post rack. Outpost servers provide local compute and networking services to sites that have limited space or smaller capacity requirements.

Racks

An Outpost form factor that is an industry-standard 42U rack. Outpost racks include rack-mountable servers, switches, a network patch panel, a power shelf and blank panels.

Figure 13-12 AWS Outposts Options

Matching Compute Utilization with Requirements

AWS has an ever-increasing expansion of compute services for a variety of customer workloads. EC2 instances power the many container offerings available from AWS (see [Table 13-7](#)), or you can choose to deploy bare-metal servers at AWS or on premises.

Table 13-7 Optimization of Compute Workloads

Compute Type	Compute Service	Compute Configuration	Compute Utilization
On-premises	None	None	Low
Cloud	Amazon Lambda	Serverless	Low
Cloud	Amazon EC2	Virtual Machines	Medium

	EC2 Instance	Container Service	ECS	AWS Lambda
Placement	Per EC2	Manual	AWS Lambda	Serverless
Pricing	On-demand, RI, Spot Instances, Savings Plans	On-demand, RI, spot instances, Savings	\$0.01025 per hour for each managed ECS instance	No cost
Scalability	Scaling Groups	Auto Scaling	Autoscaling	Serverless
Reliability	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics
Integration	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics
Management	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics	Amazon CloudWatch Metrics

instance, Placement groups, Auto Scaling groups	Task Scheduling, AWS Fargate, Auto Scaling groups	Fargate with any VM Fargate, VMware, Microsoft Hyper-V, or OpenStack	example projects have been created for each provider.
Location	AZs, Local Zones, AWS Outposts	AZs, Local Zones, AWS Outposts	AZs, Local Zones, AWS Outposts

Compute Scaling Strategies

Depending on the workload being deployed, both vertical or horizontal scaling and hibernation can also be an option (see [Table 13-8](#)):

- Every EC2 instance can be vertically scaled to a larger size EC2 instance, improving the available RAM, storage size and IOPS, and network speeds.

- Auto Scaling groups provide automatic scaling of EC2 instances and containers, minimizing costs while providing the desired performance.
- Spot instances can be set to hibernate when the EC2 service takes back a spot instance.
- Amazon Aurora Serverless deployments can scale up and down and also hibernate after a defined period of inactivity.

Table 13-8 Compute Scaling Strategies

	Vertical Scaling	Hibernation	Auto Scaling	EC2 Auto Scaling
EC2 Instances	Yes	No		
Containers	No	No		Yes
Spot Instances	No	Yes		Yes

	Vertical Scaling	Hibernation	Auto Scaling	EC2 Auto Scaling
Amazon Aurora Serverless v1/v2		Yes	Yes	Yes



Note

Amazon EKS Anywhere allows customers to create and operate Kubernetes clusters on on-premises infrastructure using VMware vSphere.

Note

Amazon Compute Optimizer will identify EC2 instance types, EBS volume configurations, and Amazon Lambda function memory sizes, using machine learning to analyze historical utilization metrics. AWS Compute Optimizer also integrates with AWS Organizations for recommendations within the organization.

Exam Preparation Tasks

As mentioned in the section “[How to Use This Book](#)” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” compute resources at AWS, and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. [Table 13-9](#) lists these key topics and the page number on which each is found.



Table 13-9 [Chapter 13](#) Key Topics

Key Topic Element	Description	Page Number

Key Topic Element	Description	Page Number
<u>Table 13-3</u>	On-demand Limits Based on vCPUs	642
<u>Figure 13-5</u>	The Limits Calculator	643
Section	EC2 Reserved Instance Types	646
<u>Table 13-4</u>	Regional Versus Zonal Reserved Instance Reservations	647
<u>Table 13-5</u>	Reserved Pricing Choices with AWS	648
Section	Savings Plans	649
Section	Spot Instances	650

Key Topic Element	Description	Page Number
Section	Spot Fleet Optimization Strategies	653
Section	EC2 Pricing Cheat Sheet	655
Section	Compute Tools and Utilities	655

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

[EC2](#)

[Reserved instance](#)

[zonal](#)

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep

Software Online.

- 1.** AWS defines the amount of CPU power assigned to each instance as a _____.
- 2.** A Dedicated Host is a physical server with _____ capacity dedicated to a single customer.
- 3.** Dedicated hosts support per-socket, per-core, or _____ software licenses.
- 4.** An EC2 instance that is turned on and assigned to your account is billed a _____ while it is powered on.
- 5.** Reserved instance pricing and Savings Plan pricing provide _____ for the on-demand instances.
- 6.** A scheduled RI reservation allows you to buy _____ for a daily, weekly, or monthly term.
- 7.** A Reserved instance for a region is a _____ that can be used _____ in the region.
- 8.** A zonal Reserved instance involves a discount for a specific _____ within an AWS region.

Chapter 14

Designing Cost-Effective Database Solutions

This chapter covers the following topics:

- [Database Design Choices](#)
- [Database Data Transfer Costs](#)
- [Data Retention Policies](#)

This chapter covers content that's important to the following exam domain and task statement:

Domain 4: Design Cost-Optimized Architectures

Task Statement 3: Design cost-optimized database solutions

Relational database choices provided by AWS include the Amazon Relational Database Service (RDS), which includes a database engine for deploying Oracle, MySQL, Microsoft SQL Server, PostgreSQL, MariaDB, and Amazon Aurora on EC2 instances. Each supported database engine has a predefined schema for the table of rows and columns and a key that uniquely identifies each row in the table. Each database is launched within a controlled VPC and typically hosted on

private subnets. Storage is provided by Amazon Elastic Block Store (EBS) SSD volumes (gp2, io1, or io) with a defined amount of IOPS.

Amazon Web Services (AWS) offers a number of nonrelational databases that can be used to store and manage data. Some examples include Amazon DynamoDB, a fast and flexible NoSQL database service that can be used to store and retrieve any amount of data; Amazon DocumentDB, a fast, scalable, and fully managed document database service that is compatible with the MongoDB API—it allows you to store, retrieve, and manage document-oriented data; Amazon Neptune, a fully managed graph database service that makes it easy to build and run applications that work with highly connected data.

“Do I Know This Already?”

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 14-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions.

You can find the answers in [Appendix A](#), “[Answers to the ‘Do I Know This Already?’ Quizzes and Q&A Sections.](#)”

Table 14-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Database Design Choices	1, 2
Database Data Transfer Costs	3, 4
Data Retention Policies	5, 6

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which of the following is the most expensive SQL database deployment scenario?

1. RDS Oracle
2. Manual SQL deployment across multiple AZs
3. RDS MySQL
4. RDS PostgreSQL

2. What RDS database engine can be deployed as a multi-region global datastore?

1. Microsoft SQL Server
2. Oracle
3. Amazon Aurora
4. Amazon DynamoDB

3. A Multi-AZ custom-deployed Microsoft SQL database is charged for what type of data transfer cost?

1. Snapshot backup
2. Read replica queries within the AZ
3. Updates to the primary database instance
4. Database replication across AZs

4. What data transfer cost is always free?

1. Communication within an availability zone
 2. Data transfer from the Internet to AWS
 3. Replication across AWS regions
 4. Replication within a region
- 5.** RDS automatic snapshot retention policies can be set up for how many days?
1. 7 days
 2. 14 days
 3. 21 days
 4. 35 days
- 6.** Point-in-time recoveries can restore data to what degree of precision?
1. To the second
 2. To the minute
 3. To the hour
 4. To the day

Foundation Topics

Database Design Choices

Production databases should be designed with a minimum of two database servers running in separate availability zones within the same AWS region.

However, when operating in the cloud, customers should always plan to deploy and maintain at least three separate copies of data. When using Amazon RDS as the database solution, at a minimum a primary database and a standby database are both kept up to date with synchronous replication. RDS disaster recovery is managed through scheduling automatic snapshots, and transaction logs are backed up every 5 minutes. Customers must make decisions about the desired resiliency, failover, and recovery of their database records and manage the overall costs of their database operations.

RDS Deployments

As mentioned, Amazon RDS deployments are SQL deployments that can be deployed in a single region or across multiple regions. The engine of RDS deployments, excluding Aurora Serverless, is defined by EC2 instances. DB instance classes supported by RDS include general-purpose and memory-optimized instances. Deployment options for RDS instances are on-demand instances with either Reserved Instance reservation or Savings Plan. Microsoft SQL licensing is bundled with the

RDS database instance cost; customers that want to bring their own Microsoft SQL license to AWS must build a custom RDS SQL deployment. RDS Oracle deployments also have a BYOL option for On-demand DB instances. [Table 14-2](#) compares the available RDS deployment options, Amazon Redshift, and Amazon ElastiCache deployment options including workload use cases, performance, backup options, and cost management.



Table 14-2 AWS Database Service Comparison

Database Engine	Amazon RDS	Amazon Aurora	Amazon ElastiCache
MySQL	MySQL	MySQL	Redis, Memcached, Amazon ElastiCache for Redis, Amazon ElastiCache for Memcached
Oracle	Oracle	Oracle	Redis, Memcached, Amazon ElastiCache for Redis, Amazon ElastiCache for Memcached
PostgreSQL	PostgreSQL	PostgreSQL	Redis, Memcached, Amazon ElastiCache for Redis, Amazon ElastiCache for Memcached
Microsoft SQL Server	Microsoft SQL Server	Microsoft SQL Server	Redis, Memcached, Amazon ElastiCache for Redis, Amazon ElastiCache for Memcached
Amazon Redshift	Amazon Redshift	Amazon Redshift	Amazon Redshift
Amazon Neptune	Amazon Neptune	Amazon Neptune	Amazon Neptune
Amazon DynamoDB	Amazon DynamoDB	Amazon DynamoDB	Amazon DynamoDB

Database Engine	Amazon RDS	Amazon Aurora	Amazon Aurora Serverless
Compute	EC2 instances	EC2 instances	Serverless Compute

Replication	Multi-AZ cluster deployment of one primary and two read replicas	Across three AZs	Across five AZs
--------------------	--	------------------	-----------------

Data type	SQL	PostgreSQL, MySQL	PostgreSQL, MySQL
------------------	-----	-------------------	-------------------

Database Engine	Amazon RDS	Amazon Aurora	Amazon Aurora Serverless
Read replicas	5	Up to 15	Up to 15
Workload	Transactional (Simple)	Analytical (simple/parallel queries)	Infreq used applic Test Deploy
Regional	Yes	Yes	Yes

Database Engine	Amazon RDS	Amazon Aurora	Amazon Aurora Serverless
Multi-region	No	Global	No
		Datastore, storage-based replication (< 1 second), secondary region	
Performance	EC2 instance, EBS volume size	Five times of RDS, parallel query	Scaled transactions to match requirements

Database Engine	Amazon RDS	Amazon Aurora	Amazon Aurora Serverless
Auto Scaling	Yes	In 10-GB chunks	In 10-GB chunks
Storage			

Auto Scaling	No. Manual compute sizing	Yes	Yes
---------------------	---------------------------	-----	-----

Backup options	Snapshots, manual snapshots	Automatic, continuous, incremental, S3, point-in-time restore, manual snapshots, Backtrack	Automatic, continuous, incremental, S3, point-in-time restore, manual snapshots, Backtrack
-----------------------	-----------------------------	--	--

Database Engine	Amazon RDS	Amazon Aurora	Amazon Aurora Serverless
Cost management	On-demand, Reserved Instances, Savings Plans, data transfer costs	Provisioned on-demand, reserved instances, database storage and I/O charges, data transfer costs, globally replicated read-writes	Serverless capacity units



RDS Costs Cheat Sheet

**Key
Topic**

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following about RDS EC2 instance usage, EBS storage and IOPS, scheduled backups, and data transfer costs:

- PostgreSQL, MySQL, and MariaDB have similar costs for storage, provisioned IOPS, and data transfer costs.
- AWS RDS costs are specific to the AWS region of deployment.
- Oracle and Microsoft SQL Server deployments can be double the price due to licensing fees.
- Amazon Aurora can be deployed using provisioned EC2 instances or serverless compute.
- Amazon Aurora can be deployed as a multi-region Global Datastore.
- AWS RDS instance pricing includes vCPU, RAM, and network speeds per chosen RDS database instances.
- AWS RDS Reserved instances can save up to 60% in compute costs for 1 to 3 years.
- Auto-provisioning with a defined Amazon Aurora storage maximum is for unpredictable storage needs.
- AWS RDS snapshot backups are free and are performed every day.

- AWS RDS retention periods determine how many automatic backups are kept. The default is 7 days; the maximum is 35 days.
- AWS RDS Multi-AZ deployments create a standby database instance with a separate replicated database instance.
- On-demand RDS instances can be stopped for 7 days, during which time compute is not charged but EBS storage volumes are still charged.
- Data transfer costs are charged when data exits the source location and enters the target location, AZ-to-AZ, or Region-to-Region.
- The amount of retained backup storage can be lessened by reducing the backup retention period.
- Manual snapshots created by customers are never removed from storage.

Note

CloudWatch metrics for RDS that can help monitor database instance costs include network usage, CPU utilization, and memory utilization.

RDS Database Design Solutions

Consider these design possibilities when choosing a managed database design solution:

- **Reserved instance pricing:** RDS deployments and provisioned versions of Amazon Aurora where customers choose the compute size can be powered by reserved instances to save money. On-demand and spot instances should not be used for database instances that are always online; on-demand instances may be too expensive for 24/7 operation, and spot instances are not guaranteed to be always available.
- **RDS Cluster deployment:** Creates a DB cluster with a primary DB instance and two readable standby DB instances (see [Figure 14-1](#)). RDS DB instances are located in different AZs, providing high availability, data redundancy, and increased query capacity.

Availability and durability

Deployment options [Info](#)
The deployment options below are limited to those supported by the engine you selected above.

Multi-AZ DB Cluster - new
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

Multi-AZ DB instance
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

Single DB instance
Creates a single DB instance with no standby DB instances.

Figure 14-1 RDS Cluster Deployment Choices

- **RDS across multiple availability zones:** RDS high-availability and failover designs are especially effective for database deployments. In addition, durability can be provided by keeping primary and alternate database instances up to date with synchronous replication. When you deploy RDS solutions, AWS does not charge for replication between database instances located in different AZs (see [Figure 14-2](#)); when you deploy custom EC2 instances across multiple AZs with a custom database design, there will be data transfer charges for the replication between the primary and alternate database instances across separate AZs and regions.
- **RDS in a single availability zone:** A single AZ does not have any high-availability or failover options because the single database server is on a single subnet. High availability or failover might not be a concern due to prudent planning and backup procedures. Perhaps hourly snapshots and transaction logs are automatically created on a schedule and backed up into multiple S3 buckets hosted in different AWS regions. If your recovery time objective (RTO) allows you to be down for a longer period of time (for example, 6 hours), a single AZ deployment may be more economical than a multi-AZ design.

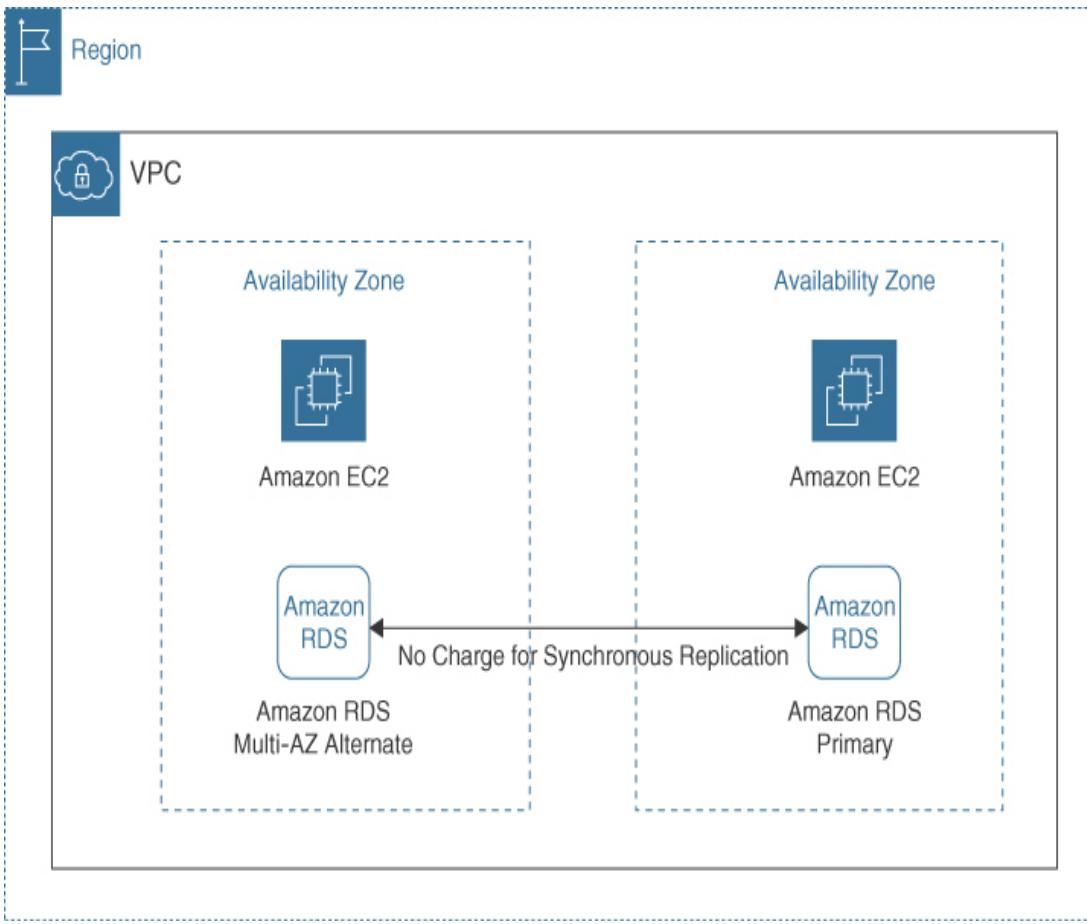


Figure 14-2 RDS Multi-AZ Deployment

- **Manual snapshots:** Snapshots can be created from RDS EBS volumes at any time. RDS deployments automate snapshots created based on a schedule; however, manual snapshots can also be created and stored in other AWS regions for safekeeping, allowing you to rebuild any EC2 instance (web, application, database, or software appliance).
- **Managing snapshots:** Without any long-term management, long-term storage of snapshots is expensive. The Amazon

Data Lifecycle Manager allows you to create lifecycle policies to schedule the creation and deletion of EBS snapshots.

- **Read replicas:** A *read replica* is a copy of the primary database that is kept up to date with asynchronous (rather than synchronous) replication. Read replicas can be promoted to a standalone RDS instance as part of a manual disaster recovery solution if the primary RDS database fails (see [Figure 14-3](#)). MySQL, MariaDB, Oracle, and Microsoft SQL Server read replicas can be promoted and made writable, whereas a PostgreSQL read replica cannot be made writable. Read replicas provide the ability to scale read traffic horizontally and also provide additional durability for the database tier.

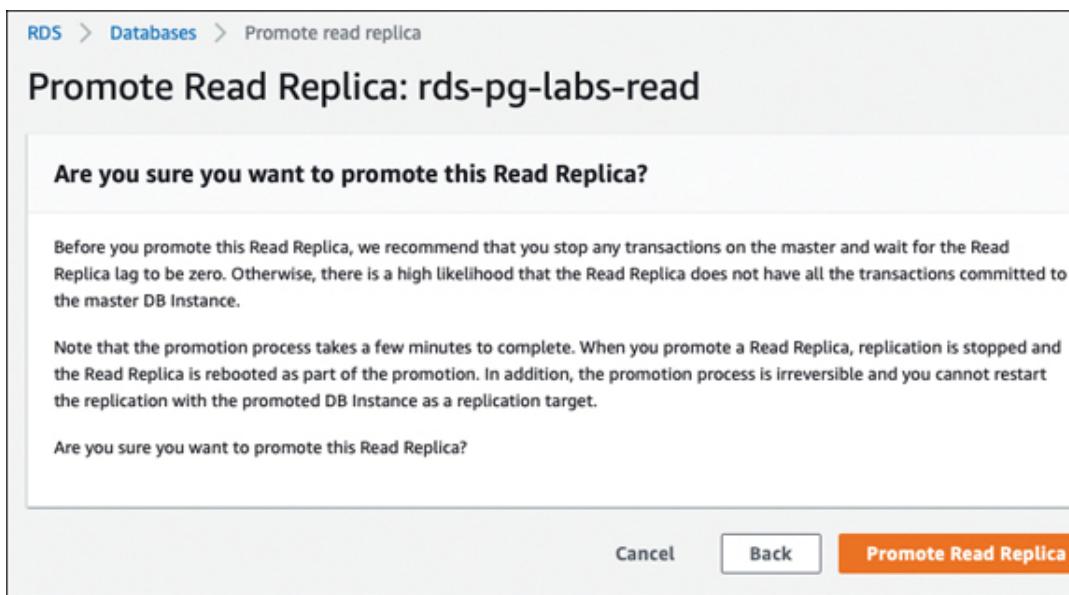


Figure 14-3 Promoting Read Replicas

- **Amazon Aurora Serverless:** Deploying Aurora Serverless allows you to pay for only the minimum capacity database resources (LCU and database storage) that you consume. You can choose either of two serverless versions:
 - Version 1 is for a single-region database deployment of a MySQL- or PostgreSQL-compatible engine. The term *capacity unit* refers to the amount of compute and memory resources assigned. Read or write capacity units (RCU/WCU) can be set from 1 capacity unit, which selects a minimum capacity of 2 GB of RAM, up to 256 capacity units, which selects 488 GB of RAM. Based on the minimum and maximum capacity unit settings, auto-scale compute capacity read/write rules are defined for the required CPU utilization, the number of connections, and required memory. When the workload for a serverless Amazon Aurora database drops below the maximum defined capacity unit threshold, Amazon Aurora automatically reduces the CPU and RAM resources made available for the database cluster.
 - Amazon Aurora version 2 can scale from hundreds to thousands of transactions completed in milliseconds both within and across AWS regions. In the background, the auto scaling service uses step scaling to ensure the proper sizing of database resources at any given time.

- Serverless deployments of Amazon Aurora v2 can be paused after a defined period of activity (see [Figure 14-4](#)). There are no charges for an Amazon Aurora database instance when it is not running, potentially saving a high percentage of your database costs when compared to a provisioned deployment of Amazon Aurora.

Capacity settings

This billing estimate is based on published prices. [Learn more](#)

Minimum Aurora capacity unit Info 2 4GB RAM	Maximum Aurora capacity unit Info 384 768GB RAM
---	---

▼ Additional scaling configuration

Force scaling the capacity to the specified values when the timeout is reached [Info](#)
Enable to force capacity scaling as soon as possible. Disable to cancel the capacity changes when a timeout is reached

Pause compute capacity after consecutive minutes of inactivity [Info](#)
You are only charged for database storage while the compute capacity is paused

0 hours	5 minutes	0 seconds
---------	-----------	-----------

Max: 24 hours

Figure 14-4 Pausing Amazon Aurora Deployments

NoSQL Deployments

NoSQL databases have several advantages when compared to SQL databases, including horizontal scaling to a cluster of

servers, built-in high availability, and support for several flexible data structures:

- **Key-value stores:** Data is defined as a collection of **key-value** pairs with an attribute name and an associated value, for example, *<company name-location>*. Use cases include gaming applications and high-traffic applications.

The most popular NoSQL deployment at AWS is DynamoDB, which supports both document and key-value data structures for both regional and global deployments for IoT, mobile, web, and gaming workloads. The main features of DynamoDB are single-digit millisecond data access, serverless deployment with no servers to manage, auto-scaling to handle any spikes, and automatic data encryption by default. DynamoDB has two pricing models:

- **On-demand capacity:** Charges on a per request basis for reading and writing requests to the associated DynamoDB table.
- **Provisioned capacity:** Billing is charged hourly for utilized read and write capacity units and the maximum amount of resources required by each database table. Provisioned capacity should be selected when your workload's application traffic is consistent and the maximum workload for your application is known. [**Table 14-3**](#) compares the available NoSQL deployment options,

including workload use cases, performance, backup options, and managing costs.

- **Graph:** Graph data is composed of a set of nodes and a set of edges connecting the nodes. For example, *mammal*, the node and *shark*, the edge relationship would be defined as “is a type of fish.” Use cases include fraud detection and recommendation engines. AWS service: Amazon Neptune.
- **Wide-column:** Uses tables and rows and columns just like a relational database; however, the names and the format of each column can vary from row to row within the same table. Use cases include written optimization and fleet management. AWS Service: Amazon Keyspaces (for Apache Cassandra).
- **Document:** Data is stored in a document using a standard encoding such as JSON, XML, or YAML. Use cases include user profiles and content management. AWS services: DynamoDB, DocumentDB.
- **Times series:** Data is stored in time-ordered streams. Nodes contain individual data values; edges are the relationships between the data values. Use cases include social networking and recommendation engines. AWS service: Amazon Timestream.
- **Ledger:** Based on logs that record events that are related to specific data values. Ledger logs can be verified

cryptographically, proving the authenticity and integrity of the data. Use cases for ledger databases include banking systems, supply chains, and blockchain. AWS service: Amazon Quantum Ledger Database (QLDB).

NoSQL Costs Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following about determining database costs for NoSQL deployments:

- For managing consistent workload usage, choose provisioned mode DynamoDB tables with auto scaling enabled to handle expected changes in demand.
- An on-demand DynamoDB table costs more than provisioned tables.
- A DynamoDB table uses auto scaling to manage the read and write capacity units assigned to each table (see [Figure 14-5](#)). Each scaling policy defines what should be scaled—read or write capacity or both—and the maximum provisioned capacity unit settings for the DynamoDB table. The defined auto scaling policy also defines a target utilization; default

utilization is set at 70% target utilization, and utilization values can be defined between 20% and 90% for both read and write capacity units.



Table 14-3 AWS NoSQL Database Service Comparisons

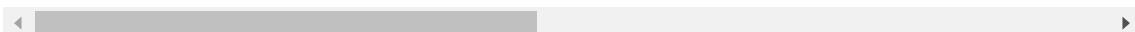
Database Engine	Amazon DynamoDB	Amazon S3	Amazon Kinesis
Compute	Serverless	Serverless storage	Server array

Database Engine	Amazon DynamoDB	Amazon S3	Amazon RDS
Datatype	Transactional Key-Value/Document store supports ACID transactions	Key-Value Hybrid backup, Database backup, Tape backup (Storage Gateway)	Relational PostgreSQL
Use case	Mobile applications	Object storage	Route 53
Read replicas/cache	ElastiCache, DynamoDB Accelerator (DAX)	—	—
Regional	Across three AZs	Across three AZs	Across three AZs

Database Engine	Amazon DynamoDB	Amazon S3	Amazon Kinesis
Multi-region	Yes	Yes	No
Performance	Scale to 10 trillion requests per day over petabytes of storage	First-byte latency retrieval in ms	On-demand AWS Lambda scale to application
Scaled storage	Yes	Yes	Provisioned
Scaled compute	Provisioned throughput	Yes	Provisioned through

Database Engine	Amazon DynamoDB	Amazon S3	Amazon Point-in-time Recovery
Backups	On-demand backups, AWS Backup, point-in-time recovery—restore to any given second	AWS Backup	recovery to any given second

Database Engine Pricing	Amazon DynamoDB	Amazon S3 Storage	Amazon Write I
Standard table, standard	DynamoDB Standard table, standard	S3 Storage classes, S3	Write units/r
Infrequent access	Infrequent access	Intelligent-tiering, S3	units, cost
On-demand capacity (data read/writes charged)	On-demand capacity (data read/writes charged)	Glacier (instant, flexible), Deep	out costs
Provisioned capacity, data transfer costs	Provisioned capacity, data transfer costs	archive, data transfer	costs



Auto Scaling

<input checked="" type="checkbox"/> Read capacity	<input checked="" type="checkbox"/> Write capacity
<input type="checkbox"/> Same settings as read	
Target utilization 70 %	70 %
Minimum provisioned capacity 5 units	5 units
Maximum provisioned capacity 40000 units	40000 units
<input checked="" type="checkbox"/> Apply same settings to global secondary indexes	<input checked="" type="checkbox"/> Apply same settings to global secondary indexes

! Please check your IAM permissions to create new service linked role for enabling Auto Scaling.
See [permissions](#).

IAM Role I authorize DynamoDB to scale capacity using the following role:

DynamoDB AutoScaling Service Linked Role

Existing role with pre-defined policies [\[Instructions\]](#)

Figure 14-5 DynamoDB Scaling Policy

- Provisioned capacity deployments with a high amount of read and write capacity units should purchase reserved capacity for one or three years.
- Automatic DynamoDB backups use additional write capacity units, raising costs, but backups provide valuable high availability.
- One [read capacity unit \(RCU\)](#) performs one strongly consistent read request per second for items up to 4 KB in size, and two eventually consistent read requests. Transactional read requests require two RCUs.

- One *write capacity unit (WCU)* performs one standard rate request per second of items up to 1 KB in size. Transactional write requests require two WCUs.
- Replicated WCUs are used with global tables. Replicated write requests are automatically written to multiple AWS regions.
- To reduce workload costs, host DynamoDB deployments in AWS regions with the lowest operating cost if possible.
- Unnecessary data can be purged using the Time to Live (TTL) feature.
- Queries for data stored in DynamoDB use the primary or index key and only charge for RCUs for the items returned.
- Scans for data stored in a DynamoDB table are much more expensive as you are charged for all rows scanned regardless of how many items are returned.
- Store infrequently accessed data in standard infrequent access DynamoDB tables.
- Use DynamoDB metrics and CloudWatch to monitor usage and storage trends.
- Use the Trusted Advisor Amazon RDS Idle DB instances check to identify DB instances with no connection over the last seven days.

Migrating Databases

Key Topic

Database migration can be carried out using AWS Database Migration Service (DMS). DMS performs a live migration into AWS with little to no downtime. Databases to be migrated can be hosted on an EC2 or RDS instance or located on premises. The DMS server is an EC2 instance hosted in the AWS cloud running replication/migration software. Source and target connections inform DMS where to extract the source database from, and where to deploy the migrated database (see [Table 14-4](#)). Scheduled tasks run on the DMS server and replicate the database from the source to the destination server location. DMS can also create database tables and associated primary keys if these items don't exist on the target instance.

Table 14-4 Database Migration Service Source and Destination Migrations

DMS On Premises, EC2 Instances, Third-Party Cloud	DMS Target (On Premises/EC2 Database Instances)
---	--

DMS On Premises,
EC2 Instances,
Third-Party Cloud

DMS Target (On Premises/EC2
Database Instances)

Oracle Database
10.2 up to 11g and
up to 12.2, 18c, and
19c Enterprise,
Standard, Standard
One, Standard Two
editions

Oracle Database 10g, 11g, 12c,
18c, and 19c Enterprise,
Standard, Standard One,
Standard Two editions

Microsoft SQL
Server 2005–2019
Enterprise,
Standard,
Workgroup, and
Developer Editions

Microsoft SQL Server versions
2005, 2008, 2008R2, 2012, 2014,
2016, 2017, and 2019 Enterprise,
Standard, Workgroup, and
Developer editions (Web and
Express editions not supported)

MySQL versions 5.5,
5.6, 5.7, and 8.0

MySQL versions 5.5, 5.6, 5.7,
and 8.0

DMS On Premises, EC2 Instances, Third-Party Cloud	DMS Target (On Premises/EC2 Database Instances)
PostgreSQL version 9.4 and for versions 9.x, 10.x, 11.x, 12.x, 13.x and 14.0	PostgreSQL version 9.4 and later (for versions 9.x), 10.x, 11.x, 12.x, 13.x, and 14.0
MongoDB versions 3.x, 4.0, 4.2, and 4.4.	DynamoDB
SAP Adaptive Server Enterprise (ASE) versions 12.5, 15, 15.5, 15.7, 16, and later	SAP Adaptive Server Enterprise (ASE) versions 15, 15.5, 15.7, 16, and later
IBM Db2 z/OS for Linux, UNIX, and Windows	Aurora MySQL, Aurora PostgreSQL, MySQL, and PostgreSQL

DMS On Premises, EC2 Instances, Third-Party Cloud	DMS Target (On Premises/EC2 Database Instances)
Microsoft Azure SQL Database	Microsoft SQL Server versions 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, and 2019 Enterprise, Standard, Workgroup, and Developer editions (Web and Express editions not supported)
Google Cloud for MySQL	MySQL versions 5.6, 5.7, and 8.0
RDS instance databases (Oracle, Microsoft SQL Server, MySQL, PostgreSQL, MariaDB), Amazon Aurora (PostgreSQL/MySQL)	Amazon RDS instance databases Amazon Redshift, Amazon DynamoDB, Amazon S3, Amazon OpenSearch Service, Amazon ElastiCache for Redis, Amazon Kinesis Data Streams, Amazon DocumentDB, Amazon Neptune, and Apache Kafka

DMS On Premises,
EC2 Instances,
Third-Party Cloud

DMS Target (On Premises/EC2
Database Instances)

Amazon
DocumentDB

Amazon DocumentDB

Amazon Redis,
Microsoft SQL,
NoSQL

Amazon Redis

Note

AWS Database Migration Service charges for the compute resources used by the replication instance during the migration. Ingress data transfer into AWS is free of charge. Migrating from a supported target to Amazon Aurora, Amazon Redshift, Amazon DynamoDB, and Amazon DocumentDB is free for up to six months.

AWS Schema Conversion Tool

**Key
Topic**

The AWS Schema Conversion Tool (AWS SCT) converts an existing database schema from one database engine to another (see [Figure 14-6](#)). The converted schema can be used with the following Relational Database Service engines: MySQL, MariaDB, Oracle, SQL Server, PostgreSQL DB, Amazon Aurora DB cluster, or Amazon Redshift cluster. The AWS SCT can convert tables, indexes, and application code to the target database engine. After a schema is converted, it is not immediately applied to the target database. Within the AWS SCT project, you can review and make changes to the converted schema before applying the converted schema to the target database.

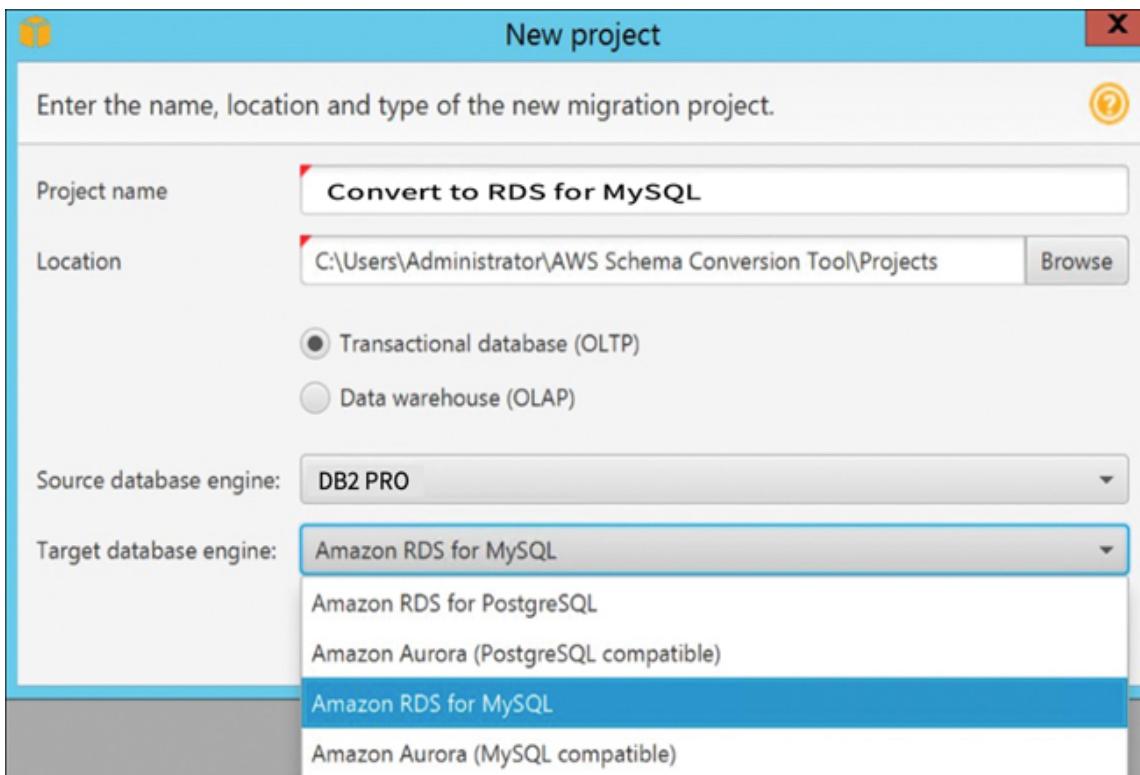


Figure 14-6 AWS Schema Conversion Tool

Database Data Transfer Costs

Data transfer costs are calculated differently for the various managed database services that can be deployed at AWS. As a reminder, there is no charge for inbound data transfer for all services in all regions at AWS (see [Figure 14-7](#)). Transferring data from an Amazon resource across the Internet results in charges for each AWS service based on the region where the service is hosted. Outbound data transfer is currently charged at \$0.09 per GB.

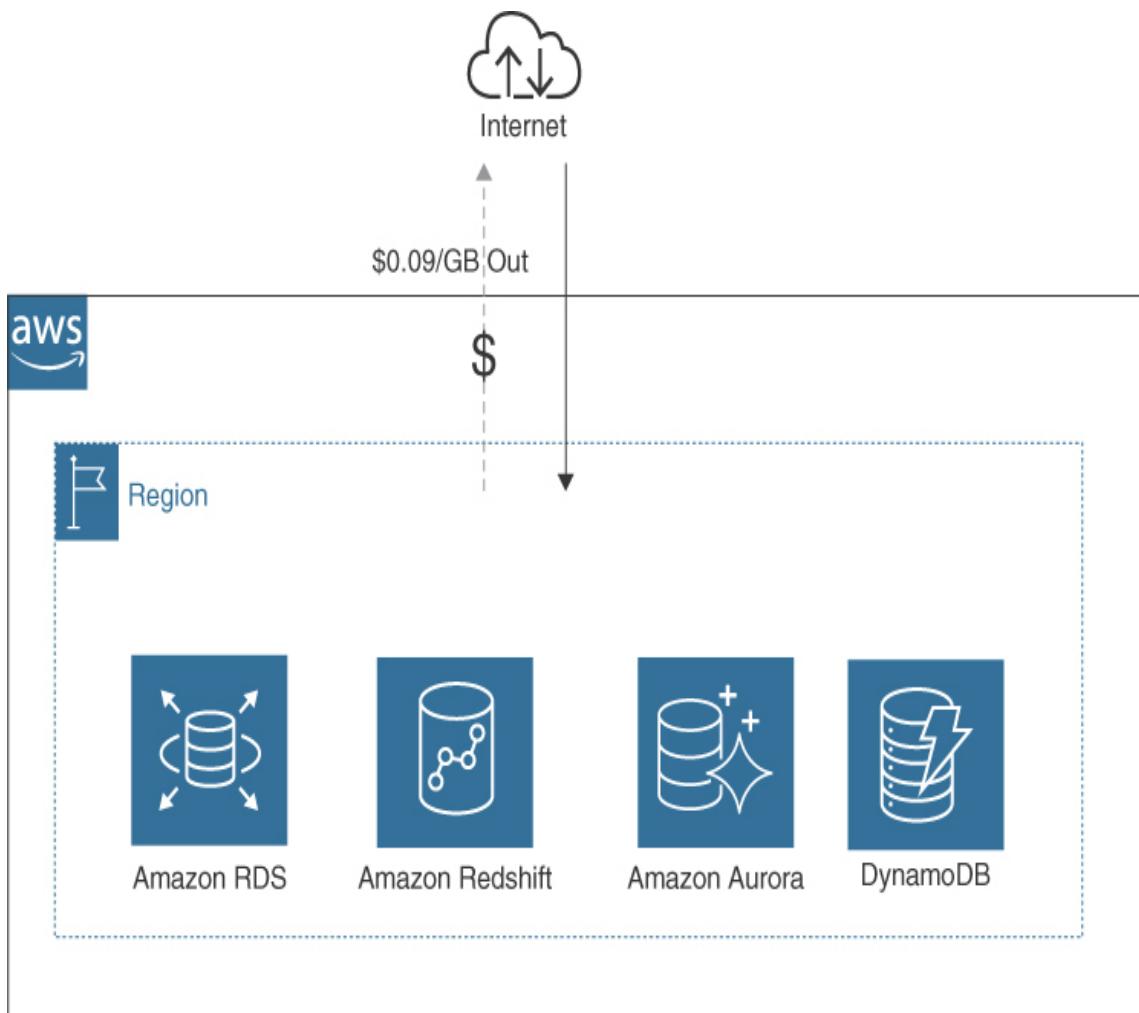


Figure 14-7 Data Transfer to the Internet

Data Transfer Costs and RDS

Key Topic

Workloads that use RDS deployments utilize EC2 instances and the EBS volumes. With multi-AZ deployment of primary and

secondary database instances, read replicas will not have any charges for data transfer to and from any EC2 and RDS instances located in the same AWS region, availability zone, and virtual private cloud. Data charges apply as follows for data transfers (see [Figure 14-8](#)) between instances:

- EC2 and RDS instances that are located across AZs within the same VPC are charged \$0.01 per GB ingress and egress.
- EC2 and RDS instances that are located across AZs and across different VPCs are charged \$0.01 per GB ingress and egress.
- EC2 and RDS instances that are located across AWS regions are charged on both sides of the data transfer from the EC2 instance to the RDS instance and vice versa at \$0.02 per GB ingress and egress.

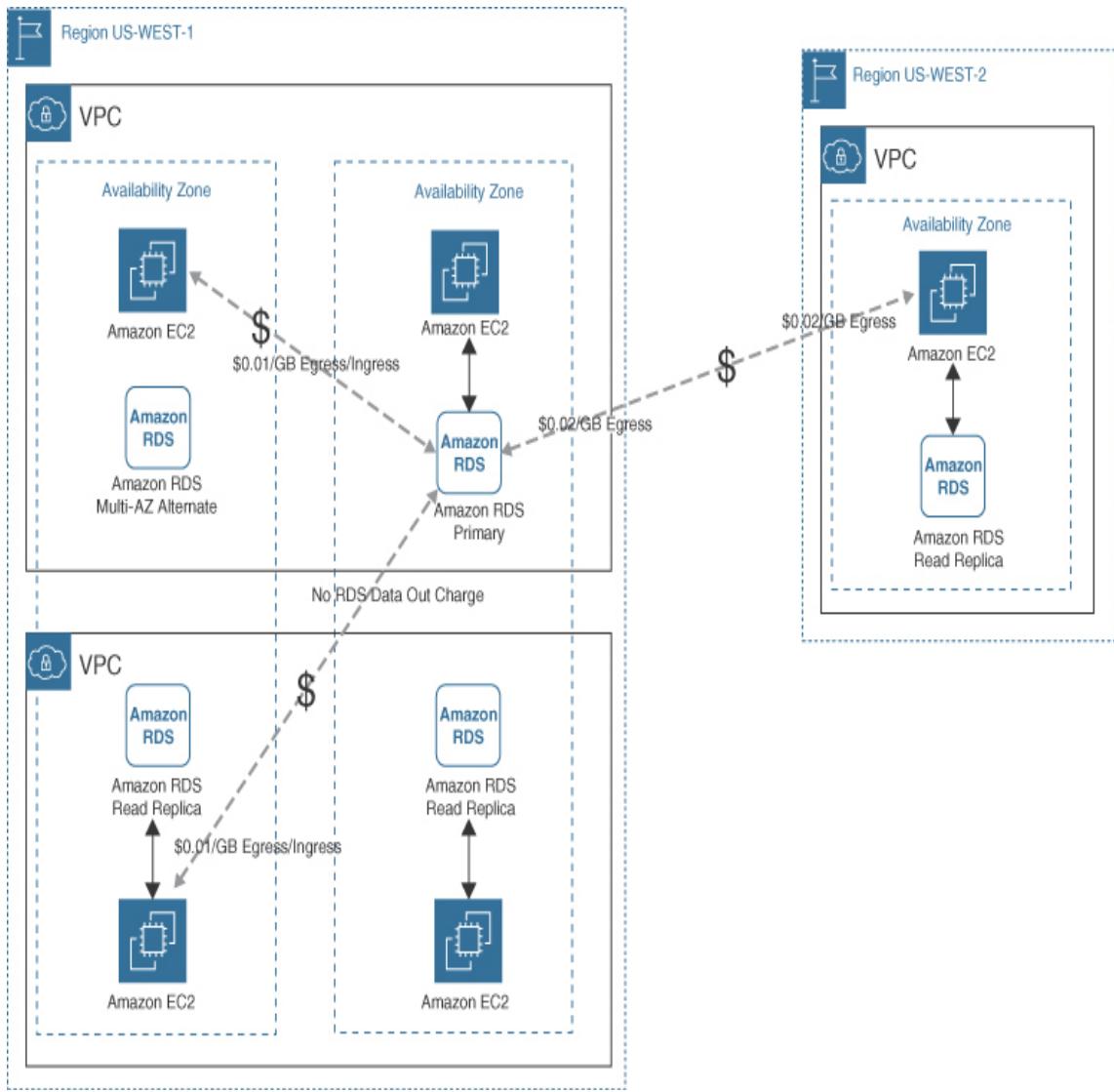


Figure 14-8 EC2 and RDS Data Transfer Across VPCs with Multiple AWS Regions

There are no data transfer charges for data replication in a multi-AZ deployment, or to any read replicas located within the same AWS region. There will also not be charges for the transfer of snapshots to the S3 bucket used for backup storage in the same AWS region. However, there will be data transfer

charges for asynchronous data updates to read replicas that are located across different regions at \$0.02 per GB egress. There will also be charges for any regional transfers for RDS snapshot copies or any automated cross-region backups at \$0.02 per GB egress.

Data Transfer Costs with DynamoDB

Workloads that use DynamoDB and DynamoDB Accelerator (DAX) will not have data transfer charges for:

- Inbound data transfers to DynamoDB
- Any data transfers between DynamoDB and EC2 instances located in the same region
- Any data transfers between EC2 instances and DAX in the same AZ (see [Figure 14-9](#))

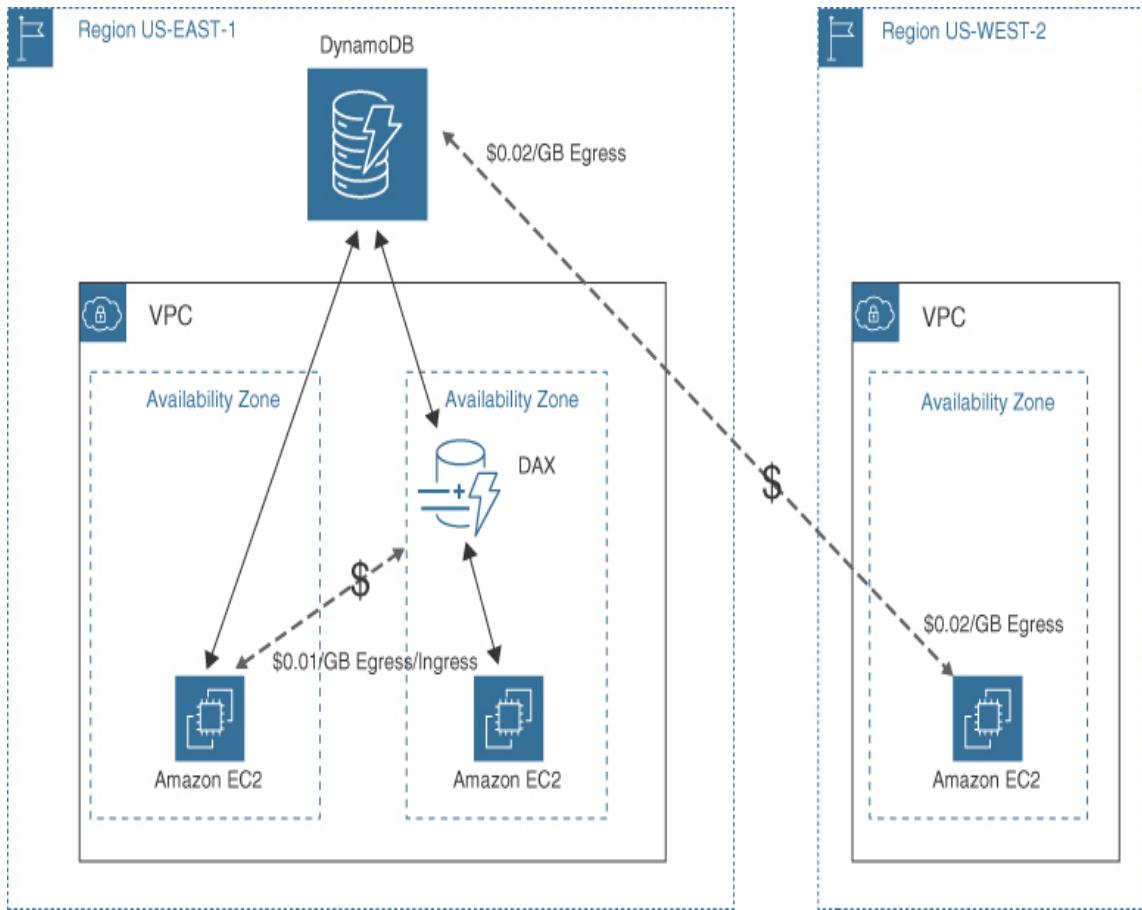


Figure 14-9 Amazon Aurora and Data Transfer Costs

For DynamoDB global table deployments, as shown in [Figure 14-10](#), the following data transfer charges will apply:

- Data transfer charges are charged between DynamoDB and a DAX deployment located in a different AZ.
- Global tables for cross-region replication charged at the source region rate of \$0.02 per GB egress.
- Any data transfers between DynamoDB and EC2 instances located in different AWS regions are charged on both sides of

the data transfer at \$0.02 per GB egress.

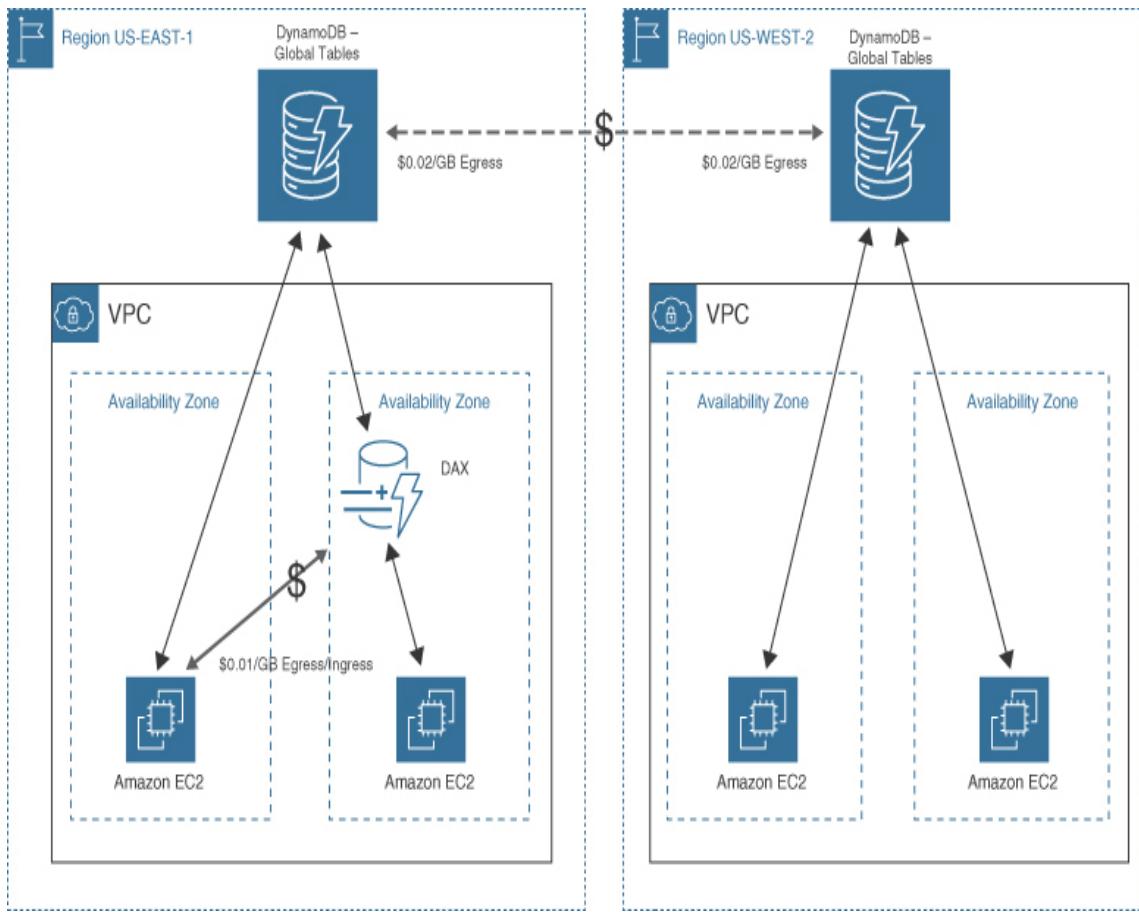


Figure 14-10 Amazon Aurora and Global Tables

Data Transfer Costs with Amazon Redshift

Workloads that use Amazon Redshift can analyze data stores using standard SQL queries and common business intelligence tools. For an ODBC application connecting to Redshift across multiple AWS regions, there are data transfer costs. For communication within the same availability zone and any data

transfers to S3 storage in the same AWS region for backup and restore, there are no data charges. For deployments utilizing multiple AWS regions, as shown in [Figure 14-11](#), the following data transfer charges will apply:

- EC2 and RDS instances that are located across AZs and across different VPCs are charged \$0.01 per GB ingress and egress.
- EC2 and RDS instances that are located across AWS regions are charged on both sides of the data transfer from the EC2 instance to the RDS instance and vice versa at \$0.02 per GB ingress and egress.

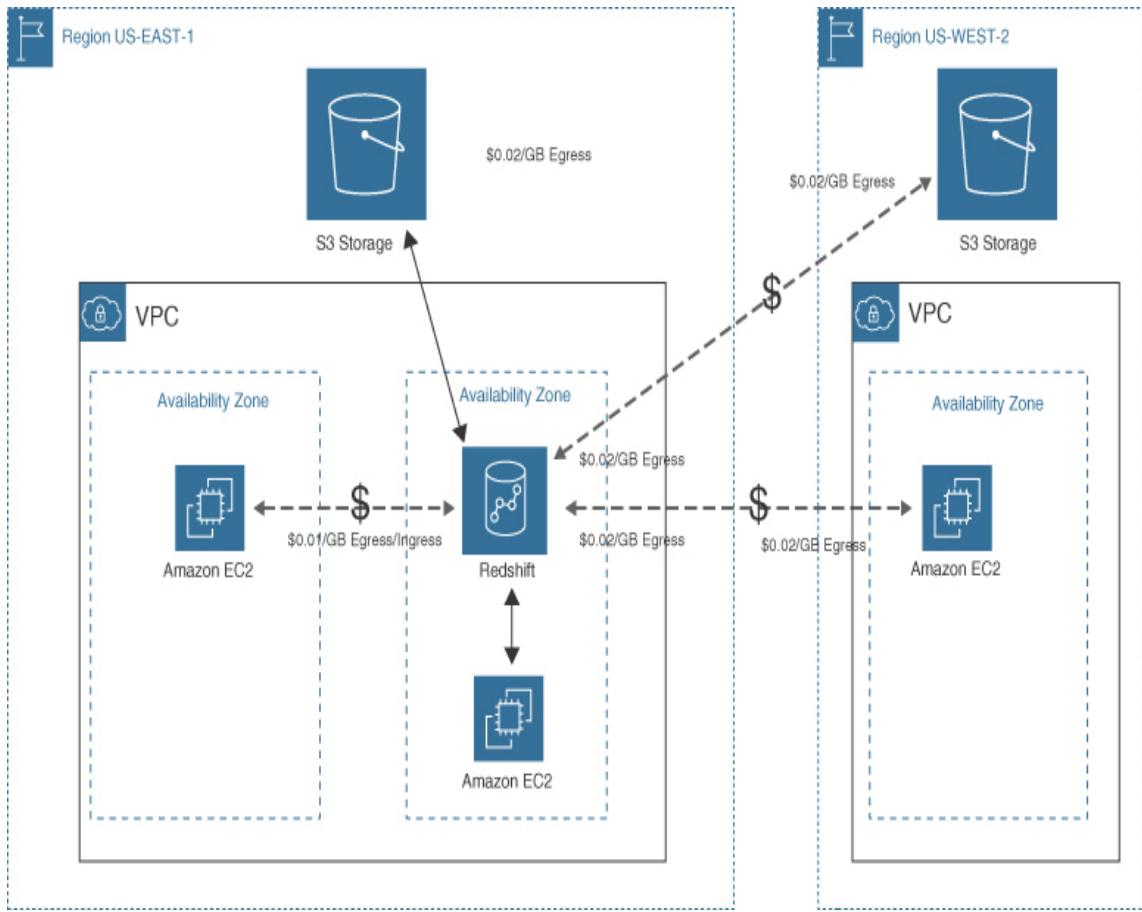


Figure 14-11 Amazon Redshift Data Transfer Costs

Data Transfer Costs with DocumentDB

Workloads that use DocumentDB are using a database service with MongoDB compatibility. An application using an EC2 instance using DocumentDB deployed as a global cluster as the data store across two AWS regions with cross-region replication will have data transfer charges (see [Figure 14-12](#)). However, read replicas in multiple AZs will have no data transfer charges for communication between any EC2 and DocumentDB instance

located in the same AZ, or for data transferred between DocumentDB instances within the same AWS region. There will also be data transfer charges for:

- EC2 instance and DynamoDB communication across availability zones
- Cross-region replication between the DocumentDB primary and secondary instances

Data Transfer Costs Cheat Sheet



When designing database deployments, consider the following options:

- Calculate data transfer charges on both sides of the communication channel. “data transfer in” to a destination is also “data transfer out” from the source.
- Use regional read replicas and alternate replicas to reduce the amount of cross-availability zone or cross-region traffic.

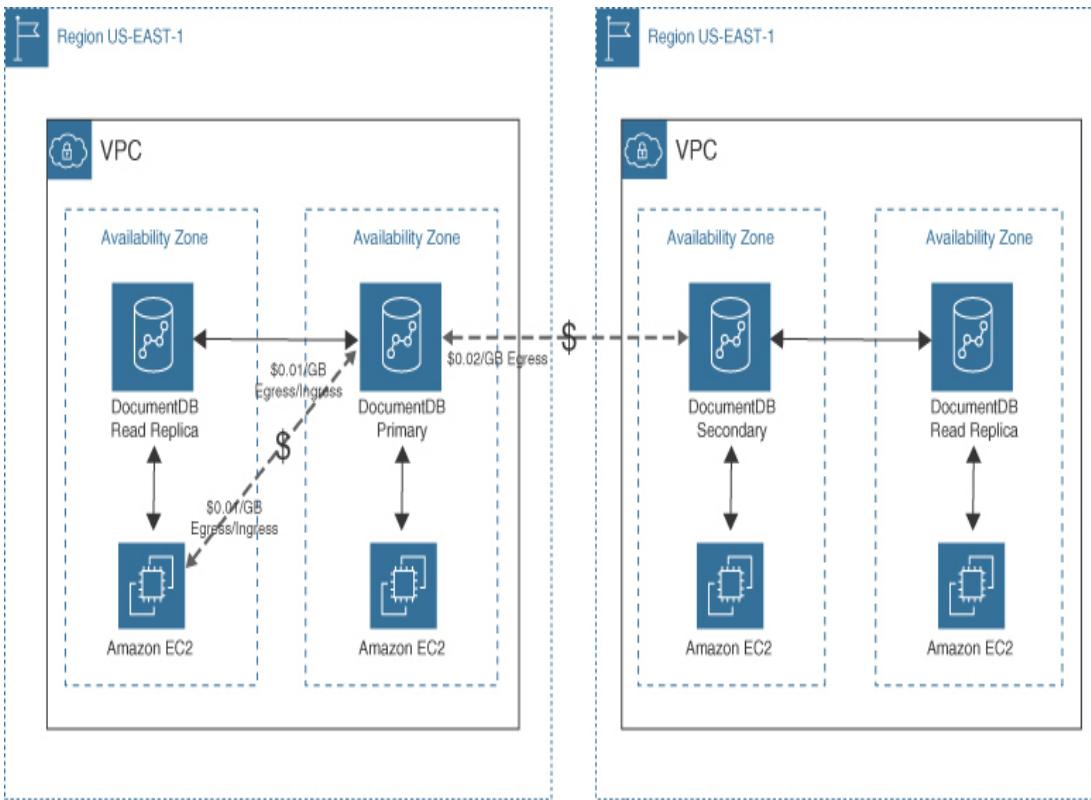


Figure 14-12 DocumentDB Data Transfer Charges

- Use data transfer tiered pricing when estimating workload pricing for data transferred out to the Internet across EC2 and RDS instances, Redshift, DynamoDB, and S3 buckets.
- Backup and snapshot requirements and how data transfer charges may apply.
- AWS offers various purpose-built, managed database offerings. Selecting the right database service for your workload can help optimize performance and cost.

- Review your application and how queries are designed, looking to reduce the amount of data transferred between your application and its data store.

Database Retention Policies

Database retention policies refer to the length of time that database backups are retained. During the defined backup window of each RDS deployment (see [Figure 14-13](#)), automatic backups of your DB instances are created and saved in controlled S3 storage. RDS creates a storage volume snapshot of the standby database; the automated backups of the RDS instance are saved according to the defined backup retention period currently specified. The automated backup contains a complete system backup, including a full database backup, transaction logs, and the DB instance properties. DynamoDB, Neptune, and DocumentDB also support continuous backup with point-in-time restoration.

Backup

Enable automated backups
Creates a point-in-time snapshot of your DB cluster

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Backup retention period Info
The number of days (1-35) for which automatic backups are kept.

7
▼
days

Backup window Info
Select the period for which you want automated backups of the DB cluster to be created by Amazon RDS.

Choose a window
 No preference

Start time

12
▼
:
00
▼
UTC

Duration

0.5
▼
hours

Figure 14-13 RDS Backup Windows

Database Backup Policies Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following about database backup policies:

- The first DB snapshot contains all the data for the complete DB instance.

- Subsequent snapshots of the same DB instance are incremental; only the data that has changed after the most recent snapshot will be saved.
- Databases can be recovered at any point in time during the backup retention period.
- RDS retains backup DB instances for a default retention period of 7 days.
- Retention periods can be set to up to 35 days.
- Point-in-time restores can specify any second during your retention period up to the latest restorable time.
- The preferred backup window is the period of time during which your DB instance is backed up.
- Amazon Aurora backs up your cluster volume automatically and retains the restored data for the length of the defined backup retention period.
- Amazon Document DB and Amazon Neptune back up your cluster volume continuously and retain the restored data for the length of the defined retention period.
- Amazon DynamoDB enables you to back up your table data continuously by using point-in-time recovery (PITR), backing up your table data automatically with per-second granularity allowing restores to any given second in the preceding 35 days.

- On-demand backups of DynamoDB can be performed using the DynamoDB service or AWS Backup.
- Amazon DocumentDB (with MongoDB compatibility) continuously backs up your data to S3 storage, allowing restoration to any point within the backup retention period of 1 to 35 days.
- Multi-AZ DB cluster deployments can be backed up with a Multi-AZ DB snapshot.
- RDS can replicate snapshots and transaction logs to another AWS region for Oracle version 12 and higher, PostgreSQL 9.6 and higher, and Microsoft SQL Server 2012 and higher.
- Manual snapshots can also be created at any time. Manual snapshots are not automatically deleted.
- Customers can have up to 100 manual snapshots per AWS region.
- Backups of RDS DB instances can be managed using AWS Backup. Resource tagging must be used to associate your DB instance with a backup plan.
- RDS snapshots can be exported to an S3 bucket from automated, manual, and AWS Backup snapshots.
- When a DB instance is deleted, a final DB snapshot can be created before deletion; the final DB snapshot can be used to restore the deleted DB instance later.

Exam Preparation Tasks

As mentioned in the section “[How to Use This Book](#)” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. [Table 14-5](#) lists these key topics and the page number on which each is found.



Table 14-5 [Chapter 14](#) Key Topics

Key Topic Element	Description	Page Number
-------------------	-------------	-------------

Table 14-2	AWS Database Service Comparison	669
----------------------------	---------------------------------	-----

Key Topic Element	Description	Page Number
Section	RDS Costs Cheat Sheet	671
Section	NoSQL Costs Cheat Sheet	676
<u>Table 14-3</u>	AWS NoSQL Database Service Comparisons	677
Section	Migrating Databases	680
Section	AWS Schema Conversion Tool	681
Section	Data Transfer Costs and RDS	682
Section	Data Transfer Costs Cheat Sheet	686
Section	Database Backup Policies Cheat Sheet	688

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

key-value

read capacity unit (RCU)

write capacity unit (WCU)

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep Software Online.

- 1.** On-demand capacity mode charges _____ for reading and writes requests.

- 2.** Provisioned capacity mode charges _____ for read and write capacity units.

- 3.** AWS SCT converts an existing _____ from one database engine to another.

- 4.** Workloads that use Amazon Redshift can analyze data stores using _____.
- 5.** Workloads that use DocumentDB are using a database service with _____.
- 6.** Database retention policies refer to the _____ that database backups are retained.
- 7.** Subsequent snapshots of the same DB instance are _____.
- 8.** Backups of RDS DB instances can be managed using _____.

Chapter 15

Designing Cost-Effective Network Architectures

This chapter covers the following topics:

- [Networking Services and Connectivity Costs](#)
- [Data Transfer Costs](#)

This chapter covers content that's important to the following exam domain and task statement:

Domain 4: Design Cost-Optimized Architectures

Task Statement 4: Design cost-optimized network solutions

Network services and all types of communication at AWS use the AWS private network and Internet connections to transfer vast quantities of data inbound (ingress) and outbound (egress). Egress packet flow is charged a *data transfer cost*. Data transfer costs may be zero (free of charge), minimal, or sometimes very expensive. In preparing for the AWS Certified Solutions Architect – Associate (SAA-C03) exam, students require a good understanding of data transfer costs.

Recall [Chapter 2](#), “[The Well-Architected Framework](#),” that one of the six pillars of the AWS Well-Architected Framework is Cost Optimization. As I mentioned in previous chapters, it’s a really good idea to download the AWS document “Cost Optimization Pillar” (see

<https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/wellarchitected-cost-optimization-pillar.pdf>) and read it thoroughly, which will help in understanding how to manage costs at AWS.

“Do I Know This Already?”

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 15-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A](#), “[Answers to the ‘Do I Know This Already?’ Quizzes and Q&A Sections.](#)”

Table 15-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
---------------------------	-----------

Networking Services and Connectivity Costs	1, 2
--	------

Data Transfer Costs	3, 4
---------------------	------

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment.

Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Data transfer costs are charged based on what type of network traffic?

1. Egress network data transfer
2. Ingress network data transfer

3. All inbound and outbound data flow

4. None of these

2. What is the charge for incoming data traffic to AWS services?

1. Incoming data is charged a tiered rate.

2. Incoming data is free.

3. Incoming data is charged a flat rate.

4. Incoming data is charged at \$0.01 per GiB.

3. How are Elastic Load Balancer (ELB) deployments charged?

1. GIB of data and LCU

2. By GIB of ingress data

3. Service charge and LCU

4. By GIB of egress data

4. How can NAT services be deployed as a highly available service?

1. A NAT gateway service per availability zone.

2. An EC2 instance NAT service per availability zone.

3. NAT can't be deployed as a HA service.

4. A NAT gateway service per AWS region.

Foundation Topics

Networking Services and Connectivity Costs

The typical network services used by workloads hosted at AWS include an Internet Gateway, a Virtual Private Gateway, Elastic Load Balancers (Application, Network, and Gateway), Amazon CloudFront, NAT services, and VPC networks and endpoint connections. Each service has a variety of operating costs, including an hourly charge for each service, and data transfer charges for data sent outbound.

Elastic Load Balancing Deployments

ELB charges are for each hour or partial hour that an ELB load balancer (NLB, ALB, or GWLB) is running, including the number of Load Balancer Capacity Units (LCUs) used per hour by each deployed load balancer. Each LCU offers

- 25 new connections per second
- 3,000 active connections per minute
- 1 GiB of processed bytes per second
- 1,000 rules evaluated per second

A Gateway Load Balancer allows organizations to centrally manage a target group of third-party load balancers distributing all incoming traffic to the virtual appliances (see [Figure 15-1](#)). Gateway Load Balancers use a virtual private

cloud (VPC) endpoint called a Gateway Load Balancer endpoint (GWLB endpoint) powered by AWS PrivateLink, allowing traffic across the GWLB endpoint. Each GWLB endpoint is priced per VPC attachment and per GiB of data processed through the endpoint. The supported protocol is GENEVE and the port is 6061.



Note

A **Load Balancer Capacity Unit (LCU)** measures the hourly characteristics and the capacity of network traffic processed by each deployed load balancer. You are charged based on the dimension with the highest hourly usage. The four dimensions are as follows:

- **New connections:** The number of newly established connections per second
- **Active connections:** The number of active connections per minute

- **Processed bytes:** The number of bytes processed by the load balancer in GiBps
 - **Rule evaluations:** The number of listener rules processed by the load balancer
-

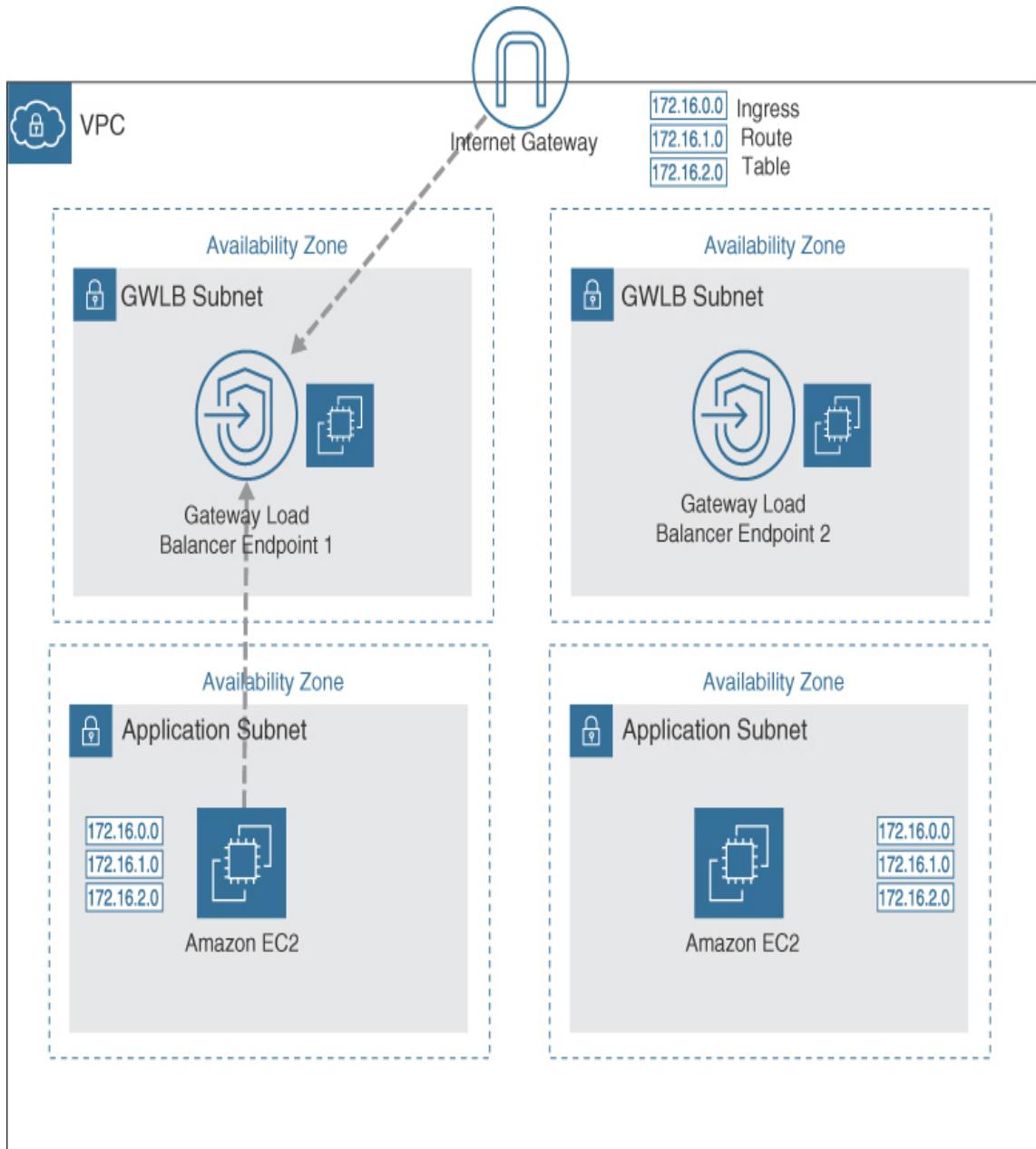


Figure 15-1 Gateway Load Balancer Deployment

NAT Devices

NAT devices relay packets between EC2 instances hosted on private subnets and Internet locations, returning responses back to the EC2 instance that sent the original request. There are hourly charges for each NAT gateway deployed and data processing charges for the GiBs of data transferred.

There are several use cases to consider when deploying NAT services at AWS:

- **NAT gateway instance use case:** An EC2 instance that has deployed a NAT AMI. Customers that choose this option must also manage updates and scale each NAT instance when more performance is required. The performance of the NAT instance will be determined by the EC2 instance type chosen. Network performance can be increased by choosing a different EC2 instance. Many EC2 instances have up to 5 GiBps bandwidth; for example, an m5n.xl instance has 50 GiBps of network bandwidth.
- **NAT gateway instance high availability:** For high-availability deployments, multiple NAT gateway instances can be deployed per availability zone (see [Figure 15-2](#)), but costs will be higher. The NAT gateway service can scale throughput up to 50 GiBps. Multiple NAT gateway service deployments per AZ provide high availability.

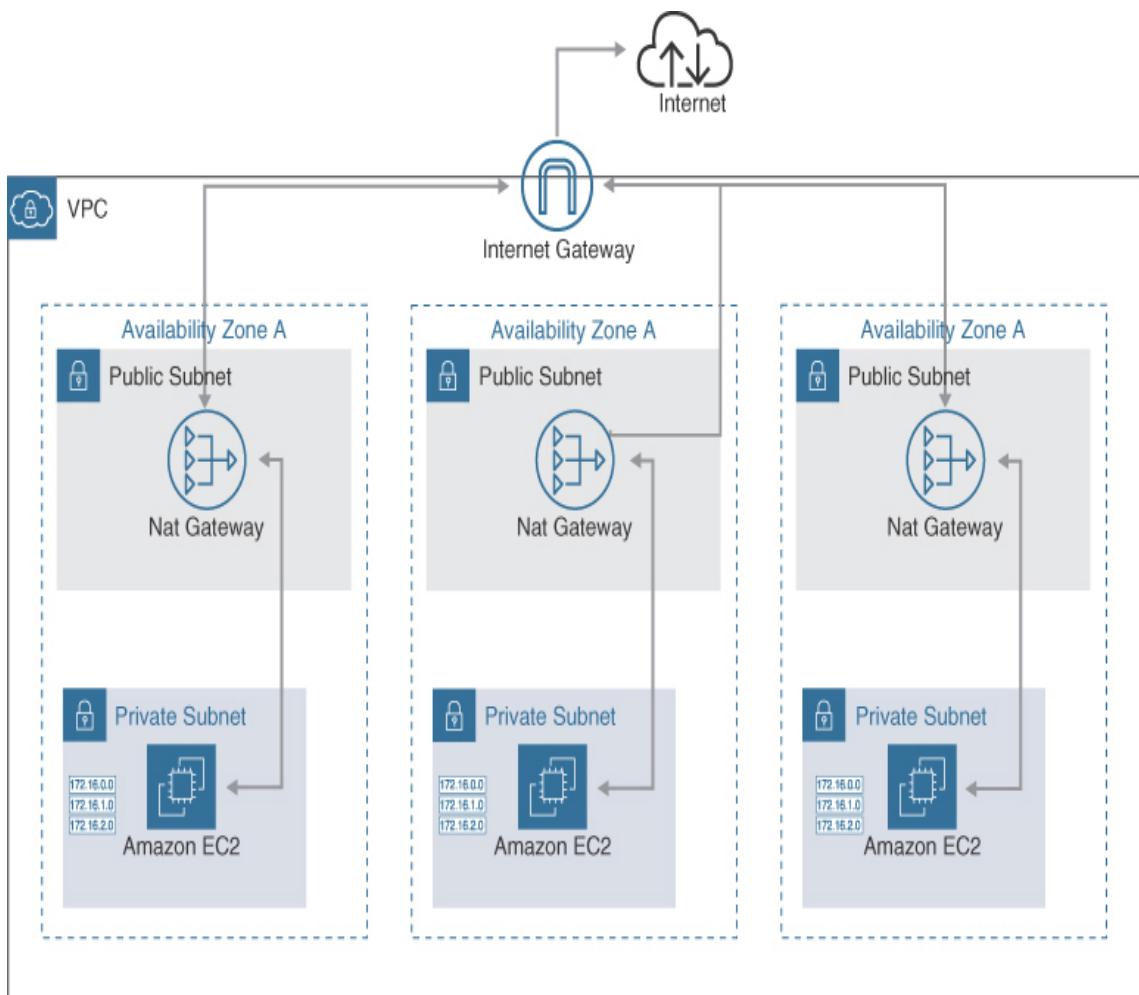


Figure 15-2 NAT Gateway Service HA Deployment

Costs can be reduced for NAT services by doing the following:

- Enable NAT gateway instances during defined maintenance windows for required updates.
- Create NAT gateways in the same AZ as the instances requiring Internet access to reduce cross-AZ data transfer charges.

- If most traffic through the NAT service is to AWS services that support VPC interface endpoints, create an Interface endpoint for each service.
- If the majority of NAT service charges are to Amazon S3 or Amazon DynamoDB, set up gateway VPC endpoints. There are no charges for using a gateway VPC endpoint.

AWS CloudFront

Amazon CloudFront delivers web and media content stored in S3 buckets to clients worldwide using one of the hundreds of edge locations. If the requested content is already cached at the edge location, it is delivered to the viewer (end user) quickly. Delivery costs are billed per GiB transferred from an edge location server to the viewer; customers are charged per 10,000 HTTP requests. The billing rate for serving data ranges from \$0.085 per GiB to \$0.170 per GiB and is determined by where the viewer request originates from. Any data transferred out to an edge location from an EC2 instance, S3 bucket, or an Elastic Load Balancer has no additional data transfer charges from each AWS service, just CloudFront charges (see [Figure 15-3](#)). Customers can save up to 30% in delivery costs and 10% off AWS WAF service charges by subscribing to a CloudFront Security Savings Bundle. Amazon CloudFront costs increase (see [Table 15-2](#)) as additional features are enabled:

- **Encryption:** Although there is a charge for encryption, less data will be sent; therefore, data transfer costs will be reduced.
- **Logging:** Enabling real-time logs costs \$0.01 per million log lines written.
- **CloudFront Origin Shield:** Improves the cache hit ratio by using CloudFront regional edge caches, which are hosted across three AZs. Enabling Origin Shield is charged per 10,000 requests.
- **CloudFront functions and Lambda@Edge functions:** Charged per request and duration.
- **Custom SSL/TLS certificates and domain names:** Charged monthly.

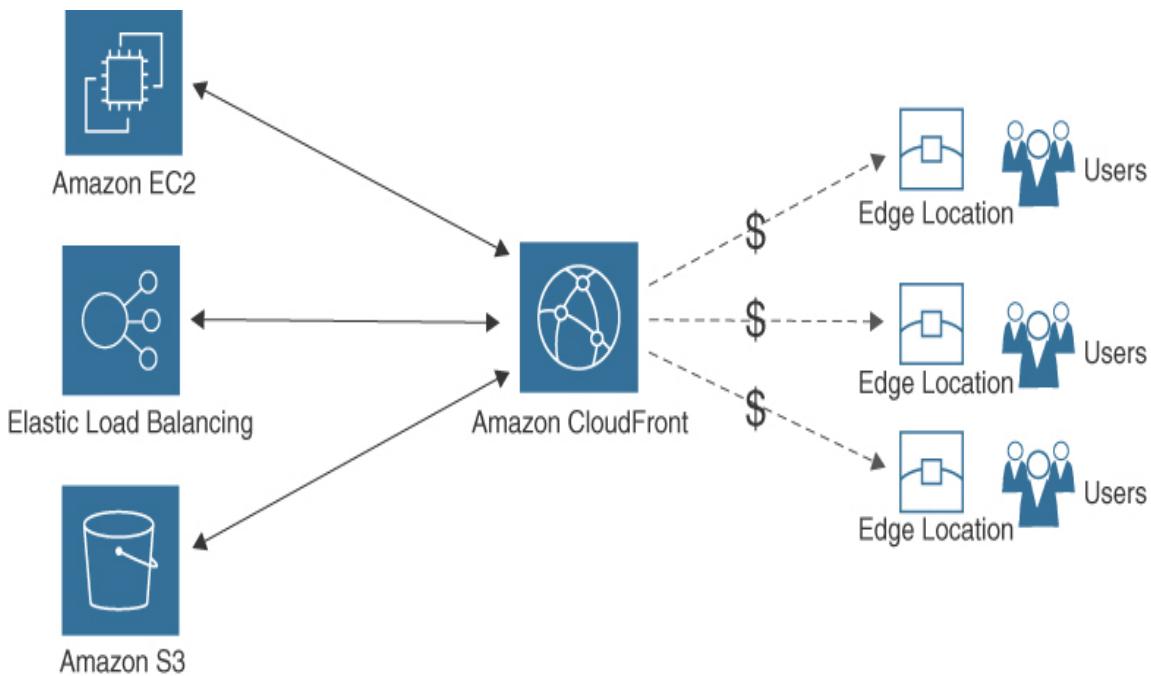


Figure 15-3 Data Transfer Charges Minimized



Table 15-2 Cost Comparison for CloudFront Costs

Data Type	Pricing	Details
CloudFront to the Internet/viewer	\$0.085–\$0.120 per GiB (first 10 TB)	Data transferred from edge location to viewer location
CloudFront to data or server origin	\$0.020–\$0.160 per GiB	Data requests to origin (POST and PUT)

Data Type	Pricing	Details
HTTP/HTTPS requests	\$0.0075–\$0.0120 per 10,000 requests	Charges for HTTP and HTTPS requests
Origin shield requests	\$0.0075–\$0.0090 per 10,000 requests	Requests to origin shield cache layer
File invalidation requests	\$0–\$0.005 per path requested	Remove files from edge location before TTL expires

Data Type	Pricing	Details
Lambda Functions requests	\$0.60 per million requests/\$0.00005001 per GiBps execution time	Charged per request and execution time
Field-Level Encryption requests	\$0.02 per 10,000 requests	Encrypt specific fields in HTTP form
Real-time log requests	\$0.01–\$0.01 for every 1 million log lines written	Log requests to distribution

Data Type	Pricing	Details
Custom SSL Certificate	\$600 per month per certificate	Used when content is delivered to browsers that don't support Server Name Indication (SNI)

CloudFront Pricing Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following important cost considerations before deploying CloudFront:

- S3 transfers under 1 GiB are free per month; however, Amazon CloudFront delivery could be faster depending on

the location of the end user.

- Transfers of data over 50 GiB per month from Amazon S3 or EC2 instances will be cheaper using an Amazon CloudFront distribution.
- If applications exclusively serve **GET** requests, direct requests to the S3 bucket are cheaper.
- Applications using both **GET** and **POST** requests will be cheaper to access using an Amazon CloudFront distribution.
- HTTP requests for data are cheaper than HTTPS requests.
- By default, all files cached at an Amazon CloudFront edge location expire after 24 hours.
- Change the minimum, maximum, and default **time to live (TTL)** values on all cached objects in the distribution to extend the cache storage time. Each object in the CloudFront cache is identified by a unique cache key. When a viewer requests an object that is stored in the edge location cache, this is defined as a cache “hit,” which reduces the load on the origin server and reduces the latency of the object delivery to the viewer. To improve the cache hit ratio, include only the minimum values in the cache key (see [**Figure 15-4**](#)) for each object. The default cache key includes the domain name of the CloudFront distribution and the URL path of the requested object. Other cache values, HTTP headers, and cookies can be defined with a cache policy.

Create cache policy

Details

Name
Enter a name for the cache policy.

Description - optional
Enter a description for the cache policy.

TTL settings Info

Minimum TTL	Maximum TTL	Default TTL
Minimum time to live in seconds. <input type="text" value="1"/>	Maximum time to live in seconds. <input type="text" value="31536000"/>	Default time to live in seconds. <input type="text" value="86400"/>

Cache key settings Info

Headers
Choose which headers to include in the cache key.

Query strings
Choose which query strings to include in the cache key.

Cookies
Choose which cookies to include in the cache key.

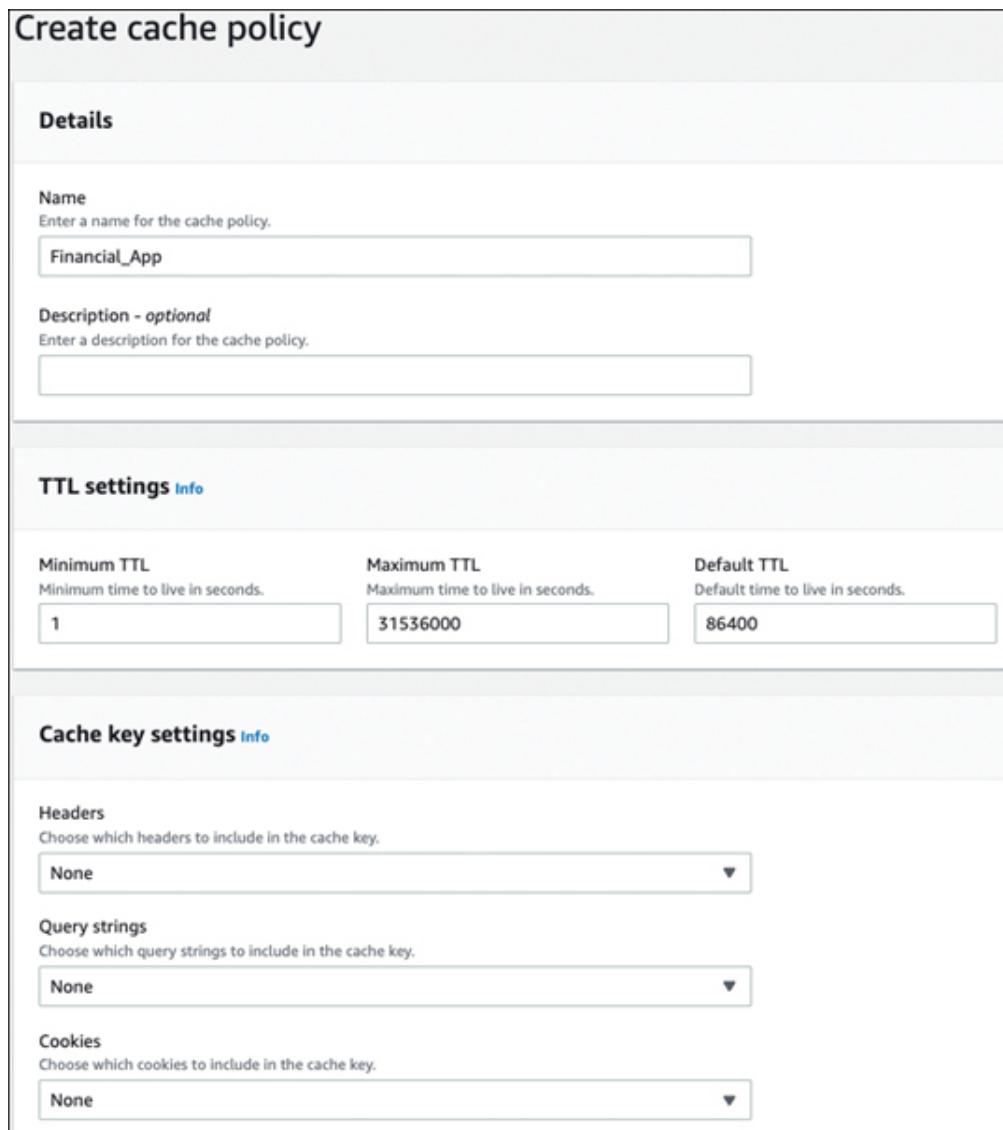


Figure 15-4 Cache Key Settings

- For cache content that will rarely or never change, setting the Cache-Control HTTP headers on the origin server will set the cache rate at the client's browser and at the edge location.
- Enabling compression will reduce data transfer costs.

- Reserve capacity a year in advance.
- Opt out of more expensive regions/edge locations to reduce data transfer costs.

VPC Endpoints

Endpoint services allow access to most AWS services across the private AWS network, providing security and speed:

- **AWS PrivateLink:** AWS PrivateLink provides private connectivity between VPCs and on-premises locations to third-party services hosted at AWS. PrivateLink endpoints can also be accessed over VPC peering, VPN, and AWS Direct Connect connections.
- **VPC endpoints:** VPC interface endpoints use elastic network interfaces (ENIs) provisioned from the selected subnet in your VPC to a supported AWS service such as Amazon Elastic Container Registry (ECR) (see [Figure 15-5](#)). Communicating from a VPC directly to an AWS service across the AWS private network does not require an Internet gateway, NAT gateway services, or AWS VPN connections, thereby saving costs. VPC gateway endpoints route traffic to Amazon DynamoDB and Amazon S3 buckets. There are no processing charges when using a VPC gateway endpoint.

Key Topic

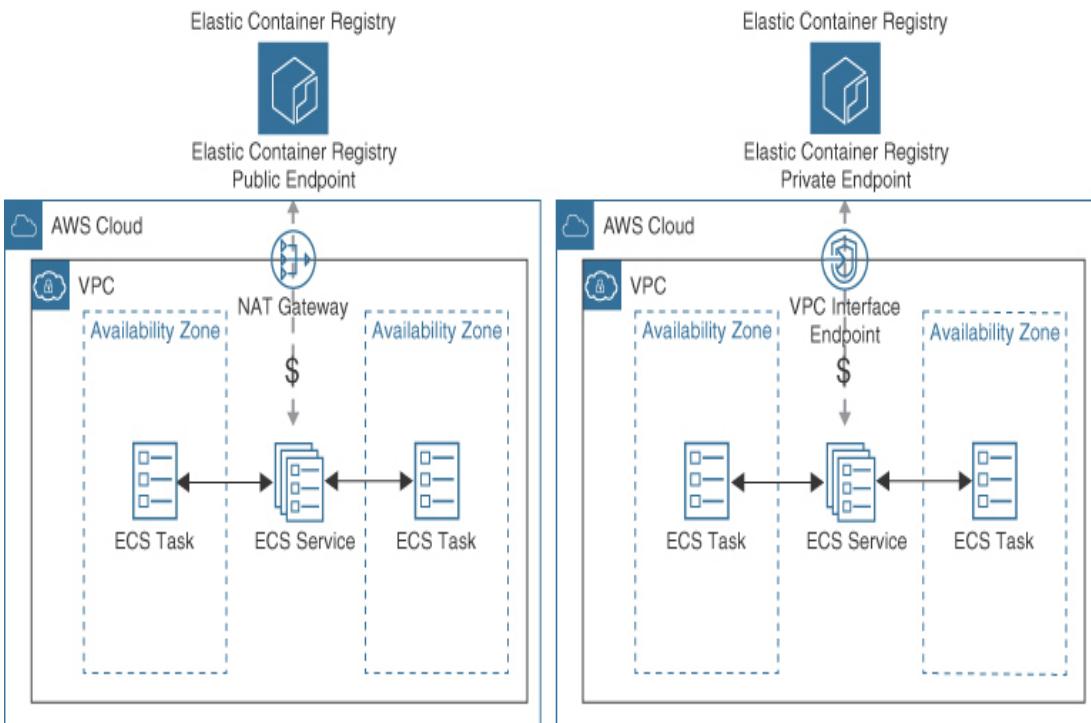


Figure 15-5 Accessing Elastic Container Registry Using an Interface Endpoint

For example, using the NAT Gateway Service to pull down images from the Amazon Elastic Container Registry is five times more expensive than using a VPC interface endpoint. VPC interface endpoints are used by AWS resources within the VPC and from on-premises locations using an AWS Direct Connect or VPN connection. VPC interface endpoint pricing is \$0.01 per connection hour and \$0.01 per GiB processed. [Table](#)

[15-3](#) compares the data transfer costs for the NAT gateway and VPC endpoints processing 100 TB of data in a 500-hour timeframe.

Table 15-3 NAT Gateway and VPC Endpoint Charges Comparison (100 TB/500 Hours)

Processing Costs by AWS Service	NAT Gateway Service	VPC Interface	VPC Endpoint
NAT gateway charge (\$0.045) per hour	\$45.00	\$45.00	\$45.00
NAT gateway processing (\$0.045) GiB	\$4,626.00	—	—
Gateway endpoint charge (\$0.00) per hour	—	—	\$0.00

Processing Costs by AWS Service	NAT Gateway Service	VPC Interface Endpoint	VPC Gateway Endpoint
Gateway endpoint processing (\$0.045) GiB	—	—	\$0.00
Interface endpoint charge (\$0.00) per hour	—	\$10.00	—
Interface endpoint processing (\$0.045) GiB	—	\$1,028.00	—
		\$4,671.00	\$1,083.00
			\$45.00

- **VPC peering:** Point-to-point connectivity provides full bidirectional direct connectivity between two VPCs. VPC

peering costs are charged only when network traffic crosses the peering connection. Best practice is to peer fewer than ten VPCs together. VPC peering has the lowest cost when compared to an AWS Transit Gateway deployment and peering has no hourly infrastructure cost. VPC peering costs are discussed in the next section, “[Data Transfer Costs](#).”

- **AWS Transit Gateway:** Hub and spoke designs can connect thousands of VPCs within the same AWS region and on-premises networks. Both VPN and AWS Direct Connect and Direct Connect gateways can be attached to a single AWS Transit Gateway deployment. AWS Transit Gateway peering allows peering Transit AWS Transit Gateway deployments within or across multiple AWS regions.

Note

Use VPC peering and/or AWS Transit Gateway for Layer 3 IP connectivity between VPCs.

- **VPC sharing:** The owner of a VPC can share a subnet and the resource hosted on the subnet to be shared, such as a database, with other participant AWS accounts. VPC sharing does not require VPC peering. There are no data transfer charges when sharing subnet resources between AWS

accounts within the same availability zone. VPC sharing is enabled using AWS Resource Access Manager.

Network Services from On-Premises Locations



Workloads hosted at AWS requiring access to on-premises data centers will incur data transfer charges when connecting using an AWS Site-to-Site VPN connection or an AWS Direct Connect connection:

- **Data transferred using an AWS Site-to-Site VPN connection:** Each AWS Site-to-Site VPN deployed will include an hourly charge for each connection and charges for data transferred from AWS across the connection (see [Figure 15-6](#)).
- **Data transferred using an AWS Direct Connect connection:** Direct Connect provides a high-speed single-mode fiber connection for connecting on-premises networks to AWS. Direct Connect connections are charged a fee for each hour the connection port is utilized and a data transfer charge for data flowing out of AWS (see [Figure 15-7](#)). All data flowing into AWS is free (\$0.00). Data transfer charges will

depend on the source AWS region and the third-party AWS Direct Connect provider location. AWS Direct Connect can also connect to an AWS Transit Gateway instance using an AWS Direct Connect gateway (see [Figure 15-8](#)), allowing multiple VPCs to be connected together.

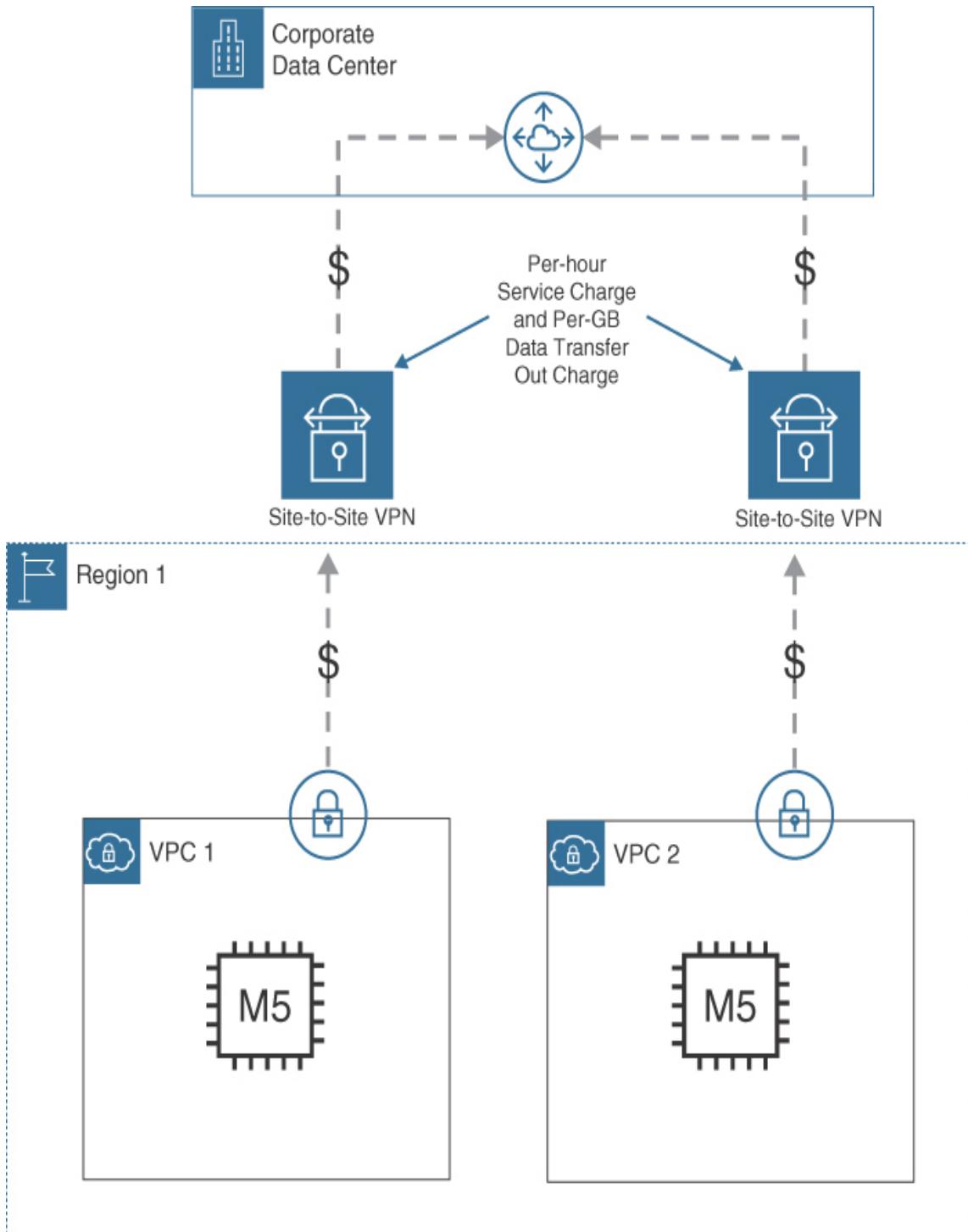


Figure 15-6 Traffic Charges for AWS Site-to-Site VPN Connections

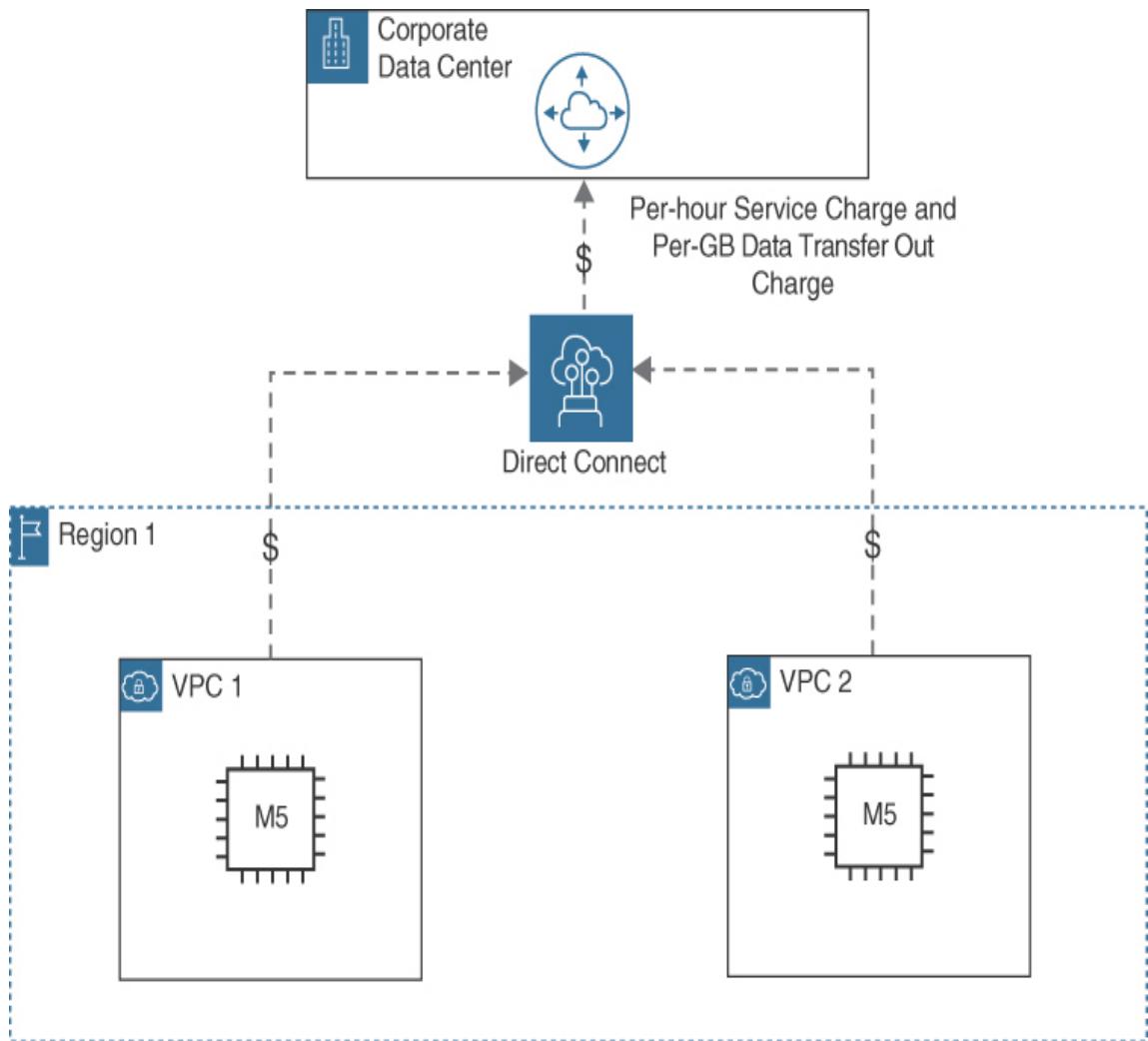


Figure 15-7 Traffic Charges for AWS Direct Connect Connections

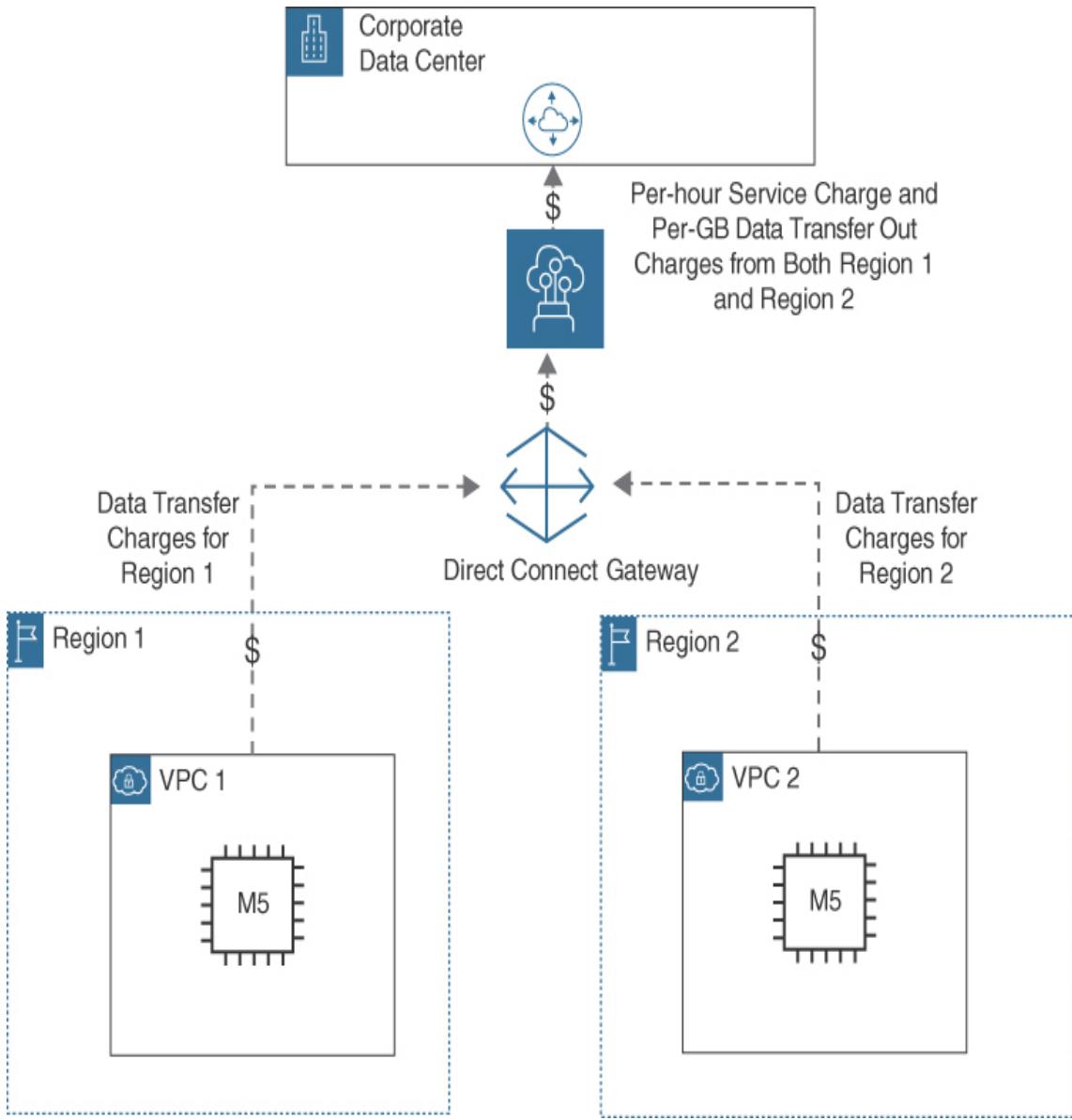


Figure 15-8 Traffic Charges for AWS Direct Connect Gateway Connections

Data Transfer Costs

To reduce networking costs, you need to look at network traffic flows—both egress and ingress. Data transfer costs are charges

for egress (outgoing) network traffic across the AWS cloud, and when exiting the AWS cloud. AWS charges you for outbound network traffic to the Internet, across availability zones or regions, or across a peered network connection. Regional data transfer costs that include networking are NAT gateway services, VPN connections, and ELB deployments. There are no data transfer costs within a single AZ. There are data transfer costs when a workload spans multiple AZs.

Your first monthly AWS bill will contain a few data transfer charge surprises. Data transfer costs are generally higher for data transfer between AWS regions than for intra-region data transfer between AZs within the same region.

Note

It's a good idea to subscribe to the Amazon pricing notification service to receive alerts when prices for AWS services change (which happens all the time). See

<https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/price-notification.html>.

Sometimes, a link to a cost calculator is present when you order AWS services, but data transfer costs will need to be calculated.

Data transfer costs can be expanded into the following breakdowns:

- Data transfer costs across AZs within a region. There are no data transfer costs for data transfer within a single AZ
- Data transfer costs between AWS regions
- Data transfer costs by service for egress data sent outbound

When you transfer data into an AWS region from any service from any other AWS region, it's free. As a rule, incoming data transfers are free for all public cloud providers. Regional data transfer costs are charged based on data transfer within a region (intra-region) or data transfer across regions (inter-region). Within each region, charges depend on whether you are communicating within an AZ or across AZs, as shown in [Figure 15-9.](#)

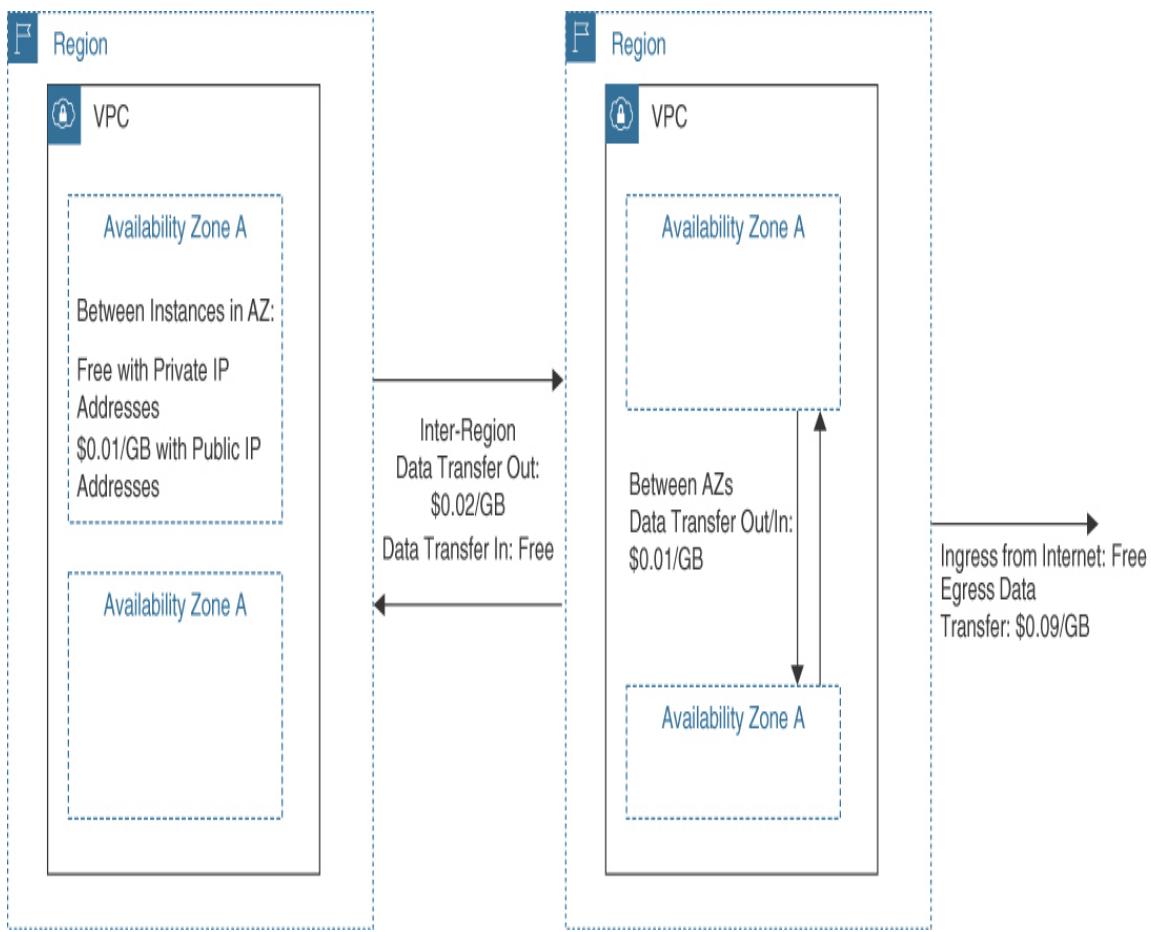


Figure 15-9 AWS Data Transfer Costs Comparison

Accessing AWS Services in the Same Region

Key Topic

Data transfer into any AWS region from the Internet is free of charge. Data transfer out to the Internet from an AWS region is billed at a region-specific tiered data transfer rate. Current EC2

on-demand pricing rates can be found at <https://aws.amazon.com/ec2/pricing/on-demand/#Data Transfer>. Data transfer from a source AWS region to another AWS region is charged at a source region-specific data transfer rate. Monthly AWS bills refer to these costs as inter-region inbound and inter-region outbound costs. If the Internet gateway is used to access the public endpoint of the AWS service in this same region, there are no data transfer charges (see [Figure 15-10](#)).

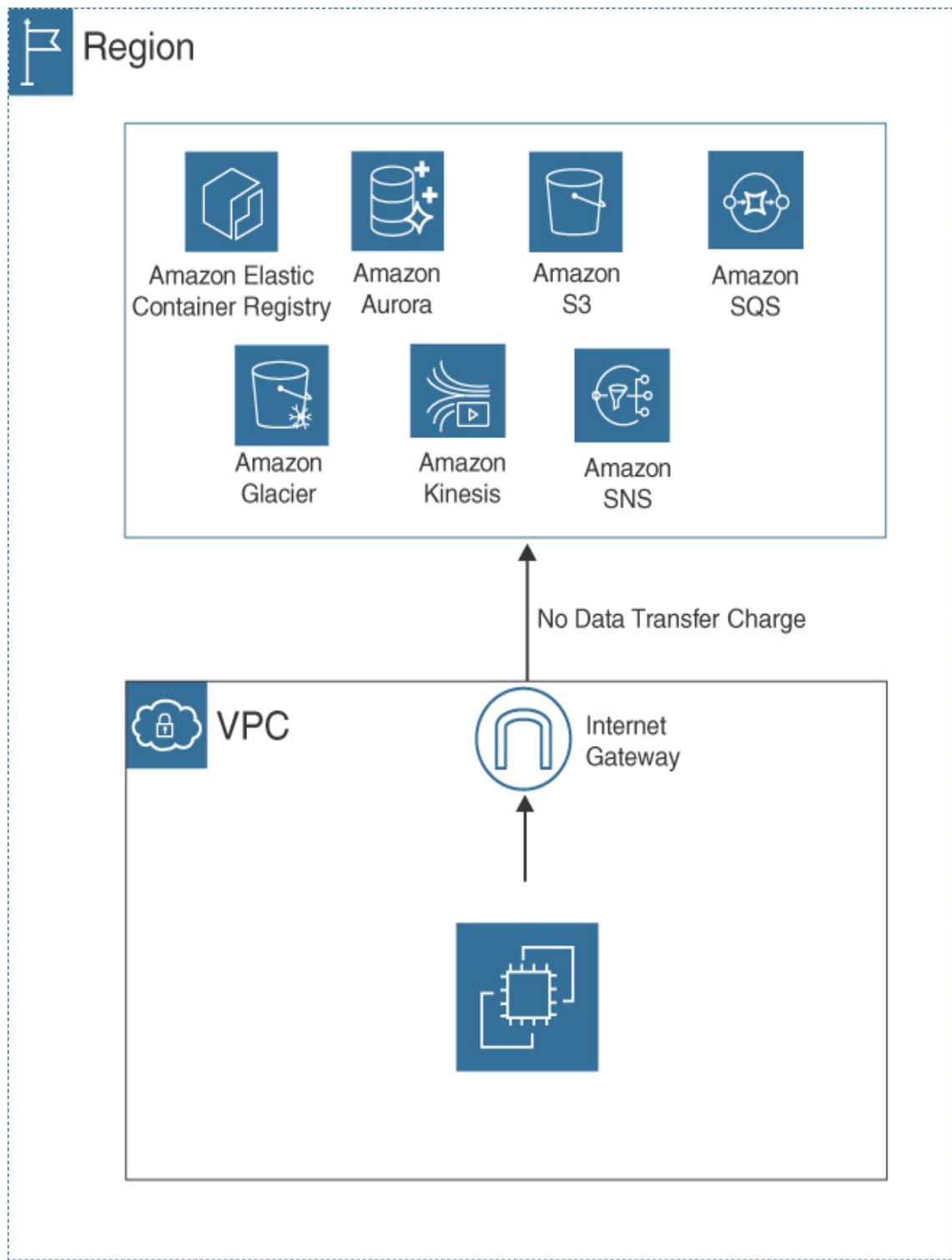
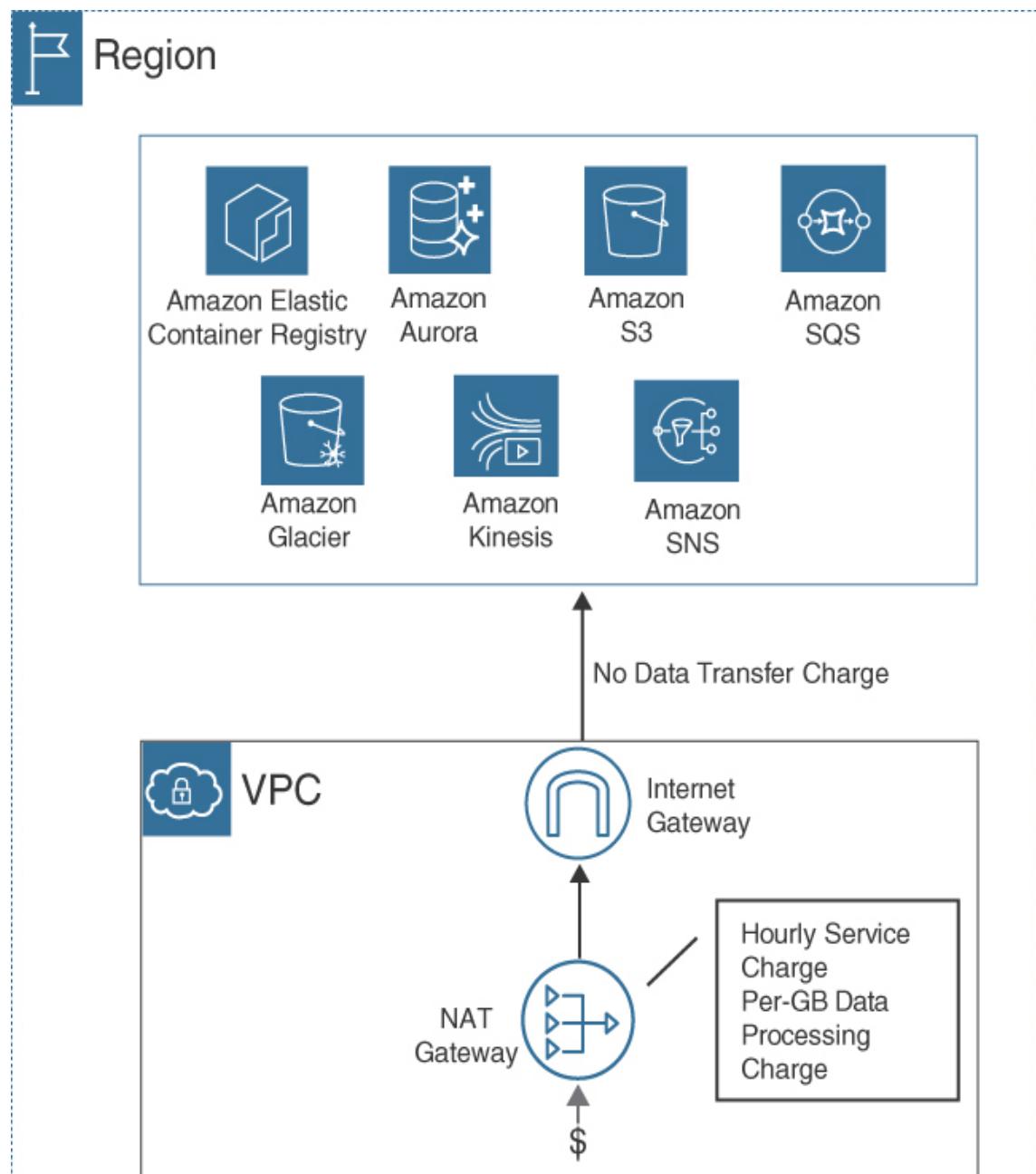


Figure 15-10 Accessing Services Using an Internet Gateway

If a NAT gateway is used to access the same AWS services from a private subnet, there will be a data processing charge per GiB for any data that passes through the NAT gateway (see [Figure 15-11](#)).



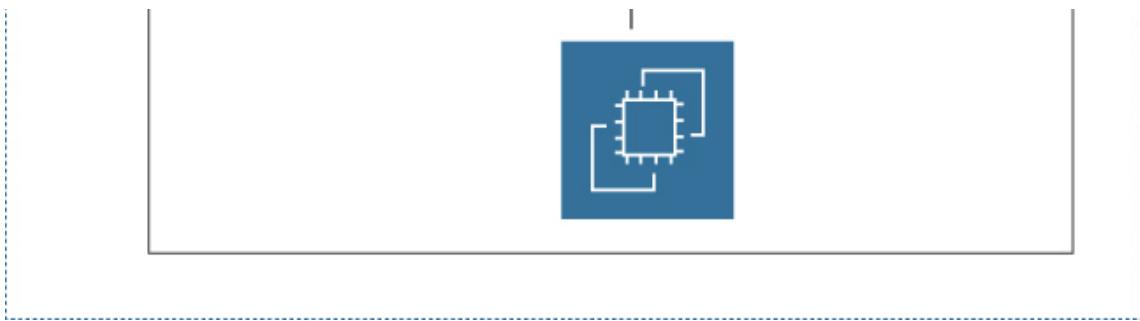


Figure 15-11 Accessing Services Using a NAT Gateway

Workload Components in the Same Region

Data transfer between EC2 instances or containers or with ENIs in the same AZ and VPC using private IPv4 or IPv6 addresses is free. Data transfer between EC2 instances or containers and Amazon S3 storage in the same AZ from the same VPC is also free. For a custom workload design that uses multiple AZs, there will be service-specific pricing for data transfers for cross-AZ communication between the EC2 instances; however, for Amazon RDS designs deployed across multiple AZs and Amazon Aurora, replication of data records across multiple AZs is free. Data transfer charges will be charged for all ingress traffic on both sides of a peering connection that crosses AZs (see [Figure 15-12](#)).

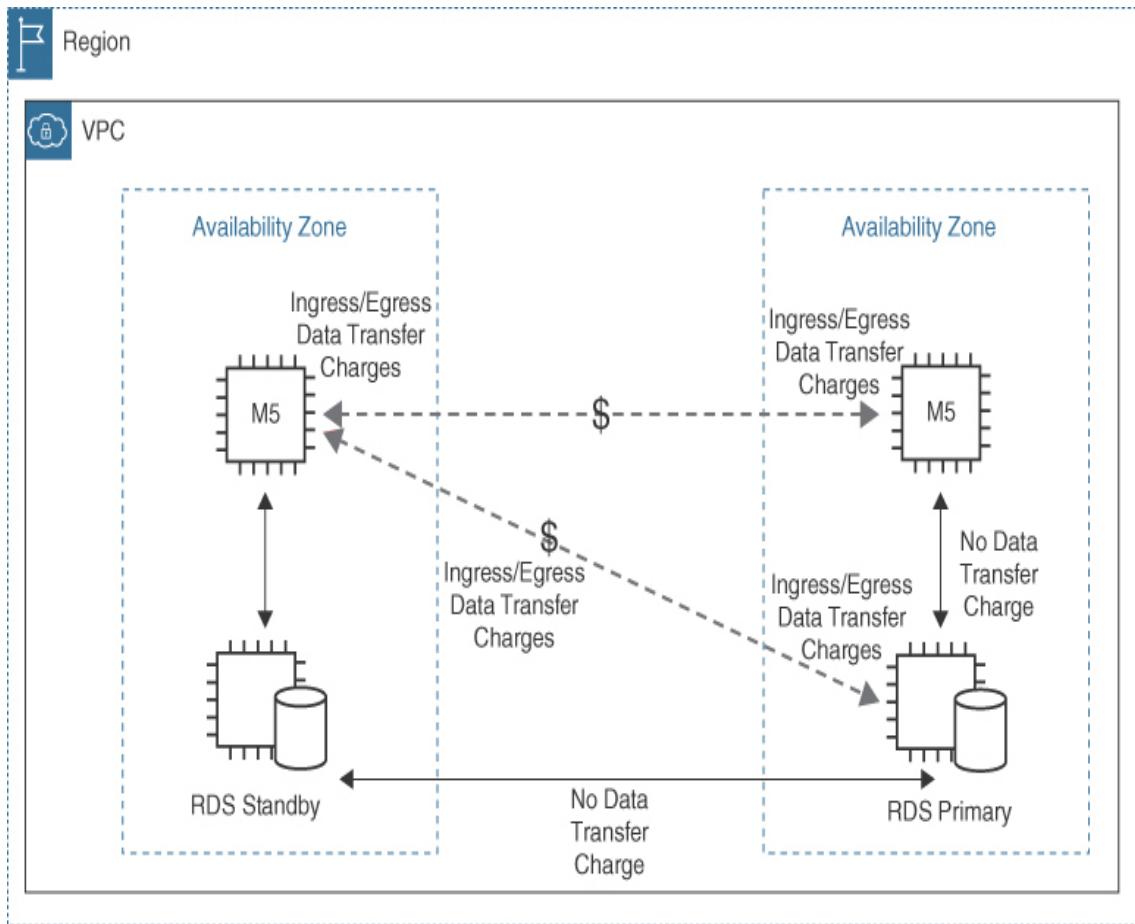


Figure 15-12 Workload Communications Across Availability Zones

A common workload design pattern is to utilize multiple VPCs in the same AWS region. Two methods of VPC-to-VPC communication are VPC peering connections or AWS Transit Gateway. Any data transfer over a VPC peering connection that stays within an AZ is free (see [Figure 15-13](#)). An AWS Transit Gateway can interconnect thousands of VPCs together. Transit Gateway costs include an hourly charge for each attached VPC, AWS Direct Connect connection VPN connection, and data

processing charges for each GiB of data to the Transit Gateway (see [Figure 15-14](#)).

Accessing AWS Services in Different Regions

Key Topic

Workloads that use services in different AWS regions will be charged data transfer fees. Charges will depend on the source and destination regions. For communication across multiple AWS regions using VPC peering connections or Transit Gateway connections, additional data transfer charges will apply. Inter-region data transfer charges will also apply for VPCs peered across regions (see [Figure 15-15](#)).

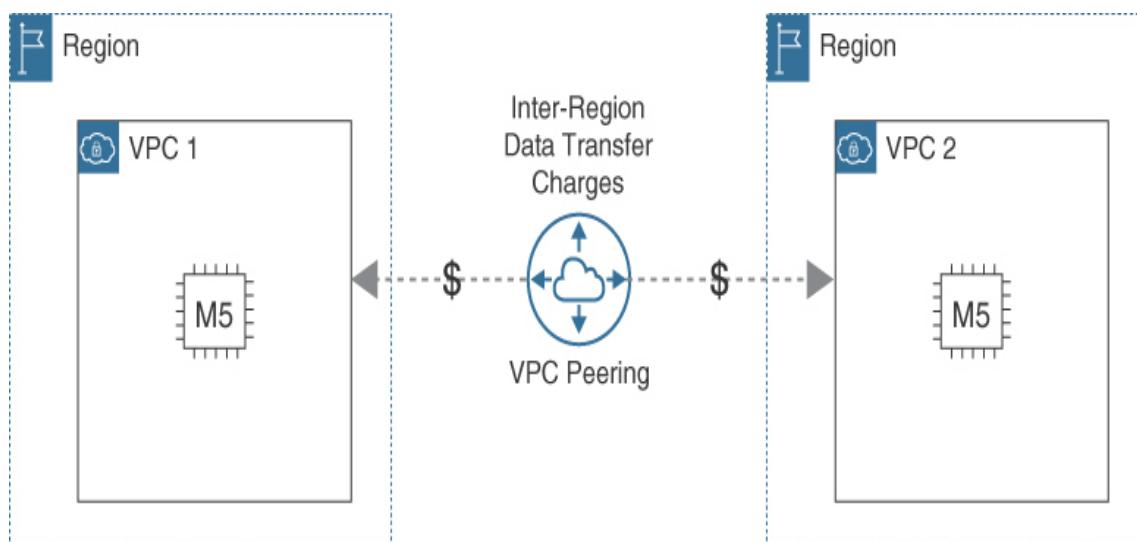


Figure 15-13 VPC Peering Connections and Charges

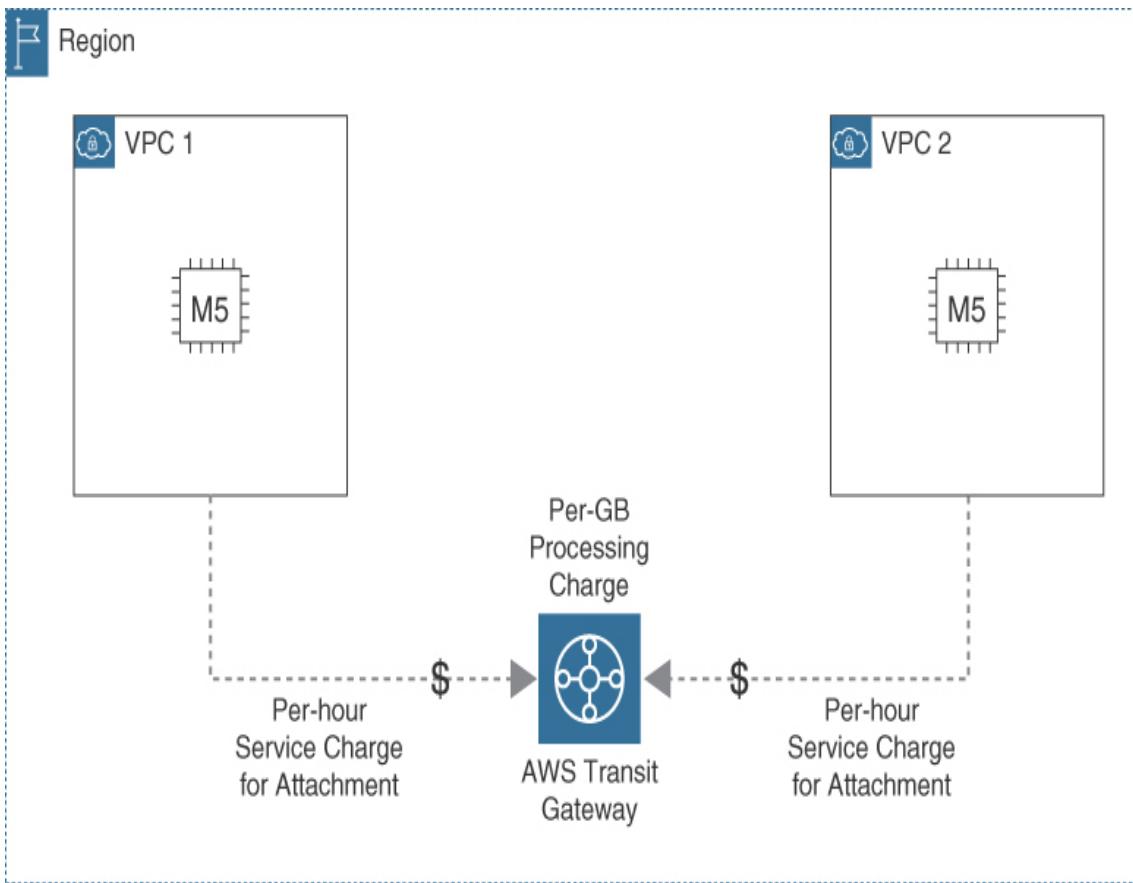


Figure 15-14 Transit Gateway Charges in the Same AWS Region

For Transit Gateway deployments that are peered together (see [Figure 15-16](#)), data transfer charges will be charged on one side of the peered connection; for example, data transfer charges do not apply for data sent from the EC2 instance to the Transit Gateway.

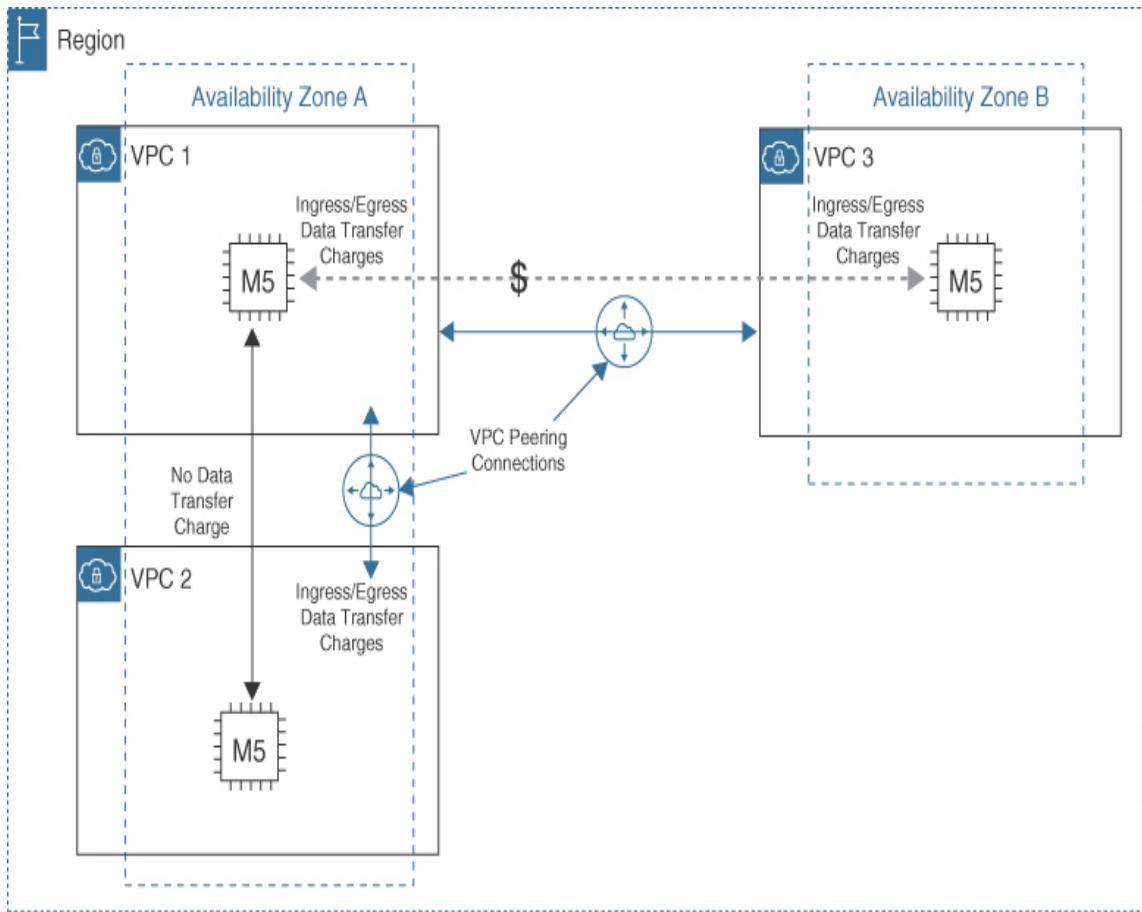


Figure 15-15 VPC Peering Charges Across AWS Regions

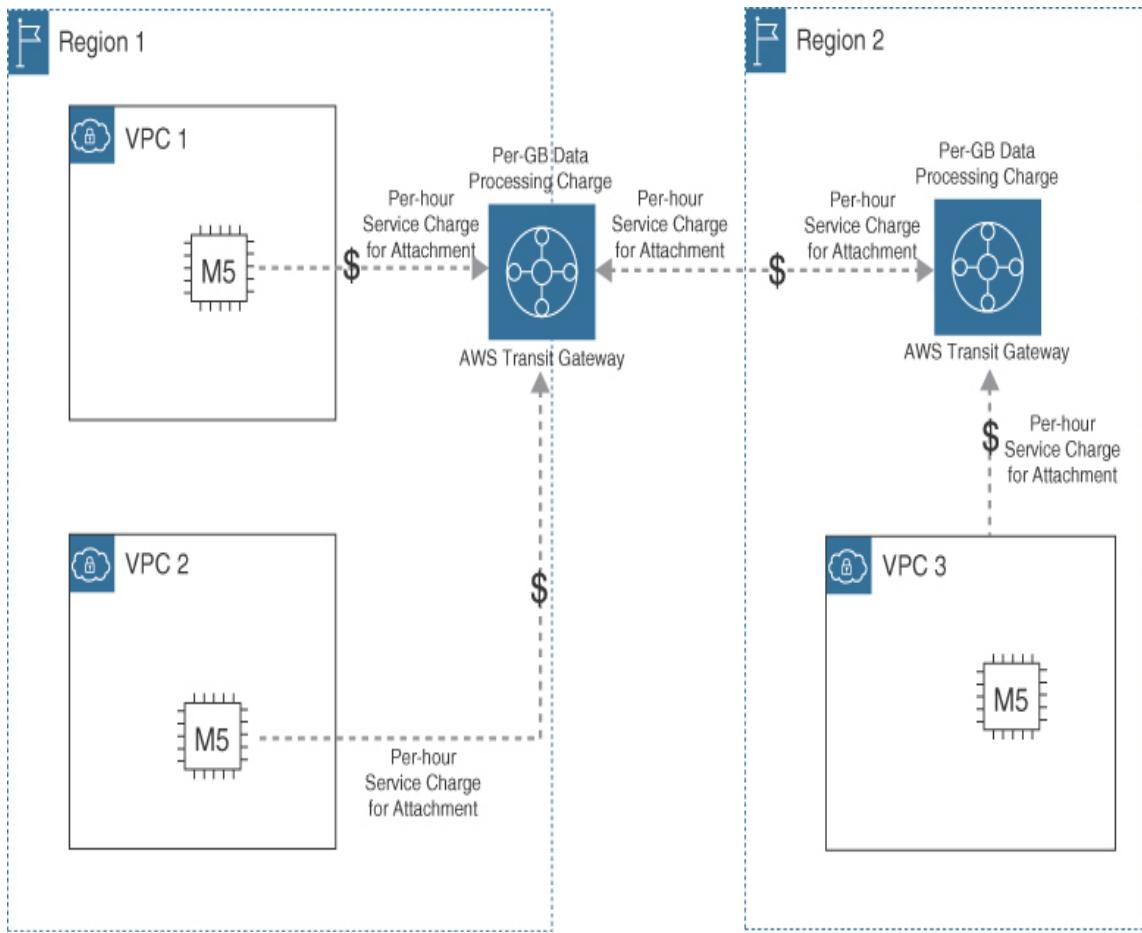


Figure 15-16 Transit Gateway Peering Charges Across AWS Regions

Data Transfer at Edge Locations

Key Topic

Viewer requests into an edge location from the Internet is free of charge. Cached data that is transferred outbound from AWS

edge locations to the Internet is billed at region-specific tiered data transfer rates.

There are three services that can be deployed (at additional cost) to improve the speed of data transfer between AWS edge locations and end users:

- **Amazon S3 Transfer Acceleration:** Optimizes transfer of files over long distances between a client location and a single S3 bucket, taking advantage of edge locations; ingress data is routed into an edge location and across Amazon's private network. Upload speeds comparing S3 direct upload speeds to S3 Transfer Acceleration speeds can be found here: <http://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html>.
- **AWS Global Accelerator:** Route users' requests across the AWS private network to the AWS application using the closest edge location. AWS Global Accelerator charges a fixed hourly fee and data transfer fees. Additional charges are for each accelerator provisioned and the amount of traffic that flows through the accelerator. There is also an EC2 data transfer out fee charged for application endpoints per hosted region.

- **AWS Site-to-Site VPN:** An AWS Site-to-Site VPN connects to a VPC or AWS Transit Gateway using IPsec tunnels. Charges are for the Site-to-Site VPN connection per hour and data transfer out charges. AWS Global Accelerator can be used with the Accelerated Site-to-site VPN option routing incoming VPN traffic from the on-premises network to the AWS edge location that is closest to your customer gateway.

Consider the following example: There is a Site-to-Site VPN connection to your Amazon VPC in us-east-2 (Ohio) from your on-premises location. The connection is active for 30 days, 24 hours a day. 2,000 GiB is transferred into the VPC; 800 GiB is transferred out through the site-to-site VPN connection.

- **AWS Site-to-Site VPN connection fee:** While connections are active, there is an hourly fee of \$0.05 per hour; \$36.00 per month in connection fees.
- **Data transfer out fee:** The first 100 GiB transferred out is free; you pay for 700 GiB at \$0.09 per GiB, paying \$63.00 per month in data transfer out fees.
- **Total charges:** \$99.00 per month for the active AWS Site-to-Site VPN connection.

Network Data Transfer

The design of your workload and its use of AWS network services will greatly determine your data transfer costs.

Network data transfer costs to understand are as follows:

- AWS services that are hosted in the same region but that are in separate AZs are charged for outgoing data transfer at \$0.01 per GiB.
- When data is transferred between AWS services hosted in different AWS regions, the data transfer charge is \$0.02 per GiB.
- Data is transferred within the same AWS region and staying within the same AZ is free of charge when using private IP addresses.
- If you are using an AWS-assigned public IP address or an assigned elastic IP public address, there are charges for data transfer out from the EC2 instance. These charges are per GiB transfer, and the minimum charge is \$0.01 per GiB.
- Data transfers between EC2 instances, AWS services, or containers using elastic network interfaces in the same availability zone and same VPC using private IPv4 or IPv6 addresses are free of charge. A common example is RDS synchronous database replication from the primary database node to the standby database node. Across AZs, there are data transfer charges for RDS synchronous replication.

Note

Always use private IP addresses rather than public IP addresses, sending data with public IP addresses is charged.

- Different AWS regions have different egress data transfer costs. If possible, architect your applications and systems for minimal data transfer across AWS regions or AZs.

Note

The AWS pricelist API enables you to query AWS for the current prices of AWS products and services using either JSON or HTML. For example, <https://pricing.us-east-1.amazonaws.com/offers/v1.0/aws/AmazonS3/current/us-east-1/index.csv>.

Public Versus Private Traffic Charges

Public traffic sent to a public IP address traveling across the Internet will incur a much higher data transfer charge than private IP address traffic. Private traffic traveling within AWS is

always cheaper than traffic on a public subnet. Wherever possible, AWS uses the private network for communication.

Private traffic that stays within a single subnet incurs no additional charges, whereas private traffic that travels across multiple private subnets that are hosted by different AZs incurs an egress charge of \$0.01 per GiB. One of the tasks to carry out for each hosted application is to create a detailed costing of its design, including charges for the replication of the databases across AZs, the monthly charge, and the traffic flow charges (see [Figure 15-17](#)).

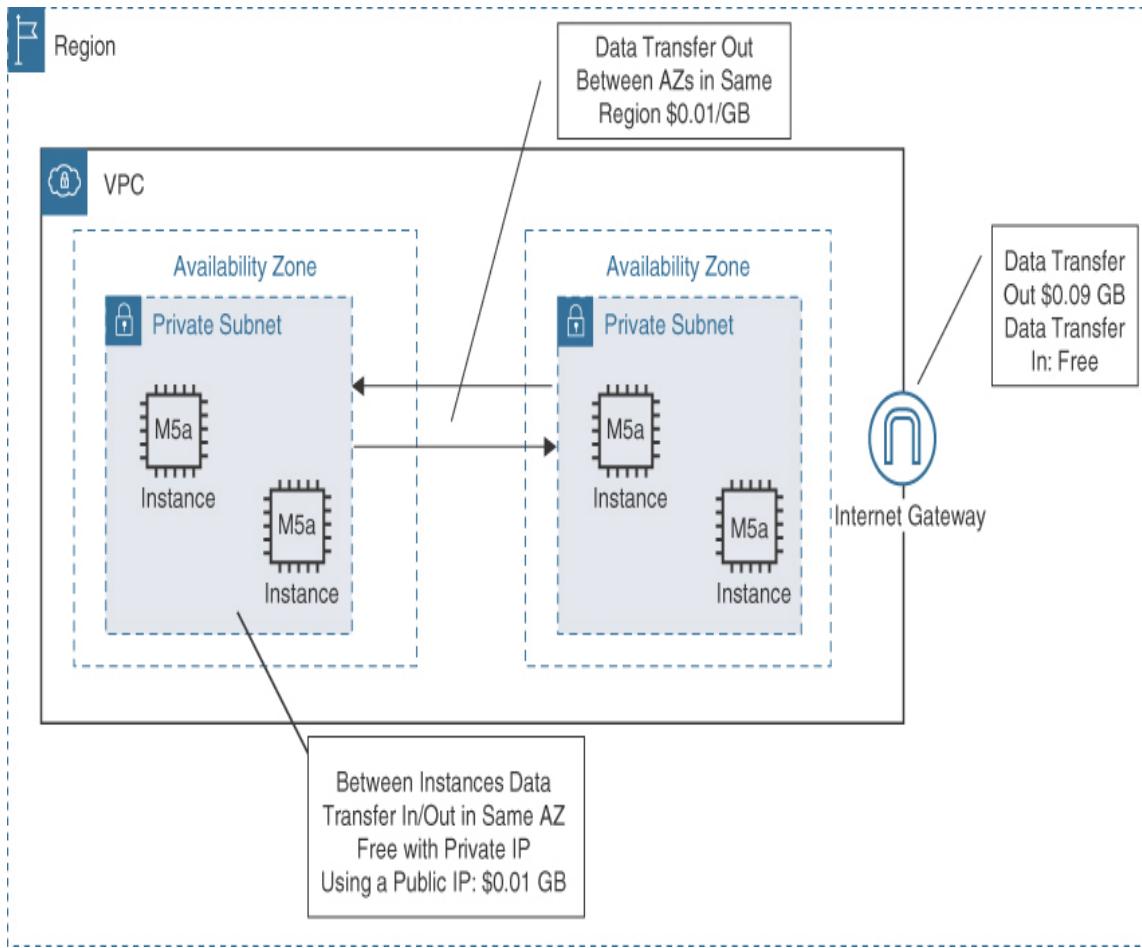


Figure 15-17 Traffic Charges at AWS

Network traffic charges are calculated based on the location of your data records, the location of end users, and the size and volume of all data transfers.

- **Data transfer modeling:** It is important to understand the locations where data transfer occurs during the operation of a workload, as well as the cost of the data transfer. For example, you might have a custom database solution that

replicates across multiple AZs within a single region. It would be more cost-effective to deploy a database using Amazon RDS if the database engine you are currently using is supported. One of the advantages of deploying an Amazon RDS database is that there are no additional charges for synchronous replication between the primary and standby database instances. Use AWS Cost Explorer or the AWS Cost and Usage Report to review the details of data transfer costs. (Details on AWS cost and budgeting tools are provided in [Chapter 10, “Determining High-Performing Database Solutions.”](#))

- **Use a dedicated network connection:** Perhaps it would be most efficient to establish a dedicated network connection to AWS using an AWS Direct Connect connection. AWS Direct Connect can be ordered in increments from 512 Mbps up to 100 GBps. Advantages include increased security due to the fiber connection being a private connection and faster network performance. Direct Connect involves two charges: port hours used and data transfer. AWS Direct Connect charges are for the bandwidth utilized and the number of connected hours used per month. A 1-Gbps connection rate would be \$0.30 per hour, whereas a 10-GiBps connection would be slightly over \$2.00 per hour. Perhaps the cost of deploying a faster connection is cheaper overall as more data

could be processed across a high-speed AWS Direct Connect connection.

- **Changing to private endpoints:** VPC endpoints allow interface and gateway connectivity between most AWS services over Amazon's private network. Data transfer over private network connections is always faster than public data transfer. There are no charges for VPC gateway connections.

Data Transfer Costs Cheat Sheet



For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following aspects of improving data transfer costs:

- Operate within the same AWS region. If possible, operate within the AWS region that has the lowest data transfer rates.
- Operate within a single AZ. Operating within the same AZ and the same VPC using private IP addresses incurs no data transfer costs.
- If the users of your application are spread out across a large geographic area and access application data across the

Internet, consider using AWS CloudFront. Data transferred out using AWS CloudFront will be less expensive and much faster than public data transfers across the Internet.

- Capture information about IP traffic within a VPC by enabling VPC flow logs.
- Use AWS CloudFront for caching content and reducing the load on the origin servers and data location.
- Avoid using public IPv4 addresses or EIP addresses as costs are higher than private IPv4 addresses.
- RDS Multi-AZ deployments include the data transfer costs for replication between primary and alternate database servers.
- Amazon EFS deployments have both single-AZ or Multi-AZ deployment options. A single-AZ deployment can save up to 40% in storage costs.
- VPC gateway endpoints have no data transfer charges when communicating with Amazon S3 and Amazon DynamoDB within the same region.
- VPC interface endpoints are charged hourly service and data transfer charges.
- Amazon EFS and Amazon RDS deployments have free cross-AZ data transfers.
- Amazon CloudFront has free data transfers for **GET** requests.
- AWS Simple Monthly Calculator allows you to review pricing and data transfer costs for most AWS services.

- AWS Pricing Calculator helps estimate the total price for your workload deployment at AWS.

Exam Preparation Tasks

As mentioned in the section “[How to Use This Book](#)” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. [Table 15-4](#) lists these key topics and the page number on which each is found.



Table 15-4 [Chapter 15](#) Key Topics

Key Topic Element	Description	Page Number
Note	An LCU measures the hourly characteristics of network traffic processed by each deployed load balancer.	695
<u>Table</u> <u>15-2</u>	Cost Comparison for CloudFront Costs	699
Section	CloudFront Pricing Cheat Sheet	699
<u>Figure</u> <u>15-5</u>	Accessing Elastic Container Registry Using an Interface Endpoint	701
Section	Network Services from On-Premises Locations	703
Section	Accessing AWS Services in the Same Region	707

Section	Accessing AWS Services in Different Regions	710
---------	---	-----

Section	Data Transfer at Edge Locations	713
---------	---------------------------------	-----

Section	Data Transfer Costs Cheat Sheet	716
---------	---------------------------------	-----

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

[Load Balancer Capacity Unit \(LCU\)](#)

[time to live \(TTL\)](#)

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep Software Online.

1. What are the two main components of calculating management service costs at AWS that are applied to every service?

- 2.** What is the key driver behind data transfer costs?
- 3.** What is the term for calculating costs depending on the usage of a service or resource?
- 4.** What are the four dimensions of a Load Balancer Capacity Unit (LCU)?

Chapter 16

Final Preparation

Has reading this book made you feel prepared and ready to take the AWS Certified Solutions Architect – Associate (SAA-C03) exam? I sure hope so. This chapter gives you specific details on certification prep, including the certification exam itself.

This chapter shares some great ideas on ensuring that you ace your upcoming exam. If you have read this book with the primary goal of mastering the AWS cloud without really considering certification, maybe this chapter will convince you to give the exam a try.

The first 15 chapters of this book covered the technologies, protocols, design concepts, and technical details required to be prepared to pass the AWS Certified Solutions Architect – Associate (SAA-C03) exam. This chapter provides a set of tools and a study plan to help you complete your preparation for the exam to supplement everything you have learned up to this point in the book.

This short chapter has four main sections. The first section provides information on the AWS Certified Solutions Architect –

Associate (SAA-C03) exam. The second section shares some important tips to keep in mind to ensure that you are ready for the exam. The third section discusses exam preparation tools that may be useful at this point in the study process. The final section provides a suggested study plan you can implement now that you have completed all the earlier chapters in this book.

Exam Information

Here are details you should be aware of regarding the AWS Certified Solutions Architect – Associate (SAA-C03) exam:

Question types: Multiple-choice and multiple-response

Number of questions: 65

Time limit: 130 minutes

Available languages (at a testing center): English, French, German, Italian, Portuguese, Spanish, Japanese, Simplified Chinese, and Korean

Available languages used by proctors of online exam:
English (Pearson VUE/PSI) and Japanese (VUE)

Online exam appointments: 24 hours a day, 7 days a week

Test providers: Pearson VUE or PSI

Exam fee: \$150

Exam ID code: SAA-C03

Delivery method: Testing center or online proctored exam from your home or office location

This exam seeks to ensure that a candidate attaining the AWS Certified Solutions Architect – Associate certification has the following required knowledge:

- Knowledge and skills in the following AWS services: compute, networking, storage, and database and deployment and management services
- Knowledge and skills in deploying, managing, and operating AWS workloads and implementing security controls and compliance requirements
- The ability to identify which AWS service meets technical requirements
- The ability to define technical requirements for AWS-based applications
- The ability to identify which AWS services meet a given technical requirement

The exam is broken up into four different domains. Here are those domains and the percentage of the exam for each of the domains:

- **Design Secure Architectures:** 30%
- **Design Resilient Architectures:** 26%
- **Design High-Performing Architectures:** 24%
- **Design Cost-Optimized Architectures:** 20%

Here is the breakdown of the task statements for the domains:

- **Domain 1: Design Secure Architectures**
 - Task Statement 1: Design secure access to AWS resources
 - Task Statement 2: Design secure workloads and applications
 - Task Statement 3: Determine appropriate data security controls
- **Domain 2: Design Resilient Architectures**
 - Task Statement 1: Design scalable and loosely coupled architectures
 - Task Statement 2: Design highly available and/or fault-tolerant architectures
- **Domain 3: Design High-Performing Architectures**
 - Task Statement 1: Design high-performing and/or scalable storage solutions

- Task Statement 2: Design high-performing and elastic compute solutions
- Task Statement 3: Determine high-performing database solutions
- Task Statement 4: Determine high-performing and/or scalable network architectures
- Task Statement 5: Determine high-performing data ingestion and transformation solutions
- **Domain 4: Design Cost-Optimized Architectures**
 - Task Statement 1: Design cost-optimized storage solutions
 - Task Statement 2: Design cost-optimized compute solutions
 - Task Statement 3: Design cost-optimized database solutions
 - Task Statement 4: Design cost-optimized network solutions

Note the following important information about the AWS Certified Solutions Architect – Associate (SAA-C03) exam:

- You can decide which exam format to take: You can either go to a testing center or take a proctored exam from a personal location like your office or home.
- After you have scheduled your exam, you will receive a confirmation email from the test provider that provides details related to your exam appointment.

- If you are taking the test in person at a testing center, remember to bring two forms of ID on the day of the test. At least one must be a signed ID with a photo, such as a driver's license, passport, or health card. The second ID can be a credit card.
- If you are taking an online proctored exam, make sure to run the system test provided by the selected exam provider to ensure that your computer is acceptable for the exam. Both VUE and PSI have system tests.
- If you are taking an online proctored exam, your exam can be started up to 30 minutes before the scheduled exam time, but if you are more than 15 minutes late for your appointment, you won't be able to start your exam.
- Breaks are allowed when you're taking a test at a testing center, but they are not allowed during an online proctored exam. With an online proctored exam, you are not allowed to move out of the view of your webcam during your appointment.
- Make sure that your exam space at home or at your office remains private. If somebody comes into your office or private space during the exam, you will not be allowed to continue the exam.

Tips for Getting Ready for the Exam

Here are some important tips to keep in mind to ensure that you are ready for the AWS Certified Solutions Architect – Associate exam, some of which apply only to taking the exam at a testing center and others that apply in all cases:

- **Build and use a study tracker:** Consider using the task statements shown in this chapter to build a study tracker for yourself. Such a tracker can help ensure that you have not missed anything and that you are confident for your exam. As a matter of fact, this book offers a sample study planner as a website supplement.
- **Log in to the AWS management portal and write down a quick one-sentence or point-form description of each AWS service:** Writing down this information will put it into long-term memory and will be very helpful when you're trying to decipher test questions.
- **Think about your time budget for questions on the exam:** When you do the math, you will see that, on average, you have 2 minutes per question. Although this does not sound like a lot of time, keep in mind that many of the questions will be very straightforward, and you will take 15 to 30 seconds on those. This leaves you extra time for other questions on the exam.
- **Watch the clock:** Check in on the time remaining periodically as you are taking the exam. You might even find

that you can slow down pretty dramatically if you have built up a nice block of extra time.

- **Get some earplugs:** The testing center might provide earplugs, but get some just in case and bring them along. There might be other test takers in the center with you, and you do not want to be distracted by their moans and groans. I personally have no issue blocking out the sounds around me, so I never worry about this, but I know it is an issue for some.
- **Plan your travel time:** Give yourself extra time to find the test center and get checked in. Be sure to arrive early. As you test more frequently at a particular center, you can certainly start cutting it closer time-wise.
- **Get rest:** Most students report that getting plenty of rest the night before the exam boosts their success. All-night cram sessions are not typically successful.
- **Bring in valuables but get ready to lock them up:** The testing center will take your phone, your smartwatch, your wallet, and other such items and will provide a secure place for them.
- **Take notes:** You will be given note-taking implements and should not be afraid to use them. I always jot down any questions I struggle with on the exam. I then memorize them at the end of the test by reading my notes over and over

again. I always make sure I have a pen and paper in the car, and I write down the issues in the parking lot just after the exam. When I get home—with a pass or fail—I research those items!

- **Use the FAQs in your study:** The Amazon test authors have told me they love to pull questions from the FAQs they publish at the AWS site. These are a really valuable read anyway, so go through them for the various services that are key for this exam.
- **Brush up with practice exam questions:** This book provides many practice exam questions. Be sure to go through them thoroughly. Don't just blindly memorize answers; use the questions to see where you are weak in your knowledge and then study up on those areas.

Scheduling Your Exam

You can schedule your AWS Certified Solutions Architect – Associate (SAA-C03) exam through the web portal

<https://www.aws.training/certification>; see [Figure 16-1](#). If you haven't yet created an AWS Training and Certification account, you need to create one now, and then you can schedule your exam.

AWS Certification

AWS Certification helps learners build credibility and confidence by validating their cloud expertise with an industry-recognized credential, and organizations identify skilled professionals to lead cloud initiatives using AWS.

[Learn more](#) on how to prepare for your exams.

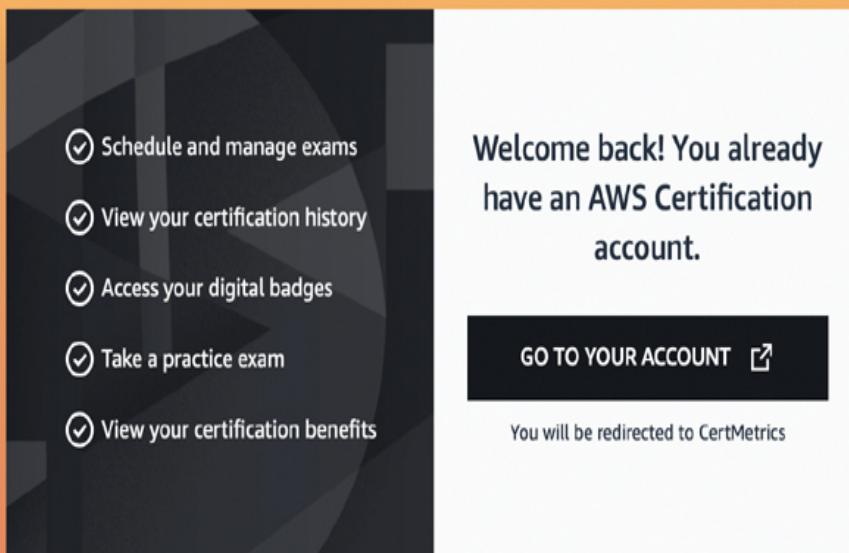


Figure 16-1 Schedule Your AWS Exam

After you've taken your exam, you can return to this portal to download your AWS digital badge, schedule another test, or set up sharing of your transcript to your current or future employer, as shown in [Figure 16-2](#).



Figure 16-2 Viewing Your AWS Training Profile

Tools for Final Preparation

This section provides some information about the available tools and how to access them.

Pearson Test Prep Practice Test Software and Questions on the Website

Register this book to get access to the Pearson Test Prep practice test software, which displays and grades a set of exam-realistic multiple-choice questions. Using Pearson Test Prep practice test software, you can either study by going through the questions in Study mode or take a simulated (timed) AWS Certified Solutions Architect – Associate (SAA-C03) exam.

The Pearson Test Prep practice test software comes with two full practice exams. These practice exams are available to you either online or as an offline Windows application. To access the practice exams that were developed with this book, see the instructions in the card inserted in the sleeve at the back of the book. This card includes a unique access code that enables you to activate your exams in the Pearson Test Prep software.

Accessing the Pearson Test Prep Software Online

The online version of this software can be used on any device with a browser and connectivity to the Internet, including desktop machines, tablets, and smartphones. To start using your practice exams online, simply follow these steps:

Step 1. Go to <https://www.PearsonTestPrep.com>.

Step 2. Select Pearson IT Certification as your product group.

Step 3. Enter your email and password for your account. If you don't have an account on <PearsonITCertification.com> or <CiscoPress.com>, you need to establish one by going to <PearsonITCertification.com/join>.

Step 4. In the My Products tab, click the Activate New Product button.

Step 5. Enter the access code printed on the insert card in the back of your book to activate your product. The product is then listed in your My Products page.

Step 6. Click the Exams button to launch the exam settings screen and start the exam.

Accessing the Pearson Test Prep Software Offline

If you wish to study offline, you can download and install the Windows version of the Pearson Test Prep software. You can find a download link for this software on the book's companion website, or you can just enter this link in your browser for a direct download:

<https://www.pearsonitcertification.com/content/downloads/pcpt/engine.zip>

To access the book's companion website and the software, simply follow these steps:

Step 1. Register your book by going to PearsonITCertification.com/register and entering the ISBN: 9780137941582.

Step 2. Respond to the challenge questions.

Step 3. Go to your account page and select the Registered Products tab.

Step 4. Click the Access Bonus Content link under the product listing.

Step 5. Click the Install Pearson Test Prep Desktop Version link in the Practice Exams section of the page to download the

software.

Step 6. After the software finishes downloading, unzip all the files on your computer.

Step 7. Double-click the application file to start the installation and follow the onscreen instructions to complete the registration.

Step 8. After the installation is complete, launch the application and click the Activate Exam button on the My Products tab.

Step 9. Click the Activate a Product button in the Activate Product Wizard.

Step 10. Enter the unique access code found on the card in the back of your book and click the Activate button.

Step 11. Click Next, and then click the Finish button to download the exam data to your application.

Step 12. You can now start using the practice exams by selecting the product and clicking the Open Exam button to open the exam settings screen.

The offline and online versions will sync together, so saved exams and grade results recorded in one version will also be

available to you on the other.

Customizing Your Exams

When you are in the exam settings screen of Pearson Test Prep, you can choose to take exams in one of three modes:

- Study mode
- Practice Exam mode
- Flash Card mode

Study mode enables you to fully customize an exam and review answers as you are taking the exam. This is typically the mode you use first to assess your knowledge and identify information gaps. Practice Exam mode locks certain customization options in order to present a realistic exam experience. Use this mode when you are preparing to test your exam readiness. Flash Card mode strips out the answers and presents you with only the question stem. This mode is great for late-stage preparation, when you really want to challenge yourself to provide answers without the benefit of seeing multiple-choice options. This mode does not provide the detailed score reports that the other two modes provide, so it is not the best mode for helping you identify knowledge gaps.

In addition to these three modes, you will be able to select the source of your questions. You can choose to take exams that cover all of the chapters, or you can narrow your selection to just a single chapter or the chapters that make up specific parts in the book. All chapters are selected by default. If you want to narrow your focus to individual chapters, simply deselect all the chapters and then select only those on which you wish to focus in the Objectives area.

You can also select the exam banks on which to focus. Each exam bank comes complete with a full exam of questions that cover topics in every chapter. You can have the test engine serve up exams from all four banks or just from one individual bank by selecting the desired banks in the exam bank area.

There are several other customizations you can make to your exam from the exam settings screen, such as the time allowed for taking the exam, the number of questions served up, whether to randomize questions and answers, whether to show the number of correct answers for multiple-answer questions, and whether to serve up only specific types of questions. You can also create custom test banks by selecting only questions that you have marked or questions on which you have added notes.

Note

The first time you run some test questions, you might find that you do badly. Don't worry about it. Your brain may simply be resisting taking a multiple-choice exam! Run the test engine again, and your brain will eventually give in and start focusing on the questions. Also, remember that as good as Pearson Test Prep and the practice exam questions are, they are not the real exam. That's okay, because what you're actually learning by answering the sample test questions is how to answer multiple-choice questions similar to the questions on the real exam.

Updating Your Exams

If you are using the online version of the Pearson Test Prep software, you should always have access to the latest version of the software as well as the exam data. If you are using the Windows desktop version, every time you launch the software, it will check to see if there are any updates to your exam data and automatically download any changes made since the last

time you used the software. This requires that you be connected to the Internet at the time you launch the software.

Sometimes, due to a number of factors, the exam data might not fully download when you activate your exam. If you find that figures or exhibits are missing, you might need to manually update your exams.

To update a particular exam you have already activated and downloaded, simply select the Tools tab and click the Update Products button. Again, this is only an issue with the desktop Windows application.

If you wish to check for updates to the Windows desktop version of the Pearson Test Prep exam engine software, simply select the Tools tab and click the Update Application button. Doing so allows you to ensure that you are running the latest version of the software engine.

Premium Edition

In addition to the free practice exam provided on the website, you can purchase additional exams with expanded functionality directly from Pearson IT Certification. The Premium Edition of this title contains an additional two full practice exams and an eBook (in both PDF and ePUB formats).

In addition, the Premium Edition includes remediation information for each question and links to the specific part of the eBook that relates to that question.

Because you have purchased the print version of this title, you can purchase the Premium Edition at a deep discount. A coupon code in the book sleeve contains a one-time-use code and instructions for where you can purchase the Premium Edition.

To view the Premium Edition product page, go to
<https://www.informit.com/title/9780137941568>.

Chapter-Ending Review Tools

Chapters 2 through 15 include several features in the “Exam Preparation Tasks” and “Q&A” sections at the end of the chapter. You might have already worked through these in each chapter. Using these tools again can also be useful as you make your final preparations for the exam.

Suggested Plan for Final Review/Study

This section provides a suggested study plan from the point at which you finish reading through Chapter 15 until you take the AWS Certified Solutions Architect – Associate (SAA-C03) exam. You can ignore this plan, use it as is, or take suggestions from it.

The plan involves three steps:

Step 1. Review key topics and “Do I Know This Already?”

(DIKTA?) questions: You can use the table that lists the key topics in each chapter or just flip the pages looking for key topics. Also, reviewing the DIKTA? questions from the beginning of the chapter can be helpful for review.

Step 2. Review “Q&A” sections: Go through the Q&A questions at the end of each chapter to identify areas where you need more study.

Step 3. Use the Pearson Test Prep to practice: You can use the Pearson Test Prep practice test engine to study, using a bank of unique exam-realistic questions available only with this book.

Summary

The tools and suggestions provided are meant to help you develop the skills required to pass the AWS Certified Solutions Architect – Associate (SAA-C03) exam. This book has been developed from the beginning to not only tell you the facts but also to help you learn how to apply the facts. No matter what your experience level leading up to taking the exam, I hope that the broad range of preparation tools and the structure of this book help you pass the exam.

Appendix A

Answers to the “Do I Know This Already?” Quizzes and Q&A Sections

Chapter 2

“Do I Know This Already?”

1. b

2. d

3. b

4. a

5. d

6. d

Q&A

1. administrative controls

2. downtime

3. monitoring

4. reliability

5. cost

6. amount, useable

7. horizontally, dynamic

8. template file

Chapter 3

“Do I Know This Already?”

1. b

2. d

3. b

4. b

5. b

6. b

7. b

8. b

9. b

10. b

11. b

12. b

13. c

14. c

Q&A

1. If you're using the root account, you will be able to change your AWS account settings from the management console. You will also be using an email address and password to log into AWS.

2. The best method to provide applications secure access to AWS services is to create an IAM role and assign it to the application by adding the role to the EC2 server where the application is hosted.

3. The advantage of using a resource-based policy is that all of the entities that need access to the resource must be named in the policy. In comparison with an identity-based policy, the IAM

user is not named in the policy directly, as the policy is directly attached to the IAM user, thereby allowing a level of access. This could be great; but consider the situation where an IAM policy is mistakenly added to an IAM user and providing access. This situation could not occur with a resource policy, as each entity that requires access must be named within the policy itself.

- 4.** The best method for controlling access to AWS resources is through the use of IAM roles. First, IAM roles provide temporary access to a resource; secondly, the role's access keys are controlled by the secure token service (AWS STS); and finally, using IAM roles means you don't have to create as many IAM users.
- 5.** Although inline policies may serve a need, such as specifying that just a specific IAM user will have access, the issue is documentation; that is, remembering what you may have done. Administrators must carefully document inline policies.
- 6.** The Policy Simulator can be used to check your IAM policies if they are not working properly.
- 7.** AWS Organizations helps you manage multiple AWS accounts in a treelike structure, allowing you to take advantage of

consolidated billing, centralized security settings using service control policies, and sharing AWS resources.

8. To run a script at the command-line interface, you must first install the AWS CLI appropriate to your operating system (Linux, Windows, or macOS). Next, you must have the access key and secret access keys of the IAM user that you will be using to execute commands from the command line interface. The final step is that you must execute AWS Configure and enter the AWS region you will be running the script in, and add your access key and secret access key.

Chapter 4

“Do I Know This Already?”

1. c

2. b

3. b

4. c

5. c

6. b

7. a

8. c

9. c

10. a

11. b

12. c

Q&A

1. All networking services provided by AWS are software services. For example, routers and load balancers and all network services are software appliances.

2. A network ACL has the ability to block a specific IP address, whereas a security group does not have that ability.

3. Because CloudTrail provides API authentication and monitoring for all AWS regions where your AWS account is operational, you can track activity in AWS regions in which you do not want to be operating.

- 4.** AWS Secrets Manager allows you to store third-party secrets securely.
- 5.** GuardDuty uses machine learning.
- 6.** A Direct Connect gateway can allow you to connect to multiple VPCs in different AWS regions to provide high-speed connectivity.
- 7.** Network assessment checks do not require the Inspector agent to be installed.
- 8.** Purchasing Business support for your AWS account enables all checks for Trusted Advisor.

Chapter 5

“Do I Know This Already?”

1. a

2. b

3. a

4. a

5. c

6. c

7. c

8. c

9. c

10. c

Q&A

1. There is no single-tenant data store service available with AWS. All data stores at AWS are multi-tenant by design. Protection of data records is carried out by enabling encryption.

2. All public access for a newly created S3 bucket is blocked until each organization makes a decision to make the S3 bucket public.

3. SSE-C encryption uses an organization-provided encryption key for both encryption and decryption. The provided key is discarded after use and must be resupplied by the organization each time. Therefore, there is no potential security risk with stored encryption keys at AWS.

- 4.** Envelope encryption involves a hierarchy of security when working with AWS KMS. AWS KMS creates data keys for encryption and decryption that are associated with a specific CMK. The keys cannot work with any other CMK and are controlled by AWS KMS.
- 5.** Amazon S3 Glacier objects are automatically encrypted when stored in vaults.
- 6.** AWS CloudHSM is a hardware storage module that is maintained by AWS. AWS backs up the contents of AWS CloudHSM, but the only person who can access the contents is the assigned organization.
- 7.** AWS KMS does not support the rotation of private keys that were imported.
- 8.** A private CA can be used to create a private CA that can renew and deploy certificates for private-facing resources such as a network load balancer deployed on private subnets.

Chapter 6

“Do I Know This Already?”

- 1.** a

2. b

3. b

4. b

5. b

6. b

7. b

8. c

9. b

10. a

Q&A

1. A sticky session has the advantage of ensuring that the end user who establishes a session with an application server can continue to communicate with that application server for the life of the session. However, the disadvantage is that if the application server fails, the user is sent to another application server, which will know nothing about the previous session. This might not be a huge issue if the user is merely reading

reports, as the user could simply start again. However, a user in the midst of purchasing a product would have to start over.

2. One advantage of using a central location to store user state information is that the storage location is redundant, and it operates in memory and therefore is fast. However, the main advantage of using a centralized storage location is that the user state information is stored in an independent location and not at the user location or at the server location, which provides a higher level of redundancy and availability.

3. Because every AWS service allows you to link issues with the associated service directly with SNS, you have the ability to respond to issues at any time, either with manual steps or through automated solution steps.

4. SQS can be a client of SNS notifications. SNS can send messages to specific queues, which have, in turn, application servers as clients. Therefore, an upload of the file to an S3 bucket could prompt a notification, which could be passed on to a queue, which could be processed by the associated application servers automatically.

5. Step Functions allows you to craft workflows using SQS and SNS and a variety of AWS services, through a GUI. Step

Functions has a logical component that can interface with the stateless services of a workflow.

6. Utilizing AWS Lambda to respond to notifications enables you to craft automated responses to any notifications that are generated.

7. Using AWS Lambda to create a serverless application allows you to focus on creating functions that map to the tasks in the application. For example, say that you enable an application that has five functions: Login, Search, Save, Download, and Logout. Using Lambda, you could create five separate functions and load those functions into Lambda. Then you could generate an application on your mobile device to call those functions as required. Each function would carry out its specific task when called. You are then charged only when the functions are called.

8. AWS Lambda can be used with API Gateway in this manner: An API call communicates with a custom Lambda function and carries out the tasks, as required, by calling various AWS services.

9. AWS Elastic Beanstalk allows you to deploy the required infrastructure to host an application that you have coded. Both the infrastructure and the hosting of the application can be

carried out automatically, including future application and infrastructure updates.

Chapter 7

“Do I Know This Already?”

1. a

2. a

3. b

4. c

5. c

6. c

7. b

8. a

9. b

10. c

11. d

12. c

13. a

14. d

Q&A

1. highly available and reliable

2. localized

3. high availability, fault tolerance, and reliability

4. regions

5. compliance

6. regions

7. traffic flow policies

8. service quota

Chapter 8

“Do I Know This Already?”

1. b

2. b

3. b

4. b

5. b

6. b

7. a

8. b

9. b

10. b

11. b

12. a

13. b

14. a

Q&A

- 1.** EFS storage has no size and sharing constraints. EBS storage can be shared across thousands of EC2 instances across multiple availability zones.
- 2.** Files stored in object storage such as S3 or S3 Glacier are stored and updated as entire files. In contrast, EBS storage can be updated block by block, either as storage changes or by snapshot.
- 3.** EBS io1 and io2 volumes support a feature called multi-attach, which allows you to attach and share the EBS volume to up to 16 instances, as long as all instances are hosted by the Nitro hypervisor.
- 4.** Before a snapshot is deleted, it is analyzed, and only the data that is exclusive to the snapshot copy is retained.
- 5.** The fastest storage that can be ordered at AWS is ephemeral storage, which is temporary storage volumes that are located on the bare-metal server where the instance is hosted. Because the EC2 instance is in exactly the same physical location as the bare-metal server, there is no network to traverse, and therefore the local storage volume is the fastest. However, ephemeral storage is not redundant; when the instance is turned off or fails, the storage is erased.

- 6.** In order to use S3 Lifecycle rules and Same-Region Replication and Cross-Region Replication, you must have versioning enabled.
- 7.** It is possible to share S3 objects with any person who does not have AWS credentials by creating and distributing a pre-signed URL. Temporary access is defined by configuring date and time expiration values.
- 8.** A WORM policy applied to an S3 bucket when operating in Compliance mode can never be removed—not even by Amazon.

Chapter 9

“Do I Know This Already?”

1. c

2. b

3. d

4. b

5. b

6. b

7. c

8. b

9. b

10. b

Q&A

1. template

2. A golden AMI is an image that is as perfect as possible, and no customizations or tweaks need to be made to it. You must have the right processes in place to properly test your images before they go into production. Once your application servers are in production, you do not need to troubleshoot your production servers if your AMIs have been fully tested in preproduction.

3. RAM/CPU, processing time

4. The CloudWatch agent has probably already been deployed. You might want to consider customizing the operation of the CloudWatch agent so that your application logs and system logs are sent to CloudWatch for analysis. CloudWatch can then send notifications when problems arise.

- 5.** Although launch configurations can be used to create EC2 instances that are performed by the Auto Scaling service as required, launch configurations are being deprecated and replaced with launch templates. Launch templates support all features of EC2 instances, whereas launch configurations do not support all current and new EC2 features.
- 6.** The recommended starting point is to deploy a target tracking policy, which allows you to define the level of application performance to be maintained. Auto Scaling adds and removes compute instances to meet this target level.
- 7.** When a workload's compute resources need to be scaled up but not aggressively; for example, an application that requires additional compute resources to be adjusted every 4 hours using a single metric such as CPU utilization. However, after changes have been made, a cooldown period must be observed before any additional changes can be implemented.
- 8.** Step scaling enables you to define the scaling of an application's compute levels based on multiple percentages when scaling both up and down. Therefore, step scaling allows you to carefully tune your scaling requirements.

Chapter 10

“Do I Know This Already?”

1. c

2. d

3. b

4. d

5. b

6. c

7. b

8. c

9. a

10. a

11. d

12. b

Q&A

- 1.** The advantage of using RDS to set up your databases is that after you describe the infrastructure and your needs to AWS, the database infrastructure is automatically set up and maintained for you, and even failover is automatically managed. All you have to do is work with your data records.
- 2.** The disadvantage of using RDS to set up most of your database types is that the standard deployment of RDS is a primary and standby database design. The other disadvantage is that RDS supports a set number of database engines and that's it; there's no flexibility. Of course, nothing stops organizations from building any database infrastructure design that required using custom EC2 instances or using RDS Custom.
- 3.** Read replicas can help improve database performance by taking the load off the primary database instance in regard to queries. The typical RDS deployment has a primary and standby database; however, the standby database is just that—it stands by and waits for disaster and does nothing else but make sure that it's up to date with the primary database instance. That's all well and good, but perhaps your primary database is becoming bogged down by handling all the queries by itself. Adding read replicas in regions where they are needed can increase performance by having the read replicas handle the queries in those specific regions.

4. AWS has two database solutions for multi-region deployments. The first is Amazon DynamoDB, a NoSQL solution that supports global tables that can span multiple AWS regions. The second solution is Amazon Aurora, which can also span multiple AWS regions with a global deployment. The only other option is a database solution that you build yourself using EC2 instances.

5. The difference between eventual consistency and strong consistency in regard to database storage, specifically Dynamo DB, is that you have a choice of living with the reality of replicated data to multiple locations, or you can choose strong consistency, which means that a check will be made to all the storage partitions to see which has the most current copies of data, and those copies will be presented to the application.

6. Amazon Aurora has a huge advantage over a standard MySQL deployment because of the data storage architecture utilizing a virtual SAN composed of SSD drives. The data is also stored in a cluster with multiple writers. As a result, the performance cannot really be compared; Aurora is much faster.

7. Continuous backups for all database solutions at AWS are stored in S3 storage.

8. The advantage of using ElastiCache to store user state information is speed, after all the cache storage is operating in RAM, and reliability. Rather than storing the user state directly on the EC2 instance where the user session is taking place, the user's session information is stored in another location just in case the EC2 instance that the end user is communicating with fails. If there is a failure, and the user begins communicating with another web server, the user's session information can be retrieved from ElastiCache, allowing the continuation of the user's session on the other server.

Chapter 11

“Do I Know This Already?”

1. c

2. c

3. b

4. b

5. a

6. c

7. d

8. b

9. a

10. d

11. d

12. d

13. b

14. b

Q&A

1. CloudFront edge locations are located worldwide, enabling an application hosted in a single AWS region to cache data records to users located anywhere in the world. CloudFront integration with an application allows the caching of requested data records to the edge location closest to the end user.

2. Gateway Load Balancer can be deployed to manage multiple third-party load balancer virtual appliances and manage

performance by scaling the virtual appliances up or down, based on demand.

3. The CIDR address range you choose for your VPC should be large enough to host all of the available instances that you will need for your application stack.

4. You should use separate VPCs for development, testing, and production environments.

5. There is no good reason for using public IP addresses for your web servers. Instead, locate your web servers behind load balancers hosted on public subnets. Your web servers should be hosted on private subnets that protect your web servers from direct Internet access.

6. You can move your public IP addresses to AWS by using a bring-your-own IP (BYOIP) address service.

7. All networking services provided by AWS are software services. For example, routers and load balancers and all network services are software appliances.

8. A network ACL has the ability to block a specific IP address, whereas a security group does not.

9. Elastic IP addresses can be used to add static public IP addresses to an EC2 instance or can be assigned to a NAT gateway service.

10. Endpoint interface or gateway connections ensure that traffic to the selected AWS service remains on the AWS private network.

Chapter 12

“Do I Know This Already?”

1. b

2. c

3. b

4. c

5. c

6. c

7. d

8. c

9. d

10. c

11. a

12. d

Q&A

- 1.** The two main components of calculating management service costs at AWS that are applied to every service are the compute and storage used to carry out the management service.
- 2.** Data transfer costs are incurred for the egress traffic from an AWS availability zone or an AWS region to another availability zone or region.
- 3.** Tiered pricing is calculated based on the usage of a service or resource.
- 4.** The two additional components of storage charges, in addition to data transfer charges, are storage and retrieval pricing.
- 5.** After cost allocation tags have been activated, AWS uses the tags to organize and display costs on cost allocation reports,

making it easier to track costs.

6. An S3 lifecycle rule defines the management of S3 objects on a defined schedule, such as movement or deletion after a defined timeframe, whereas an AWS Backup lifecycle management policy defines the storage tier location where the backup is stored: either warm storage or cold storage.

7. Backups can be copied to multiple AWS regions on demand or automatically as part of a defined backup plan.

8. An AWS Storage Gateway volume gateway provides block storage to on-premises applications using iSCSI connections that store the volumes in S3. An AWS Storage Gateway file gateway allows storage and retrieval of objects stored in S3 using NFS or SMB protocols.

Chapter 13

“Do I Know This Already?”

1. b

2. b

3. b

4. b

5. d

6. c

Q&A

1. virtual CPU

2. EC2 instance

3. per-VM

4. compute charge

5. pricing discounts

6. capacity reservations

7. regional reservation, anywhere

8. availability zone

Chapter 14

“Do I Know This Already?”

1. b

2. c

3. d

4. b

5. d

6. a

Q&A

1. per request

2. hourly

3. database schema

4. standard SQL queries

5. MongoDB compatibility

6. length of time

7. incremental

8. AWS Backup

Chapter 15

“Do I Know This Already?”

1. a

2. b

3. c

4. a

Q&A

1. The two main components of calculating management service costs at AWS that are applied to every service are the compute and storage used to carry out the management service.
2. The key driver behind data transfer costs is the egress traffic from AWS or from an availability zone or from a region to an external location.
3. With a tiered, or sliding, price point based on usage, costs are calculated depending on the usage of a service or resource.
4. The four dimensions of an LCU are new connections, active connections, processed bytes, and rule evaluations.

Appendix B

AWS Certified Solutions Architect – Associate (SAA-C03) Cert Guide Exam Updates

Over time, reader feedback allows Pearson to gauge which topics give our readers the most problems when taking the exams. To assist readers with those topics, the authors create new materials clarifying and expanding on those troublesome exam topics. As mentioned in the Introduction, the additional content about the exam is contained in a PDF on this book's companion website, at

[https://www.pearsonITcertification.com/title/9780137941582.](https://www.pearsonITcertification.com/title/9780137941582)

This appendix is intended to provide you with updated information if Amazon makes minor modifications to the exam upon which this book is based. When Amazon releases an entirely new exam, the changes are usually too extensive to provide in a simple update appendix. In those cases, you might need to consult the new edition of the book for the updated content. This appendix attempts to fill the void that occurs with any print book. In particular, this appendix does the following:

- Mentions technical items that might not have been mentioned elsewhere in the book
- Covers new topics if AWS adds new content to the exam over time
- Provides a way to get up-to-the-minute current information about content for the exam

Always Get the Latest at the Book's Product Page

You are reading the version of this appendix that was available when your book was printed. However, given that the main purpose of this appendix is to be a living, changing document, it is important that you look for the latest version online at the book's companion website. To do so, follow these steps:

Step 1. Browse to

<https://www.pearsonITcertification.com/title/9780137941582>.

Step 2. Click the Updates tab.

Step 3. If there is a new Appendix B document on the page, download the latest Appendix B document.

Note

The downloaded document has a version number.

Comparing the version of the print [Appendix B](#) (Version 1.0) with the latest online version of this appendix, you should do the following:

- **Same version:** Ignore the PDF that you downloaded from the companion website.
 - **Website has a later version:** Ignore this [Appendix B](#) in your book and read only the latest version that you downloaded from the companion website.
-

Technical Content

The current Version 1.0 of this appendix does not contain additional technical coverage.

Glossary of Key Terms

A

access key A special set of keys linked to a specific AWS IAM user.

ACID The storage consistency of a relational database, based on atomicity, consistency, isolation, and durability.

active-active Multi-region active-active deployment of resources across multiple regions for workloads requiring high availability and failover.

alarm A warning issued when a single metric crosses a set threshold over a defined number of time periods.

Amazon CloudFront The AWS content delivery network (CDN) hosted in all edge locations.

Amazon Elastic Block Storage (EBS) A virtual hard disk block storage device that is attached to Amazon EC2 instances.

Amazon Elastic Compute Cloud (EC2) A web service that provides secure, resizable compute capacity in the cloud. It enables you to launch and manage virtual servers, called

Amazon Elastic Compute Cloud (EC2) instances, in the AWS cloud.

Amazon ElastiCache A distributed in-memory data store.

Amazon Machine Image (AMI) A template of an instance's root drive.

application programming interface (API) A defined set of protocols that enables applications and services to communicate with each other.

archive An Amazon S3 Glacier grouping of compressed and encrypted files.

asymmetric key One key of a public/private key pair.

Auto Scaling An AWS service that adjusts compute capacity to maintain desired performance.

Auto Scaling group A group of Amazon EC2 instances that is controlled (that is, scaled up, scaled down, or maintained) using the EC2 Auto Scaling service.

availability zone (AZ) An insulated separate location within a region that contains at least one data center.

AWS Artifact Allows AWS customers to review the compliance standards supported by AWS.

AWS Direct Connect A dedicated private fiber connection to AWS VPCs or AWS public services.

AWS EC2access control list (ACL) A list that enables you to control access to Amazon S3 buckets by granting read/write permissions to other AWS accounts.

AWS Identity and Access Management (IAM) The hosted security system for the AWS cloud that controls access to AWS resources.

AWS Key Management Service (KMS) An AWS service that centrally manages AWS customers' cryptographic keys and policies across AWS services that require data encryption.

AWS well-architected framework A framework for designing, deploying, and operating workloads hosted at AWS.

B

block storage Data records stored in blocks on a storage area network.

bucket The storage unit for an Amazon S3 object.

bucket policy A resource policy that is assigned directly to a storage entity such as an Amazon S3 bucket.

burst capacity The ability of a storage unit or a compute instance to increase processing power for a short period of time.

burst credits Performance credits that make it possible to burst above a defined performance baseline.

C

capacity units A measure of Amazon DynamoDB performance in terms of either reading or writing.

certificate authority (CA) A company or an entity that validates the identities of websites or domains using cryptographic public/private keys.

CloudWatch log group A group that logs information in near real time.

codebase The body of source code for a software program or application.

cold storage Infrequently accessed storage.

condition Special rule in a permission policy.

connection draining The process of deregistering (removing) a registered instance from a load balancer target group.

cooldown period A defined time period when no changes are allowed.

cost allocation tags Tags that are used to categorize and track AWS costs displayed with monthly and hourly cost allocation reports.

Cost and Usage Report (CUR) Tracks your AWS usage and provides estimated charges associated with your account for the current month.

D

data consistency A definition of how data records are either the same or not the same due to replication.

data transfer Incoming (ingress) and outgoing (egress) packet flow.

defense in depth (DiD) Deployment of multiple security controls (physical, administrative, and technical) to protect a hosted workload.

dependencies Cloud services, applications, servers, and various technology components that depend upon each other when providing a business solution.

Direct Connect *See AWS Direct Connect.*

distributed session A user session for which user state information is held in a separate durable storage location.

E

EC2 *See Amazon Elastic Compute Cloud (EC2).*

egress-only Internet gateway (EOIG) A one-way gateway connection for EC2 instances with IPv6 addresses.

Elastic Block Storage (EBS) *See Amazon Elastic Block Storage (EBS).*

Elastic IP (EIP) address A static public IP address that is created and assigned to your AWS account.

ElastiCache *See Amazon ElastiCache.*

endpoint A location where communication is made; a private connection from a VPC to AWS services.

ephemeral storage Temporary local block storage.

event notification Communications about changes in the application stack.

externally authenticated user A user that has authenticated outside Amazon before requesting access to AWS resources.

F–H

FedRAMP Federal Risk and Authorization Management Program, establishes the security requirements for usage of cloud services for federal government agencies.

health check A status check for availability.

high availability A group of compute resources that continue functioning even when some of the components fail.

I–K

IAM group A group of AWS IAM users.

IAM role A permission policy that provides temporary access to AWS resources.

Identity and Access Management (IAM) *See AWS Identity and Access Management (IAM).*

immutable During deployment and updates components are replaced rather than changed.

input/output operations per second (IOPS) A performance specification that defines the rate of input and output per second when storing and retrieving data.

Internet gateway (IG) An AWS connection to the Internet for a virtual private cloud (VPC).

Key Management Service (KMS) *See AWS Key Management Service (KMS).*

key-value An item of data where the key is the name and the value is the data.

L

Lambda@Edge A custom-created function to control ingress and egress Amazon CloudFront traffic.

launch template A set of detailed EC2 instance installation and configuration instructions.

LCU See load balancer capacity unit (LCU).

lifecycle hook A custom action to be performed before or after an Amazon EC2 instance is added to or removed from an Auto Scaling Group.

lifecycle policy A set of rules for controlling the movement of Amazon S3 objects between S3 storage classes.

lifecycle rules Rules that allow customers to transition backups that are stored in warm storage to cheaper cold storage.

listener A load balancer process that checks for connection requests using the defined protocols and ports.

load balancer capacity unit (LCU) Defines the maximum resource consumed calculated on new connections, active, connections, bandwidth, and rule evaluations.

Local Zone A single deployment of compute, storage, and select services close to a large population center.

M

metric Data collected for an AWS CloudWatch variable.

mount point A logical connection to a directory in a file system; a method to attach Amazon EFS storage to a Linux workload.

multi-factor authentication (MFA) Authentication that involves multiple factors, such as something you have and something you know.

multipart upload An upload in which multiple parts of a file are synchronously uploaded.

N

NAT gateway service A service that provides indirect Internet access to Amazon EC2 instances that are located on private subnets.

network access control list (NACL) A stateless subnet firewall that protects both inbound and outbound subnet traffic.

Nitro The latest AWS hypervisor, which replaces the Xen hypervisor and provides faster networking, compute, encryption, and management services.

NoSQL A database that does not follow SQL rules and architecture, hence the name “no” SQL.

NVMe Non-Volatile Memory Express, a standard hardware interface for SSD drives connected using PCI Express bus.

O

object storage Data storage as a distinct object with associated metadata containing relevant information.

origin access identity (OAI) A special AWS IAM user account that is provided the permission to access the files in an Amazon S3 bucket.

origin failover An alternate data source location for Amazon CloudFront distributions.

P

password policy A policy containing global password settings for AWS account IAM users.

peering connection A private networking connection between two VPCs or two transit gateways.

Pilot light An active/passive disaster recovery design that involves maintaining a limited set of compute and data records to be used in case of a disaster to the primary application

resources. The compute records are turned off until needed, but the data records are active and are kept up-to-date.

primary database The primary copy of database records.

Q–R

queue A redundant storage location for messages and application state data for processing.

read capacity unit One strongly consistent read per second, or two eventually consistent reads per second, for items up to 4 KB in size.

read replica A read-only copy of a linked primary database.

recovery point objective (RPO) A metric that specifies the acceptable amount of data that can be lost within a specified period.

recovery time objective (RTO) A metric that specifies the maximum length of time that a service can be down after a failure has occurred.

region A set of AWS cloud resources in a geographic area of the world.

regional edge cache A large throughput cache found at an edge location that provides extra cache storage.

regional endpoint A device that provides HTTPS access to AWS services within a defined AWS region.

reliability The reasonable expectation that an application or service is available and performs as expected.

Reserved instance An Amazon EC2 instance for which you have prepaid.

RPO *See* recovery point objective (RPO).

RTO *See* recovery time objective (RTO).

S

scale out To increase compute power automatically.

scaling policy A policy that describes the type of scaling of compute resources to be performed.

security group A stateful firewall protecting Amazon EC2 instances' network traffic.

Server Message Block (SMB) A network protocol used by Windows systems on the same network to store files.

serverless A type of computing in which compute servers and integrated services are fully managed by AWS.

server-side encryption (SSE) Encryption of data records at rest by an application or a service.

service-level agreement (SLA) A commitment between a cloud service provider and a customer indicating the minimum level of service to be maintained.

service-level indicator (SLI) Indicates the quality of service an end user is receiving at a given time. SLIs are measured as a level of performance.

service-level objective (SLO) An agreement defined as part of each service-level agreement. Objectives could be uptime or response time.

service quota A defined limit for AWS services created for AWS accounts.

simple scaling Scaling instances up or down based on a single AWS CloudWatch metric.

SLA See service-level agreement (SLA).

snapshot A point-in-time incremental backup of an EBS volume.

Snow device A variety of network-attached storage devices that can be used to transfer and receive data records to and from Amazon S3 storage.

standby database A synchronized copy of a primary database that is available in the event of a failure.

stateful Refers to a service that requires knowledge of all internal functions.

stateless Refers to a self-contained redundant service that has no knowledge of its place in the application stack.

step scaling Scaling up or down by percentages.

sticky session A user session for which communication is maintained with the initial application server for the length of the session. It ensures that a client is bound to an individual backend instance.

Structured Query Language (SQL) The de facto programming language used in relational databases.

subnet A defined IP address range hosted within a VPC.

symmetric key A key that can both lock and unlock.

T

T instance An instance provided with a baseline of compute performance.

table A virtual structure in which Amazon DynamoDB stores items and attributes.

target group A group of registered instances that receives specific traffic from a load balancer.

task definition A blueprint that describes how a Docker container should launch.

Throughput Optimized An EBS hard disk drive (HDD) volume option that provides sustained throughput of 500 Mb/s.

tiered pricing The more you use the less you are charged.

time to live (TTL) A value that determines the storage time of an Amazon CloudFront cache object.

U–V

uptime the percentage of time that a website is able to function during the course of a calendar year.

user state Data that identifies an end user and the established session between the end user and a hosted application.

versioning A process in which multiple copies of Amazon S3 objects, including the original object, are saved.

virtual private cloud (VPC) A logically isolated virtual network in the AWS cloud.

virtual private gateway (VPG) The AWS side of a VPN connection to a VPC.

W–Z

warm standby An active/passive disaster recovery design that maintains a limited set of compute and data records that are both on and functioning. When the primary application resources fail, the warm standby resources are resized to production values.

write capacity unit (WCU) One write per second for items up to 1 KB in size.

write-once/read-many (WORM) A security policy that can be deployed on an Amazon S3 bucket or in S3 Glacier storage. The policy indicates that the contents can be read many times but are restricted from any further writes once the policy is enacted.

zonal Refers to an availability zone location.

Index

A

- Access Advisor, [131](#)
- access key/s, [82](#), [751](#)
 - IAM user, [92–94](#)
 - rotating, [97–99](#)
- access logs, [553](#)
- access point, S3 (Simple Storage Service), [401–402](#)
- account
 - access, [124–126](#)
 - IAM (Identity and Access Management), [95–96](#)
 - user, [88](#). *See also* [user](#)
- ACID (atomicity, consistency, isolation, durability), [509](#), [751](#)
- ACL (access control list), [23](#), [113](#), [752](#)
- actions, [87–88](#), [109–110](#), [546–547](#)
- active-active failover, [340–343](#), [751](#)
- adaptive capacity, [506–507](#)
- AES (Advanced Encryption Standard), [21](#)
- Agile, [62](#), [267](#)
- alarm, [460–461](#), [751](#)
- ALB (Application Load Balancer), [541–543](#)
 - access logs, [553](#)
 - cheat sheet, [553](#)

health checks, [548–550](#)
listeners and routing, [543–545](#)
rules, conditions, and actions, [545–547](#)
sticky session support, [551–552](#)
target group attributes, [550–551](#)
target groups, [547–548](#)
alias records, [352](#)

Amazon

- A2C (App2 Container), [246](#)
- Amazon EventBridge, [256–258](#)
- API Gateway, [258–259](#)
 - cheat sheet, [261–262](#)
 - choosing the API protocol to use, [260–261](#)
 - communication options, [259–260](#)
 - selecting an authorizer, [261](#)
- Aurora, [340](#), [493–495](#)
 - cheat sheet, [500–501](#)
 - communicating with, [499–500](#)
 - deployment options, [494–496](#)
 - replication, [498–499](#)
 - serverless, [674–675](#)
 - storage, [496–498](#)
- CloudFront, [151](#), [238–239](#), [527](#), [751](#)
 - cheat sheet, [536](#)

edge functions, [534–536](#)
how it works, [527–528](#)
HTTPS access, [529–530](#)
origin failover, [532–533](#)
regional edge caches, [528–529](#)
restricting distribution of content, [532](#)
serving private content, [530–532](#)
use cases, [529](#)
video-on-demand and live streaming, [533–534](#)

Cognito, [176–177](#)

federated identity pool, [179–180](#)
user pool, [177–179](#)

DynamoDB, [238](#), [299](#), [501–503](#)

Accelerator, [511](#)
ACID, [509](#)
adaptive capacity, [506–507](#)
backup and restore, [511–512](#)
cheat sheet, [512](#)
data consistency, [507–509](#)
data transfer costs, [683–685](#)
global tables, [510–511](#)
provisioning table capacity, [504–506](#)
tables, [503–504](#)

EBS (Elastic Block Storage), [212](#), [753](#)

enabling, [212–213](#)
enabling for each AWS region, [215](#)
key rotation, [213–214](#)
select KMS key, [214–215](#)

EC2 Image Builder, [435–436](#)

ECS (Elastic Container Service), [244](#), [443](#)
task definition choices, [443–446](#)

EFS (Elastic File System), [379–380](#)
cheat sheet, [383–384](#)
lifecycle management, [383](#)
performance modes, [380–381](#)
security, [382](#)
storage classes, [382](#)
throughput modes, [381](#)

EKS (Elastic Kubernetes Service), [244](#), [446–447](#)

ElastiCache, [512–513](#), [751](#)
for Memcached, [513–514](#)
for Redis, [514–517](#)

ELB (Elastic Load Balancer), [539](#)
access logs, [553](#)
application load balancer deployment, [541–545](#)
costs, [695–696](#)
features, [540–541](#)
health checks, [548–550](#)

rules, conditions, and actions, [545–547](#)
sticky session support, [551–552](#)
target group, [547–548](#)
target group attributes, [550–551](#)
FSx for Windows File Server, [386–388](#)
Global Accelerator, [536–538](#)
Glue, [413](#)
 components, [413–414](#)
 ETL job flow, [414](#)
GuardDuty, [187–188](#)
 cheat sheet, [189](#)
 types of security analysis, [188–189](#)
Inspector, [195–196](#)
Kinesis, [417](#)
Lambda, [436–438](#)
 cheat sheet, [441](#)
 integration, [438–439](#)
 settings, [439–440](#)
Macie, [189–191](#), [219](#)
RDS (Relational Database Service). *See* [RDS \(Relational Database Service\)](#)
Redshift, [517–520](#), [685–686](#)
Route [53](#), [59](#), [150](#), [348–349](#)
 alias records, [352](#)

health checks, [349–350](#)

resolver, [352–353](#)

routing policies, [350](#)

traffic flow policies, [351](#)

S3 (Simple Storage Service), [9](#), [216](#), [388–390](#)

access points, [401–402](#)

bucket concepts, [390–393](#)

bucket policy, [217–218](#)

cheat sheet, [403–404](#)

data consistency, [393](#)

Glacier storage at rest, [222–223](#)

management, [396–400](#)

multi-region access points, [402](#)

object lock policies, [221–222](#)

permission settings, [216–217](#)

preselected URLs for objects, [403](#)

presigned URL, [218–219](#)

storage at rest, [220–221](#)

storage classes, [394–396](#)

S3 Glacier, [404–405](#)

cheat sheet, [406](#)

Deep Archive, [406](#)

retrieval policies, [405–406](#)

vault, [405](#)

SNS (Simple Notification Service), [248–249](#)
 cheat sheet, [250](#)
 creating a notification topic, [250](#)
 publisher and subscriber options, [249–250](#)

SQS (Simple Queue Service), [250–254](#)

VPC (Virtual Private Cloud), [15](#)

AMI (Amazon Machine Image), [428–430](#)
 AWS Marketplace, [431–432](#)
 build considerations, [434–435](#)
 choosing, [430–431](#)
 custom, [432–434](#)
 custom instance store, [434](#)
 golden pipeline, [436](#)
 prebuilt, [430](#)
 Windows, [431](#)

analytical tools, [412–413](#), [415–416](#)

API (application programming interface), [18](#), [751](#)

API Gateway, [258–259](#)
 cheat sheet, [261–262](#)
 communication options, [259–261](#)
 selecting an authorizer, [261](#)

application/s
 dependency, [27–28](#)
 deprecation, [28](#)

integration services, [247–248](#)

- Amazon EventBridge, [256–258](#)
- Amazon SNS (Simple Notification Service), [248–250](#)
- Amazon SQS (Simple Queue Service), [250–254](#)
- AWS Step Functions, [254–256](#)

load balancer, [23, 58](#). *See also* [load balancer](#)

migrating, [24](#)

- allow access to on-premises data records, [26](#)
- define a value proposition, [24–25](#)
- lift and shift, [26–27](#)
- solve a single problem, [26](#)
- start with low value/low risk, [25–26](#)

replacing, [28](#)

security, [23–24, 46](#)

stateful, [239](#)

stateless, [239–243](#)

archive, [405, 751](#)

ASG (Auto Scaling group), [465–466](#)

- lifecycle hooks, [472–473](#)
- management options, [470–471](#)
- scaling, [466–470](#)

asymmetric key, [751](#)

authentication

- Amazon Cognito

federated identity pool, [179–180](#)
user pool, [177–179](#)
external, [81](#)
IAM (Identity and Access Management), [82–84](#)
multifactor, [80–81](#), [99](#)
authorization, IAM (Identity and Access Management), [85–87](#)
auto scaling, [461–463](#), [751](#)
AWS, [473–474](#)
cheat sheet, [473](#)
EC2 (Elastic Compute Cloud), [463](#)
 ASG (Auto Scaling group), [465–471](#)
 launch configuration, [464](#)
 launch template, [464](#)
 lifecycle hooks, [472–473](#)
 termination policy, [471–472](#)
automatic failover, [60](#)
automation, [16](#)
 cooldown period, [471](#)
 Elastic Beanstalk, [279–281](#)
 modifying the capacity of the application
 infrastructure, [281](#)
 updating applications, [282–283](#)
 Service Catalog, [277–279](#)
 tools, [266–277](#)

availability, [293–295](#). *See also* [high availability](#); [reliability](#)
outages and, [306](#)
workload, [48](#)

availability zone, [300–301](#), [752](#)
distribution, [301–303](#)

RDS (Relational Database Service), [488](#)
storage, [329](#)

AWS. *See also* [cloud computing](#); [Well-Architected Framework](#)
analytical services, [415–416](#)
Application Discovery Service, [26](#)
Application Migration Service, [26](#)
Architecture Center, [6](#)
Artifact, [752](#)
Artifact utility, [311](#)
auto scaling, [473–474](#)
availability zone, [300–303](#)
Backup, [337](#), [618–619](#)
 cheat sheet, [620–621](#)
 lifecycle rules, [619–620](#)
Budgets, [607–609](#)
CDN, placement, [56–57](#)
Certificate Manager, [227–228](#)
cloud provider responsibilities, [20–21](#)
cloud services, [15–16](#)

Cloud9, [24–25](#)

CloudFormation, [16](#), [268–269](#)

- components, [269](#)
- creating an EC2 instance, [273–274](#)
- stack sets, [276–277](#)
- stacks, [272–273](#)
- templates, [270–272](#)
- updating with change sets, [275](#)

CloudHSM, [227](#)

CloudTrail, [16](#), [191–192](#)

- cheat sheet, [194](#)
- creating a custom trail, [192–194](#)

CloudWatch, [16](#)

CodeCommit, [64](#)

Cognito, [83](#)

compute, [15](#), [55–56](#). *See also* [compute](#)

Config, [199–200](#), [600–602](#)

Control Tower, [138–139](#)

Cost and Usage Reports, [609–610](#)

costs, calculating, [597–598](#)

data backup and replication, [223–224](#)

Data Lake, [407–409](#), [412–413](#)

data replication, placement, [57–58](#)

database/s, [481](#)

DataSync, [384–385](#)

Direct Connect, [149](#), [185–186](#), [752](#)

- cheat sheet, [187](#)
- gateway, [186–187](#)

edge locations. *See also* [edge locations](#)

- AWS Shield, [151–152](#)
- network services, [150–151](#)

Elastic Beanstalk, [18](#), [67](#)

- essential characteristics, [6–8](#)
- failover, architecture, [60](#)

GovCloud, [317](#), [318–319](#)

IAS (Identity and Access Management), [14](#)

Identity Center, [132–133](#)

infrastructure, [16](#). *See also* [infrastructure](#)

KMS (Key Management Service), [224](#)

- cheat sheet, [226–227](#)
- console, [224–225](#)
- envelope encryption, [225–226](#)

Lake Formation, [409–411](#)

Lambda, [238](#)

- load balancer, placement, [57–58](#)
- managed service, [19–20](#), [293](#), [308–310](#)
- management console, [9](#)

Marketplace, [431–432](#)

NIST compliance, [316–317](#)
operational benefits, [19–20](#)
Organizations, [134–136](#)
outages, [306](#)
Outposts, [7](#), [357](#)
PaaS (platform as a service), [17–18](#)
RDS (Relational Database Service), [481–482](#)
 best practices, [491](#)
 cheat sheet, [493](#)
 engines, [482](#)
 failover, [487–488](#)
 high-availability design, [485–488](#)
 installation, [488–490](#)
 instance class types, [485](#)
 instances, [483–484](#)
 Multi-AZ deployment, [488](#)
 performance monitoring, [490–491](#)
 Proxy, [492–493](#)
 standby, [487](#)
region, selection criteria, [310](#). *See also* [region/s](#)
 compliance rules, [311–319](#)
 latency concerns, [319–320](#)
 pricing, [321](#)
 services, [320](#)

regulatory compliance
rules, [311–314](#)
standard/s, [315](#)

Resource Access Manager, [136–138](#)

Schema Conversion tool, [681](#)

Secrets Manager, [194–195](#)

security, [21](#)
application, [23–24](#)
data, [21–22](#)
network, [22–23](#)

self-service, [9](#)

servers, [19](#)

Service Catalog, [277–279](#)

services, cheat sheet, [31–36](#). *See also* [service/s](#)

Shield, [150](#), [151–152](#)

SLA (service-level agreement), [47–48](#)

SP 800–145, “The NIST Definition of Cloud Computing”, [8–9](#)
broad network access, [10–11](#)
on-demand self-service, [9](#)
measured service, [12–13](#)
rapid elasticity, [11–12](#)
resource pooling, [10](#)

stateless processes, [68](#)

Step Functions, [254–256](#)
storage, [19](#), [362](#), [363–365](#). *See also* [storage](#)
Storage Gateway, [625–627](#)
STS (Security Token Service), [120](#), [126–128](#)
tiered pricing, [599–600](#)
Trusted Advisor, [196–198](#)
uptime, [331](#)
user, [88–90](#)
VMWare, [16](#)
VPN (virtual private network)
 route propagation, [184–185](#)
 solutions, [183–184](#)
WAF (Web Application Firewall), [151](#)
Well-Architected Framework, [4](#), [28–30](#), [39–40](#), [752](#)
 cost optimization, [51](#)
 operational excellence, [44–45](#)
 performance efficiency, [49–51](#)
 reliability, [47–49](#)
 security, [45–47](#)
 sustainability, [51–52](#)
AZ (availability zone), [155](#)

B

BAA (Business Associate Addendum), [316](#)

backing services, [66–67](#)

backup and restore, [223–224](#), [332–333](#)

- Amazon DynamoDB, [511–512](#)
- AWS Backup, [618–619](#)
- cheat sheet, [620–621](#)
- lifecycle rules, [619–620](#)

database retention policies, [687–689](#)

- fast snapshot, [374–375](#)
- snapshot, [295](#), [362](#)

warm standby, [337–339](#)

- Amazon Aurora, [340](#)
- multi-region, [339](#)

bastion host, [164–165](#)

best practices

- IAM (Identity and Access Management), [128–130](#)
- RDS (Relational Database Service), [491](#)

Big Bang, [62](#)

billing. *See also* [cost/s](#); [pricing](#)

- measured service, [12–13](#)
- traffic, [578–579](#)

block storage, [362](#), [752](#)

born-in-the-cloud mentality, [14](#)

broad network access, [10–11](#)

bucket, [752](#)

policy, [217–218](#)
S3 (Simple Storage Service), [390–393](#)
versioning, [400–401](#)
budget, [607–609](#)
build stage, Elastic Beanstalk, [67](#)
building, serverless web app, [262](#)
 create a static website, [263](#)
 create the backend components, [264](#)
 register for the conference, [266](#)
 set up the API gateway, [265](#)
 user authentication, [263–264](#)
burst capacity, [752](#)
burst credit, [369–370](#), [752](#)
bursting mode, EFS, [381](#)
business continuity, [60](#)
BYOIP (Bring Your Own IP), [579–580](#)

C

CA (certificate authority), [752](#)
canary deployment, [327](#)
capacity units, [752](#)
CDN
 placement, [56–57](#)
 POP (point of presence), [56](#)

change sets, [275](#)

cheat sheet

ALB (Application Load Balancer), [553](#)

Amazon API Gateway, [261–262](#)

Amazon Aurora, [500–501](#)

Amazon DynamoDB, [512](#)

Amazon EFS (Elastic File System), [383–384](#)

Amazon ElastiCache, [514](#), [516](#)

Amazon Macie, [190–191](#)

Amazon Redshift, [519–520](#)

Amazon S3 Glacier, [406](#)

Amazon SNS (Simple Notification Service), [250](#)

Amazon SQS (Simple Queue Service), [253–254](#), [403–404](#)

auto scaling, [473](#)

AWS Backup, [620–621](#)

AWS CloudTrail, [192–194](#)

AWS Lambda, [441](#)

AWS Storage Gateway, [625–627](#)

CloudFront, [536](#)

CloudWatch, [461](#)

cost management, [610–611](#)

data transfer costs, [686–687](#), [716–717](#)

dedicated host, [637](#)

disaster recovery, [344–345](#)

EBS (Elastic Block Store), [372–373](#)
FSx for Windows File Server, [388](#)
IAM (Identity and Access Management), [132](#)
IP address, [577–578](#)
KMS (Key Management Service), [226–227](#)
NACL (network access control list), [169–170](#)
NAT (network address translation), [176](#)
NoSQL costs, [676–680](#)
RDS (Relational Database Service), [493](#), [671](#)
route table, [158](#)
service quota, [347–348](#)
SG (security group), [161–162](#)
single and multi-region recovery, [343–344](#)
snapshot, [376–377](#)
subnet, [572–573](#)
VPC (Virtual Private Cloud), [560–561](#)

CIDR block
primary, [566–568](#)
secondary, [568–569](#)

Cloud CoE (Cloud Center of Excellence), [44](#)
cloud computing. *See also* [Well-Architected Framework](#)
availability, [293–295](#)
AWS, essential characteristics, [6–8](#)
failover, architecture, [60](#)

IaaS (infrastructure as a service), [14–16](#)
load balancer, placement, [57–58](#). *See also* [load balancer](#)
providers, [39](#)
public cloud, [6–7](#)
reliability, [295–296](#)

SaaS (software as a service), [13](#)
service/s. *See also* [service/s](#)
CDN, [56–57](#)
costs, [598–599](#)
data replication, [57–58](#)
data residency and compute locations, [55–56](#)
placing, [55](#)
shared responsibility model, [79–80](#)

SP 800–145, “The NIST Definition of Cloud Computing”
broad network access, [10–11](#)
on-demand self-service, [9](#)
measured service, [12–13](#)
rapid elasticity, [11–12](#)
resource pooling, [10](#)

Cloud Foundry, [17](#)

Cloud9, [24–25](#)

CloudFormation, [16](#), [268–269](#)
components, [269](#)
creating an EC2 instance, [273–274](#)

stack sets, [276–277](#)
stacks, [272–273](#)
templates, [270–272](#)
updating with change sets, [275](#)

CloudFront, [151](#), [527](#)
 cheat sheet, [536](#)
 costs, [698–701](#)
 edge functions, [534–536](#)
 how it works, [527–528](#)
 origin failover, [532–533](#)
 regional edge caches, [528–529](#)
 restricting distribution of content, [532](#)
 serving private content
 HTTPS access, [530](#)
 using an origin access identity, [531–532](#)
 using signed URLs, [530–531](#)
 use cases, [529–530](#)
 video-on-demand and live streaming, [533–534](#)

CloudTrail, [16](#)

CloudWatch, [16](#), [53](#), [421](#), [447–448](#)
 alarm and action settings, [460–461](#)
 basic monitoring, [448–449](#)
 cheat sheet, [461](#)
 collecting data, [451–452](#)

creating an alarm, [459–460](#)
integration, [453–455](#)
log group, [752](#)
logs, [449–451](#)
metrics, [555–556](#)
planning for monitoring, [452–453](#)
terminology, [455–459](#)

code repository, [63](#)
codebase, [63–64](#), [752](#)
CodeCommit, [64](#)
cold storage, [753](#)
command/s
 create-policy, [105](#)
 iostat, [370](#)
 list-policies, [105](#)

compliance
 NIST, [316–317](#)
 regulatory. *See also* [regulatory compliance](#)

components
 Amazon SQS (Simple Queue Service), [251–253](#)
 AWS CloudFormation, [269](#)
 Composer, [65](#)
 compute, [15](#), [425–427](#). *See also* [EC2 \(Elastic Compute Cloud\)](#)
 Amazon Lambda, [436–438](#)

EC2 (Elastic Compute Cloud), [427–428](#)

- AMI (Amazon Machine Image), [428–435](#)
- dedicated host, [636–637](#)
- dedicated instance, [638](#)
- on-demand instance service quotas, [641–643](#)
- on-demand pricing, [640–641](#)
- Fleet, [655](#)
- Image Builder, [435–436](#)
- instance choices, [634–636](#)
- instance purchasing options, [638–640](#)
- instance types, [633](#)
- placement groups, [638](#)
- pricing, [655](#)
- Reserved instance, [644–647](#)
- Savings Plans, [649–650](#)
- vCPU, [634](#)
- matching utilization with requirements, [659–660](#)
- optimizing, [656–659](#)
- scaling, [661](#)
- selecting a location, [55–56](#)
- Spot Fleet, [651–653](#)
- spot instance, [650–651](#)
- tools and utilities, [655–656](#)

conditional policy, [86](#), [116](#)

configuration files, [66](#)
connection draining, [753](#)
connection tracking, [161](#)
connectivity options, VPC, [583](#)
containers and container management, [441–443](#)
 Amazon ECS (Elastic Container Service), [443–446](#)
 Amazon EKS (Elastic Kubernetes Service), [446–447](#)
 migrating applications to, [246](#)
 orchestration, [244–245](#)
Control Tower, [138–139](#)
controlled storage, [373](#)
controls
 detective, [210–212](#)
 IAM (Identity and Access Management), [210](#)
cooldown period, [471](#), [753](#)
corporate mindset, [13](#)
Cost Explorer, [604–607](#)
cost/s
 allocation tags, [612–613](#), [753](#)
 AWS, [321](#), [597–598](#)
 cheat sheet, [610–611](#)
 cloud service, [598–599](#)
 CloudFront, [698–701](#)
 data transfer, [681–682](#), [706–707](#)

accessing AWS services in different regions, [710–713](#)
accessing AWS services in the same region, [707–709](#)
cheat sheet, [686–687](#)
DocumentDB, [686–687](#)
DynamoDB, [683–685](#)
edge locations, [713](#)
network, [714](#)
public versus private traffic charges, [714](#)
RDS, [682–683](#)
Redshift, [685–686](#)
workload components in the same region, [709–710](#)
ELB (Elastic Load Balancer), [695–696](#)
management tools, [602–604](#)
NAT (network address translation), [696–697](#)
network services from on-premises locations, [703–705](#)
optimization, [51](#)
protection, [152](#)
reliability, [295](#)
storage, [613–617](#)
createdBy tag, [612](#)
create-policy command, [105](#)
creating
CloudTrail trail, [192–194](#)
IAM policy, [105–106](#)

IAM user, [91–92](#)
VPC (Virtual Private Cloud), [561–564](#)
Credential Report, [130](#)
CRM (customer relationship management), [156](#)
cross-account access, [124–126](#)
CUR (Cost and Usage Report), [753](#)
custom AMI, [432–434](#)
custom policy, [102](#)
custom route table, [155–158](#)
custom SG (security group), [162–163](#)
customer gateway, [182–183](#)

D

dashboard, IAM, [79](#)
data
access, governance, [207](#)
classification, [207–209](#)
consistency, [507–509](#), [753](#)
lake, [407–409](#)
replication, [57–58](#), [223–224](#)
security, [21–22](#)
stateful, [239–243](#)
stateless, [239–243](#)
storage. *See* [storage](#)

structured, [411–412](#)
transfer, [621–625](#), [753](#)
 accessing AWS services in different regions, [710–713](#)
 accessing AWS services in the same region, [707–709](#)
 costs, [706–707](#)
 costs cheat sheet, [716–717](#)
 edge locations, [713](#)
 public versus private traffic charges, [714](#)
 workload components in the same region, [709–710](#)
unstructured, [412](#)

database/s, [299](#)
 Amazon Aurora, [493–495](#)
 cheat sheet, [500–501](#)
 communicating with, [499–500](#)
 deployment options, [494–496](#)
 replication, [498–499](#)
 serverless, [674–675](#)
 storage, [496–498](#)
 Amazon DynamoDB, [501–503](#)
 Accelerator, [511](#)
 ACID, [509](#)
 adaptive capacity, [506–507](#)
 backup and restore, [511–512](#)
 cheat sheet, [512](#)

data consistency, [507–509](#)
global tables, [510–511](#)
provisioning table capacity, [504–506](#)
tables, [503–504](#)

Amazon Redshift, [517–520](#)

AWS, [15](#), [481](#)

data transfer costs, [681–682](#)
DocumentDB, [686–687](#)
DynamoDB, [683–685](#)
RDS, [682–683](#)
Redshift, [685–686](#)

design choices, [668](#)

migration, [680–681](#)

NoSQL, [675–677](#)
costs cheat sheet, [676–680](#)
service comparisons, [676–677](#)

RDS (Relational Database Service), [481–482](#), [668–670](#)
best practices, [491](#)
cheat sheet, [493](#)
costs cheat sheet, [671](#)
design solutions, [672–675](#)
engines, [482](#)
failover, [487–488](#)
high-availability design, [485–488](#)

installation, [488–490](#)
instance class types, [485](#)
instances, [483–484](#)
Multi-AZ deployment, [488](#)
performance monitoring, [490–491](#)
Proxy, [492–493](#)
read replica, [673](#)
standby, [487](#)
retention policies, [687–689](#)
schema conversion, [681](#)
SQL (Structured Query Language), [503](#)
declare and isolate dependencies, [65](#)
dedicated host, [636–637](#)
dedicated instance, [638](#)
default VPC, [569–570](#)
Defense in Depth, [45–47, 753](#)
on-demand instance service quotas, [641–643](#)
on-demand self-service, [9](#)
dependency/ies, [753](#)
application, [27–28](#)
declare and isolate, [65](#)
infrastructure-level, [63–64](#)
manager, [65](#)
workload, [48, 54](#)

deployment

- Amazon Aurora, [494–496](#)
- canary, [327](#)
- Multi-AZ, [488](#)
- pilot light, [333–337](#)
- detective controls, [210–212](#)
- development, [70](#)
 - Agile, [61–62](#)
 - Big Bang, [62](#)
 - frameworks, [66](#)
 - Waterfall, [61–62](#)
- DevOps, [267](#)
- Direct Connect, [149](#), [185–186](#), [753](#)
 - cheat sheet, [187](#)
 - gateway, [186–187](#)
- disaster recovery, [54](#)
- distributed design, [321–322](#)
 - high availability and fault tolerance, [322–325](#)
 - removing single points of failure, [325–327](#)
- distributed session management, [243–247](#), [753](#)
- DocumentDB, data transfer costs, [686–687](#)
- DR (disaster recovery), [330](#), [331](#)
 - backup and restore, [332–333](#)
 - cheat sheet, [344–345](#)

pilot light, [333–337](#)
warm standby, [337–339](#)

E

EBS (Elastic Block Storage), [50](#), [365–366](#), [751](#), [753](#)
attaching a volume, [371–372](#)
cheat sheet, [372–373](#)
encryption, [212](#), [294–295](#)
enabling, [212–213](#)
enabling for each AWS region, [215](#)
key rotation, [213–214](#)
select KMS key, [214–215](#)
multi-attach, [366](#)
recycle bin, [376](#)
snapshot, [373](#)
administration, [375–376](#)
cheat sheet, [376–377](#)
fast restore, [374–375](#)
taking from a Linux instance, [373–374](#)
taking from a Windows instance, [374](#)
volume types, [367–369](#)
elastic EBS, [370–371](#)
General Purpose SSD, [369–370](#)

EC2 (Elastic Compute Cloud), [80](#), [751](#). *See also* [Reserved instance](#)

access to AWS resources, [119–121](#)

auto scaling

ASG (Auto Scaling group), [465–471](#)

launch configuration, [464](#)

launch template, [464](#)

bastion host, [164–165](#)

dedicated host, [636–637](#)

on-demand pricing, [640–641](#)

Fleet, [655](#)

Image Builder, [435–436](#)

immutable infrastructure, [327](#)

instance

choices, [634–636](#)

dedicated, [638](#)

on-demand service quotas, [641–643](#)

purchasing options, [638–640](#)

Reserved, [644–647](#)

storage volume, [377–378](#)

types, [633](#)

placement groups, [638](#)

pricing, [655](#)

Savings Plans, [649–650](#)

spot capacity pool, [653–655](#)
task definition choices, [443–446](#)
vCPU, [634](#)

ECS (Elastic Container Service), [443](#)
edge locations, [303](#)
AWS Shield, [151–152](#)
data transfer costs, [713](#)
scalable delivery, [238–239](#)
WAF (Web Application Firewall), [152–153, 154–167](#)

EFS (Elastic File System), [379–380](#)
cheat sheet, [383–384](#)
lifecycle management, [383](#)
performance modes, [380–381](#)
security, [382](#)
storage classes, [382](#)
throughput modes, [381](#)

EKS (Elastic Kubernetes Service), [244, 446–447](#)
elastic, [305](#)

Elastic Beanstalk, [18, 279–281](#)
build stage, [67](#)
modifying the capacity of the application infrastructure,
[281](#)
updating applications, [282–283](#)

elastic EBS volumes, [370–371](#)

elastic IP address, [575–577](#), [753](#)
elasticity, [12](#), [462](#)
encryption, [21](#)
 Amazon EBS (Elastic Block Storage), [212](#)
 enabling, [212–213](#)
 enabling for each AWS region, [215](#)
 key rotation, [213–214](#)
 select KMS key, [214–215](#)
 envelope, [225–226](#)
 field-level, [238–239](#)
 in transit, [228–229](#)
endpoint, [753](#)
 services, [588–589](#)
VPC (Virtual Private Cloud), [585](#)
 costs, [701–703](#)
 gateway, [585–586](#)
 interface, [586–588](#)
entity, IAM (Identity and Access Management), [82](#)
envelope encryption, [225–226](#)
EOIG (egress-only Internet gateway), [753](#)
ephemeral ports, [159](#), [165–167](#)
ephemeral storage, [362–363](#), [754](#)
event notification, [754](#)
explicit allow permission, [94](#)

external authentication, [81](#), [83](#)
external connections, VPC (Virtual Private Cloud), [180](#)–[181](#)
 customer gateway, [182](#)–[183](#)
 VPG (virtual private gateway), [181](#)–[182](#)
externally authenticated user, [754](#)

F

failover, [21](#), [330](#)–[331](#). *See also* [DR \(disaster recovery\)](#); [high availability](#)
 active-active, [340](#)–[343](#)
 architecture, [60](#)
 multi-region, [555](#)
 origin, [532](#)–[533](#)
 RDS (Relational Database Service), [487](#)–[488](#)
FAQs, [4](#)
fast snapshot restore, [374](#)–[375](#)
fast startup, [69](#)–[70](#)
fault tolerance, [288](#), [293](#), [322](#)–[325](#)
federated identity pool, Amazon Cognito, [179](#)–[180](#)
federation
 SAML 2.0, [122](#)–[124](#)
 web identity, [121](#)–[122](#)
FedRAMP (Federal Risk and Authorization Management Program), [317](#), [754](#)

field-level encryption, [238–239](#)
Firecracker, [437–438](#)
firewall
 NACL (network access control list), [168–169](#)
 cheat sheet, [169–170](#)
 implementation, [169](#)
 rule processing, [170–172](#)
Web Application, [152](#)
 behaviors, [152–153](#)
 rules, [154–167](#)
FISMA, [317](#)
flow log, VPC, [172–174](#), [581–582](#)
FSx for Windows File Server, [386–388](#)

G

gateway endpoint, [585–586](#)
Gateway Load Balancer, [695](#)
gateway service, NAT (network address translation), [174–175](#)
General Purpose SSD, [369–370](#)
GitHub, [63](#)
Glacier, storage at rest, [222–223](#)
global service, [303](#)
global tables, Amazon DynamoDB, [510–511](#)
golden AMI pipeline, [436](#)

GovCloud, [317](#), [318–319](#)
governance, data access, [207](#)
graceful shutdown, [69–70](#)
group, IAM (Identity and Access Management), [82](#), [94](#)
GuardDuty, [187–188](#)
 cheat sheet, [189](#)
 types of security analysis, [188–189](#)
guardrails, [139](#)

H

health check, [754](#)
 ELB, [466](#), [548–550](#)
 Route [53](#), [349–350](#)
Heroku, [17–18](#), [60](#)
high availability, [21](#), [287](#), [288](#), [293](#), [754](#)
 distributed design, [322–325](#)
 endpoints, [304–305](#)
 failover strategies, [330–331](#)
 infrastructure, third-party solutions, [277](#)
 RDS (Relational Database Service), [485–488](#)
HIPAA (Health Insurance Portability and Accountability Act),
regulatory compliance, [316](#)
horizontal scaling, [12](#), [51](#)
hosting, re-, [26–27](#)

hyperthreading, [634](#)

I

IaaS (infrastructure as a service), [6](#), [14–16](#)

IAM (Identity and Access Management), [14](#), [46](#), [79](#), [752](#)

account, options, [95–96](#)

actions, [87–88](#)

authentication, [82–84](#)

external, [83](#)

multifactor, [99](#)

authorization, [85–87](#)

best practices, [128–130](#)

cheat sheet, [132](#)

controls, [210](#)

dashboard, [79](#)

entity, [82](#)

features, [80–81](#)

group, [82](#), [94](#), [754](#)

permission, explicit allow, [94](#)

policy, [81–82](#), [99–100](#)

 ACL (access control list), [113](#)

 actions, [109–110](#)

 conditional, [86](#), [116](#)

 creating, [105–106](#)

elements, [106–107](#)
identity-based, [100–102](#)
inline, [104–105](#)
managed, [100–101](#)
password, [96](#)
permission boundaries, [110–112](#)
permissions, [114–115](#)
resource-based, [102–104](#)
rules, [107–109](#)
service control, [112](#)
session, [113–114](#)
statement, [82, 107](#)
trust, [118](#)
version, [106, 115](#)

principal, [82](#)
requesting access to AWS resources, [84–85](#)
resource, [82](#)
role/s, [82, 118–119, 754](#)
 attaching to EC2 instance, [119–121](#)
 cross-account access, [124–126](#)
 SAML 2.0 federation, [122–124](#)
 service-linked, [119](#)
 for third-party access, [121](#)
when to use, [119](#)

rotating access keys, [97–99](#)
security tools, [130–132](#)
service-linked roles, [80–81](#)
tags, [116–117](#)
user, [81–82](#), [88](#), [90–91](#)
 access keys, [92–94](#)
 creating, [91–92](#)
 signing in as, [94](#)

ID key, [83](#)

identity, [82](#). *See also* [web identity federation](#)
 -based policy, [100–102](#)
 origin access, [531–532](#)

Identity Center, [132–133](#)

IG (Internet gateway), [569](#), [754](#)

Image Builder, [435–436](#)

immutable infrastructure, [327–329](#)

implementation, NACL (network access control list), [169](#)

infrastructure. *See also* [network](#); [Twelve-Factor App Methodology](#)

 authentication, [266](#)
 automation, [277](#). *See also* [AWS](#), Service Catalog; [CloudFormation](#)
 AWS, [16](#)
 as code, [267](#)

dependencies, [63–64](#)
distributed design, [321–322](#)
 high availability and fault tolerance, [322–325](#)
 removing single points of failure, [325–327](#)
immutable, [327–329](#)
security, [209](#)
zone
 Local, [306–307](#)
 Wavelength, [308](#)
inline policy, [104–105](#)
installation, RDS (Relational Database Service), [488–490](#)
instance
 Amazon RDS (Relational Database Service), [483–484](#)
 NAT (network address translation), [175–176](#)
 storage volume, [377–378](#)
integration and integration services
 Amazon EventBridge, [256–258](#)
 Amazon SNS (Simple Notification Service), [248–249](#)
 cheat sheet, [250](#)
 creating a notification topic, [250](#)
 publisher and subscriber options, [249–250](#)
 Amazon SQS (Simple Queue Service), [250–251](#)
 cheat sheet, [253–254](#)
 compatibility with AWS services, [253](#)

components, [251–253](#)
triggered Lambda function, [251](#)
AWS Step Functions, [254–256](#)
CloudWatch, [453–455](#)
interface endpoint, [586–588](#)
intra-AZ connections, [302](#)
IOPS (input/output operations per second), [365](#), [754](#)
iostat command, [370](#)
IP address. *See also* [BYOIP \(Bring Your Own IP\)](#).
 cheat sheet, [577–578](#)
 elastic, [575–577](#)
 private, [573–574](#)
 public, [574–575](#)
IPv6, [580–581](#)
ISO/IEC 27001 security standard, [80](#)
ITIL (Information Technology Infrastructure Library), [267](#)

J-K

Jassy, A., [6](#)
key, [390](#)
key rotation, EBS, [213–214](#)
key-value, [754](#)
KMS (Key Management Service), [224](#), [752](#), [754](#)
 cheat sheet, [226–227](#)

console, [224–225](#)
envelope encryption, [225–226](#)

L

labs, AWS Well-Architected, [4–5](#)
Lambda@Edge, [535–536](#), [754](#)
latency, region selection and, [319–320](#)
launch configuration, [464](#)
launch template, [464](#), [754](#)
LCU (Load Balancer Capacity Unit), [695](#), [755](#)
least privilege, [46](#)
lifecycle
 hook, [472–473](#), [755](#)
 management, EFS (Elastic File System), [383](#)
 policy, [755](#)
 rules, [619–620](#), [755](#)
lift and shift, [26–27](#)
Linux, taking an EBS snapshot, [373–374](#)
listener, [543–545](#), [755](#)
list-policies command, [105](#)
live streaming, [238](#), [533–534](#)
load balancer, [240–241](#)
 Amazon ELB, [539](#)
 application load balancer deployment, [541–545](#)

costs, [695–696](#)
features, [540–541](#)
sticky session support, [551–552](#)

application, [23](#), [541–543](#)
access logs, [553](#)
cheat sheet, [553](#)
health checks, [548–550](#)
listeners and routing, [543–545](#)
rules, conditions, and actions, [545–547](#)
sticky session support, [551–552](#)
target group attributes, [550–551](#)
target groups, [547–548](#)

network, [554](#)
cheat sheet, [554–555](#)
multi-region failover, [555](#)

placement, [57–58](#)

local instance storage, [377–378](#)

Local Zone, [306–307](#), [755](#)

logs and logging, [70](#)
access, [553](#)
CloudWatch, [449–451](#)
flow, [172–174](#), [581–582](#)

M

main route table, [155](#)
managed policy, [100–101](#)
managed service, [293](#)
 AWS, [19–20](#)
 Lambda@Edge, [535–536](#)
 use cases, [308–310](#)
management console, AWS, [9](#)
management options, ASG (Auto Scaling group), [470–471](#)
measured service, [12–13](#)
Memcached, Amazon ElastiCache, [513–514](#)
metadata
 object, [391](#)
 XML, [123](#)
metrics, CloudWatch, [53](#), [447](#), [455](#), [555–556](#)
MFA (multifactor authentication), [22](#), [80–81](#), [99](#), [755](#)
Microsoft Azure, [6](#), [39](#)
migration
 application, [24](#)
 allow access to on-premises data records, [26](#)
 define a value proposition, [24–25](#)
 lift and shift, [26–27](#)
 with many local dependencies, [27–28](#)
 solve a single problem, [26](#)
 start with low value/low risk, [25–26](#)

applications that should remain on premises, [28](#)
to containers, [246](#)
data transfer options, [621–625](#)
database, [680–681](#)

mindset

born-in-the-cloud, [14](#)
corporate, [13](#)
startup, [14](#)

modular design, [237](#)

monitoring, [16](#), [490–491](#). *See also* [CloudWatch](#)
multipart upload, [755](#)
multi-region warm standby, [339](#)

N

NACL (network access control list), [168–169](#), [244](#), [755](#)
cheat sheet, [169–170](#)
implementation, [169](#)
rule processing, [170–172](#)

NAT (network address translation), [174](#)
cheat sheet, [176](#)
costs, [696–697](#)
gateway service, [174–175](#), [755](#)
instance, [175–176](#)

network, [51](#). *See also* [edge locations](#)

access control list, [168–169](#), [244](#)
 cheat sheet, [169–170](#)
 implementation, [169](#)
 rule processing, [170–172](#)

address translation, [174](#)
 cheat sheet, [176](#)
 gateway service, [174–175](#)
 instance, [175–176](#)

BYOIP (Bring Your Own IP), [579–580](#)

data transfer costs, [714](#)

IP address
 elastic, [575–577](#)
 private, [573–574](#)
 public, [574–575](#)

load balancer, [58](#), [554](#)
 cheat sheet, [554–555](#)
 multi-region failover, [555](#)

resiliency, [304](#)

security, [22–23](#), [149–150](#), [151–152](#). *See also* [security](#)
 shared security model, [557–558](#)
 terminology, [558–559](#)
 topology, planning, [303–306](#)
 traffic charges, [578–579](#)

VPC (Virtual Private Cloud), [154](#), [556–557](#). See also [VPC \(Virtual Private Cloud\)](#).

- calculating number required, [564–565](#)
- cheat sheet, [560–561](#)
- connectivity options, [583](#)
- creating, [561–564](#)
- creating the CIDR block, [565–569](#)
- default, [569–570](#)
- endpoints, [585–589](#)
- flow log, [581–582](#)
- peering, [583–585](#)
- route table, [154–158](#)
- SG (security group), [158–168](#)
- subnet, [570–573](#)

NIST (National Institute of Standards and Technology)

- compliance, [316–317](#)

- SP 800–145, “The NIST Definition of Cloud Computing”, [8–9](#)

- broad network access, [10–11](#)
- on-demand self-service, [9](#)
- measured service, [12–13](#)
- rapid elasticity, [11–12](#)
- resource pooling, [10](#)

Nitro, [755](#)

non-persistent data store, [378](#)
NoSQL, [675–677](#), [755](#)
 costs cheat sheet, [676–680](#)
 service comparisons, [676–677](#)
NVMe (Non-Volatile Memory Express), [755](#)

O

OAI (origin access identity), [531–532](#), [756](#)
object
 metadata, [391](#)
 S3, [390](#)
 storage, [362](#), [756](#)
object lock policy, Amazon S3, [221–222](#)
operational benefits, AWS, [19–20](#)
operational excellence, [43](#), [44–45](#)
origin failover, [756](#)
outages, [306](#)
Outposts, [7](#), [357](#)

P

PaaS (platform as a service), [17–18](#)
 Cloud Foundry, [17](#)
 Elastic Beanstalk, [18](#)

Heroku, [17–18](#)
password policy, [96](#), [756](#)
Paxos, [508](#)
PCI DSS (Payment Card Industry Data Security Standard),
compliance checklist, [313–314](#)
peering, [583–585](#), [756](#)
performance
 efficiency, [49–51](#)
 modes, EFS, [380–381](#)
 RDS (Relational Database Service), [490–491](#)
 and reliability, [54](#)
 Well-Architected Framework, [29](#)
permission/s, [105](#), [114](#)
 Amazon S3, [216–217](#)
 boundaries, [110–112](#)
 explicit allow, [94](#)
 summary table, [114–115](#)
PII (personally identifiable information), [207](#)
pilot light, [333–337](#), [756](#)
PIOPS (provisioned input/output operations per second), [365](#)
placement group, EC2 (Elastic Compute Cloud), [638](#)
placing cloud services, [55](#)
 CDN, [56–57](#)
 data replication, [57–58](#)

data residency and compute locations, [55–56](#)

load balancer, [57–58](#)

planning

- network topology, [303–306](#)
- security group, [167–168](#)

policy/ies

- ACL (access control list), [113](#)
- bucket, [217–218](#)
- conditional, [86, 116](#)
- database retention, [687–689](#)
- IAM (Identity and Access Management), [81–82, 99–100](#)
 - actions, [109–110](#)
 - creating, [105–106](#)
 - elements, [106–107](#)
 - identity-based, [100–102](#)
 - inline, [104–105](#)
 - permission boundaries, [110–112](#)
 - resource-based, [102–104](#)
 - rules, [107–109](#)
 - session, [113–114](#)
 - statement, [82](#)
 - version, [106](#)
- identity-based
 - custom, [102](#)

managed, [100–101](#)
lifecycle, [755](#)
object lock, [221–222](#)
password, [96](#)
permissions, [105](#), [114–115](#)
retrieval, [405–406](#)
routing, [350](#)
scaling, cooldown period, [471](#)
service control, [112](#)
stickiness, [552](#)
termination, [471–472](#)
traffic flow, [351](#)
trust, [118](#)
version, [115](#)
WORM (write-once/read-many), [221](#)

POP (point of presence), [56](#)
port binding, [69](#)
pricing
 AWS, [321](#)
 CloudFront, [700–701](#)
 EC2 (Elastic Compute Cloud), [655](#)
 Reserved instance, [648–649](#)
 tiered, [599–600](#)
primary CIDR block, [566–568](#)

primary database, [756](#)
principal, IAM (Identity and Access Management), [82](#)
private IP address, [573–574](#)
product, Service Catalog, [277–279](#)
production, [70](#)
providers, [39](#)
provisioned mode, EFS, [381](#)
public cloud, [6–7](#)
public IP address, [574–575](#)

Q

queue, [756](#)
quotas
on-demand service, [641–643](#)
service, [345–348, 391](#)

R

rapid elasticity, [11–12](#)
RDS (Relational Database Service), [481–482, 668–670](#)
best practices, [491](#)
cheat sheet, [493](#)
costs cheat sheet, [671](#)
data transfer costs, [682–683](#)

design solutions, [672–675](#)
engines, [482](#)
failover, [487–488](#)
high-availability design, [485–488](#)
installation, [488–490](#)
instance class types, [485](#)
instances, [483–484](#)
Multi-AZ deployment, [488](#)
performance monitoring, [490–491](#)
Proxy, [492–493](#)
read replica, [673](#)
standby, [487](#)
read capacity unit, [756](#)
read replica, [673](#), [756](#)
recycle bin, EBS (Elastic Block Store), [376](#)
Redis, Amazon ElastiCache, [514–517](#)
redundancy, [21](#), [48](#), [54](#)
regional edge cache, [756](#)
regional Reserved instance, [647](#)
region/s, [296–299](#), [756](#)
 cheat sheet, [343–344](#)
DR (disaster recovery)
 backup and restore, [332–333](#)
 pilot light, [333–337](#)

edge cache, [528–529](#)
GovCloud, [318–319](#)
selection criteria, [310](#)
 compliance rules, [311–319](#)
 latency concerns, [319–320](#)
 pricing, [321](#)
 services, [320](#)
warm standby, [337–339](#), [340](#)

regulatory compliance, [207](#)
 HIPAA (Health Insurance Portability and Accountability Act), [316](#)
 rules, [311–314](#)
 standards, [315](#)

re-hosting, [26–27](#)

reliability, [287](#), [295–296](#), [757](#)
 and performance, [54](#)
 Well-Architected Framework, [29](#), [47–49](#)

replacing, applications, [28](#)

replication, [223–224](#)
 Amazon Aurora, [498–499](#)
 S3 (Simple Storage Service), [397–398](#)

Reserved instance, [644–645](#), [757](#)
 payment options, [646](#)
 pricing, [648–649](#)

regional versus zonal, [647](#)
reviewing monthly charges, [648](#)
scheduled reservation, [646–647](#)
scope, [647](#)
term commitment, [645](#)
types, [646](#)

resiliency, [237](#), [246–247](#), [288](#), [304](#)

resolver, Route [53](#), [352–353](#)

Resource Access Manager, [136–138](#)

resource pooling, [10](#)

resource/s

- actions, [87–88](#)
- based policy, [102–104](#)
- IAM (Identity and Access Management), [82](#)
 - requesting access, [84–85](#)
- responsibilities, AWS cloud provider, [20–21](#)
- retrieval policy, Amazon S3 Glacier, [405–406](#)

role/s

- IAM (Identity and Access Management), [82](#), [118–119](#)
 - attaching to EC2 instance, [119–121](#)
 - cross-account access, [124–126](#)
 - service-linked, [119](#)
 - when to use, [119](#)
- SAML 2.0 federation, [122–124](#)

for third-party access, [121](#)
web identity federation, [121–122](#)

root user, [88–90](#)

Route [53](#), [59](#), [150](#), [348–349](#)
alias records, [352](#)
health checks, [349–350](#)
resolver, [352–353](#)
routing policies, [350](#)
traffic flow policies, [351](#)

route propagation, [184–185](#)

route table
cheat sheet, [158](#)
custom, [155–158](#)
main, [155](#)

routing, ALB (Application Load Balancer), [543–545](#)

RPO (recovery point objective), [54](#), [331](#), [756](#)

RTO (recovery time objective), [54](#), [331](#), [756](#)

rules
actions, [546–547](#)
Amazon ELB, [545–547](#)
Amazon Inspector, [195–196](#)
AWS Service Catalog, [278](#)
compliance, [311–319](#)
IAM policy, [107–109](#)

lifecycle, [619–620](#), [755](#)
NACL (network access control list), [170–172](#)
regulatory compliance, [311–314](#)
SG (security group), [162](#)
WAF (Web Application Firewall), [152](#), [154–167](#)

S

S3 (Simple Storage Service)
batch operations, [396](#)
bucket versioning, [400–401](#)
inventory, [399](#)
object lock, [396](#)
replication, [397–398](#)
SAA-CO3 exam, [721–724](#)
preparation tools, [726–731](#)
sample questions, [5–6](#)
scaled scoring, [4](#)
scheduling, [725–726](#)
tips, [724–725](#)
updates, [749–750](#)
SaaS (software as a service), [13](#)
SAML 2.0 federation, [122–124](#)
sample questions, SAA-CO3 exam, [5–6](#)
Savings Plans, [649–650](#)

scale out, [757](#)

scaled scoring, [4](#)

scaling. *See also* [auto scaling](#)

ASG (Auto Scaling group), [466–470](#)

auto, [461–463](#)

AWS, [473–474](#)

cheat sheet, [473](#)

EC2 (Elastic Compute Cloud), [463–473](#)

compute, [661](#)

cooldown period, [471](#)

horizontal, [12](#), [51](#)

policy, [757](#)

termination policy, [471–472](#)

scope, Reserved instance, [647](#)

Scrum, [267](#)

SDN (software-defined network), [14](#)

secondary CIDR block, [568–569](#)

secret access key, [83](#)

security. *See also* [authentication](#); [encryption](#)

Amazon Macie, [189–191](#)

AWS, [21](#)

application, [23–24](#)

data, [21–22](#)

network, [22–23](#)

controls

- detective, [210–212](#)
- IAM, [210](#)
 - Defense in Depth, [45–47](#)
 - edge location, [150–151](#)
 - AWS Shield, [151–152](#)
 - WAF (Web Application Firewall), [152–154](#)
 - EFS, [382](#)
 - group, [23](#), [757](#)
 - IAM (Identity and Access Management), [79](#). *See also* [IAM \(Identity and Access Management\)](#)
 - access keys, [92–94](#)
 - account options, [95–96](#)
 - ACL (access control list), [113](#)
 - actions, [87–88](#)
 - authorization, [85–87](#)
 - best practices, [128–130](#)
 - cheat sheet, [132](#)
 - conditional policy, [116](#)
 - custom policy, [102](#)
 - dashboard, [79](#)
 - entity, [82](#)
 - explicit allow permission, [94](#)
 - external authentication, [83](#)

features, [80–81](#)
group, [82](#), [94](#)
identity-based policy, [100–102](#)
inline policy, [104–105](#)
managed policy, [100–101](#)
MFA (multifactor authentication), [99](#)
password policy, [96](#)
permission boundaries, [110–112](#)
policy, [81–82](#), [99–100](#)
policy, creating, [105–106](#)
policy actions, [109–110](#)
policy elements, [106–107](#)
policy rules, [107–109](#)
policy statement, [107](#)
policy version, [106](#), [115](#)
principal, [82](#)
requesting access to AWS resources, [84–85](#)
resource, [82](#)
resource-based policy, [102–104](#)
role, [82](#), [118–121](#)
rotating access keys, [97–99](#)
service-linked roles, [80–81](#)
session policy, [113–114](#)
signing in as a user, [94](#)

tags, [116–117](#)
tools, [130–132](#)
trust policy, [118](#)
user, [81–82](#), [90–91](#)
user, creating, [91–92](#)
infrastructure, [209](#)
network, [149–150](#)
 AWS Shield, [151–152](#)
 VPC, [154–176](#)
Well-Architected Framework, [29](#), [43–44](#), [45–47](#)
workshops, [5](#)
self-service, AWS, [9](#)
server
 AWS, [19](#)
 immutable, [327–328](#)
serverless, [237–238](#), [757](#)
 Amazon Aurora, [674–675](#)
 web app, building, [262](#)
 create a static website, [263](#)
 create the backend components, [264](#)
 register for the conference, [266](#)
 set up the API gateway, [265](#)
 user authentication, [263–264](#)
service control policy, [112](#)

service/s

analytical, [415–416](#)

AWS, [15–16](#)

 cheat sheet, [31–36](#)

 compute, [15](#)

 database, [15](#)

 monitoring, [16](#)

 PaaS, [17–18](#)

 storage, [15](#)

 VMWare, [16](#)

AWS CloudTrail, [191–192](#)

 cheat sheet, [194](#)

 creating a custom trail, [192–194](#)

AWS Config, [199–200](#)

AWS Trusted Advisor, [196–198](#)

backing, [66–67](#)

backup and restore, [332–333](#)

CDN, placement, [56–57](#)

compute, [425–427](#)

 Amazon Lambda, [436–441](#)

 EC2, [427–436](#). *See also* [EC2 \(Elastic Compute Cloud\)](#)

container, [441–443](#). *See also* [containers and container management](#)

Amazon ECS (Elastic Container Service), [443–446](#)

Amazon EKS (Elastic Kubernetes Service), [446–447](#)

costs, [598–599](#)

data replication, placement, [57–58](#)

detective control, [211–212](#)

endpoint, [588–589](#)

global, [303](#)

IaaS, [14–16](#)

immutable infrastructure, [328–329](#)

-linked roles, [80–81](#)

load balancer, placement, [57–58](#)

managed, [293](#), [308–310](#)

PaaS

- Cloud Foundry, [17](#)
- Heroku, [17–18](#)
- placing, [55–56](#)
- quota, [3–4](#), [345–348](#), [757](#)
- serverless, [237–238](#)
- storage, [329–330](#)
- tiered pricing, [599–600](#)

session policy, [113–114](#)

SG (security group), [158–161](#)

- administration access, [164–165](#)
- cheat sheet, [161–162](#)
- custom, [162–163](#)

database server inbound ports, [163–164](#)
ephemeral ports, [159](#), [165–167](#)
planning, [167–168](#)
rules, [162](#)
web server inbound ports, [163](#)

shared memory segment, [371](#)
shared responsibility model, [79–80](#)
shared security model, [557–558](#)
signing in as user, IAM (Identity and Access Management), [94](#)
simple scaling, [757](#)
single points of failure, removing, [325–327](#)
SLA (service-level agreement), [14](#), [20–21](#), [47–48](#), [52–53](#), [294](#),
[757](#)
SLI (service-level indicator), [52–53](#), [757](#)
SLO (service-level objective), [52–53](#), [757](#)
SMB (Server Message Block), [757](#)
snapshot, [295](#), [362](#), [757](#)
 cheat sheet, [376–377](#)
EBS (Elastic Block Store), [373](#)
 administration, [375–376](#)
 taking from a Linux instance, [373–374](#)
 taking from a Windows instance, [374](#)
fast restore, [374–375](#)
Snow device, [757](#)

SP 800–145, “The NIST Definition of Cloud Computing”, [8–9](#)

- broad network access, [10–11](#)
- on-demand self-service, [9](#)
- measured service, [12–13](#)
- rapid elasticity, [11–12](#)
- spot capacity pool, [653–655](#)

Spot Fleet, [651–653](#)

spot instance, [650–651](#)

SQL (Structured Query Language), [503, 758](#)

SSE (server-side encryption), [757](#)

SSO (single sign-on), [83, 132–133](#)

stack sets, [276–277](#)

stacks, AWS CloudFormation, [272–273](#)

staging, [70](#)

standard/s

- ISO/IEC 27001, [80](#)
- regulatory compliance, [311–312, 315](#)

standby database, [757](#)

startup, mentality, [14](#)

stateful, [161, 239–243, 758](#)

stateless, [239–243, 758](#)

statement, policy, [82, 107](#)

Step Functions, [254–256](#)

step scaling, [758](#)

sticky session, [243](#), [551–552](#), [758](#)
storage, [329–330](#), [362](#)
 Amazon Aurora, [496–498](#)
 Amazon S3, [15](#)
 AWS, [19](#)
 block, [362](#)
 classes
 EFS, [382](#)
 S3, [394–396](#)
 cold, [753](#)
 controlled, [373](#)
 costs, [613–617](#)
 EBS (Elastic Block Store), [365–366](#). *See also* [EBS \(Elastic Block Store\)](#)
 administration, [375–376](#)
 attaching a volume, [371–372](#)
 cheat sheet, [372–373](#)
 elastic EBS, [370–371](#)
 fast snapshot restore, [374–375](#)
 General Purpose SSD, [369–370](#)
 multi-attach, [366](#)
 recycle bin, [376](#)
 snapshot, [373–376](#)
 volume types, [367–369](#)

ephemeral, [362–363](#)
instance, [377–378](#)
object, [362](#)
resiliency, [246–247](#)
at rest, [220–221](#), [222–223](#)
workload requirements, [363–365](#)
streaming, [238](#)
structured data, [411–412](#)
STS (Security Token Service), [120](#), [126–128](#)
subnet, [570–573](#), [758](#)
sustainability, Well-Architected Framework, [30](#), [51–52](#)
symmetric key, [758](#)
syntax, IAM policy, [107–109](#)

T

T instance, [758](#)
tag
cost allocation, [612–613](#)
createdBy, [612](#)
IAM (Identity and Access Management), [116–117](#)
target group, [465–466](#), [547–548](#), [550–551](#), [758](#)
task definition, [443–446](#), [758](#)
template
AWS CloudFormation, [270–272](#)

launch, [464](#)
termination policy, [471–472](#)
throughput modes, EFS, [381](#)
tiered pricing, [599–600](#), [758](#)
time
 availability, [293–295](#). *See also* [availability](#)
 up, [323](#), [331](#)
tool/s
 analytical, [412–413](#)
 Artifact, [311](#)
 automation, [266–277](#)
 AWS Config, [600–602](#)
 AWS Schema Conversion, [681](#)
 cost management, [602–604](#)
 Budgets, [607–609](#)
 Cost and Usage Reports, [609–610](#)
 Cost Explorer, [604–607](#)
 exam preparation, [726–731](#)
 IAM (Identity and Access Management), [130–132](#)
 Well-Architected Framework, [5](#), [30–31](#)
traffic
 billing, [578–579](#)
 flow policy, [351](#)
trust policy, [118](#)

TTL (time to live), [758](#)

Twelve-Factor App Methodology, [60–61](#), [62](#)

declare and isolate dependencies, [65](#)

execute an app as one or more stateless processes, [67–68](#)

export services via port binding, [69](#)

keep development, staging, and production similar, [70](#)

maximize robustness with fast startup and graceful shutdown, [69–70](#)

run admin/management tasks as on-off processes, [71](#)

scale out via the process model, [69](#)

separate build and run stages, [67](#)

store configuration in the environment, [66](#)

treat backing services as attached resources, [66–67](#)

treat logs as event streams, [70](#)

use one codebase, [63–64](#)

U

unstructured data, [412](#)

uptime, [323](#), [331](#), [758](#)

use cases

Amazon CloudFront, [529](#)

Lambda@Edge, [535–536](#)

managed service, [308–310](#)

user, [88](#)

IAM (Identity and Access Management), [81–82](#), [90–91](#)
 creating, [91–92](#)
 signing in as, [94](#)
root, [88–90](#)
session management
 distributed, [243–247](#)
 sticky sessions, [243](#)
state, [758](#)
user pool, Amazon Cognito, [177–179](#)

V

value proposition, define, [24–25](#)
vault, Amazon S3 Glacier, [405](#)
vCPU (virtual CPU), [634](#)
versioning, [758](#)
 bucket, [400–401](#)
 IAM policy, [106](#), [115](#)
virtual machine/s, [8](#)
VM (virtual machine), [442](#)
VMWare, on AWS, [16](#)
Vogels, W., [288](#)
volume
 EBS (Elastic Block Store)
 attaching, [371–372](#)

snapshot, [373](#)
types, [369–371](#)

instance storage, [377–378](#)

VPC (Virtual Private Cloud), [15](#), [22–23](#), [154](#), [556–557](#), [758](#)
calculating number required, [564–565](#)
cheat sheet, [560–561](#)
connectivity options, [583](#)
creating, [561–564](#)
creating the CIDR block, [565–566](#)
primary, [566–568](#)
secondary, [568–569](#)
default, [569–570](#)
endpoint/s, [585](#)
 costs, [701–703](#)
 gateway, [585–586](#)
 interface, [586–588](#)
 services, [588–589](#)
external connections, [180–181](#)
 customer gateway, [182–183](#)
 route propagation, [184–185](#)
 VPG (virtual private gateway), [181–182](#)
flow log, [172–174](#), [581–582](#)
NACL (network access control list), [168–169](#)
 cheat sheet, [169–170](#)

implementation, [169](#)
rule processing, [170–172](#)

NAT (network address translation), [174](#)
cheat sheet, [176](#)
gateway service, [174–175](#)
instance, [175–176](#)

network terminology, [558–559](#)

peering, [583–585](#)

route table, [154](#)
cheat sheet, [158](#)
custom, [155–158](#)
main, [155](#)

SG (security group), [158–161](#)
administration access, [164–165](#)
cheat sheet, [161–162](#)
custom, [162–163](#)
database server inbound ports, [163–164](#)
ephemeral ports, [159](#), [165–167](#)
planning, [167–168](#)
rules, [162](#)
web server inbound ports, [163](#)

shared security model, [557–558](#)

subnet, [570–573](#)

VPG (virtual private gateway), [181–182](#), [758](#)

VPN (virtual private network), [10](#), [149](#)

AWS solutions, [183–184](#)

route propagation, [184–185](#)

W

WAF (Web Application Firewall), [23](#), [151](#), [152–153](#), [154–167](#)

warm standby, [759](#)

Amazon Aurora, [340](#)

multi-region, [339](#)

Waterfall, [61–62](#)

Wavelength Zone, [308](#)

WCU (write capacity unit), [759](#)

web identity federation, [121–122](#)

web server inbound ports, security group, [163](#)

Well-Architected Framework, [28–30](#), [39–40](#), [42](#), [287–288](#), [752](#)

best practices, [42](#)

cost optimization, [51](#)

Microsoft Azure, [39](#)

operational excellence, [43](#), [44–45](#)

performance efficiency, [49–51](#)

reliability, [47–49](#)

security, [43–44](#), [45–47](#)

sustainability, [51–52](#)

tool, [30–31](#)

Wiggins, A., [60](#)

Windows

AMI (Amazon Machine Image), [431](#)

taking an EBS snapshot, [374](#)

workload, [293](#)

availability, [48](#), [294](#)

dependencies, [48](#), [54](#)

reliability, [295](#)

SLA (service-level agreement), [52–53](#)

storage requirements, [363–365](#)

workshop, AWS security, [5](#)

WORM (write-once/read-many) policy, [221](#), [759](#)

X-Y-Z

XML, metadata, [123](#)

zonal Reserved instance, [647](#)

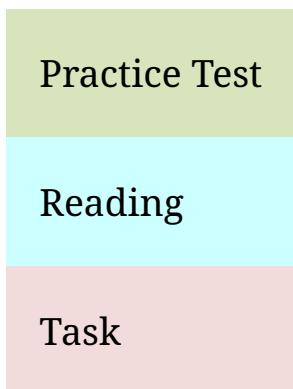
zone. *See also* [availability zone](#)

Local, [306–307](#)

Wavelength, [308](#)

Appendix C

Study Planner



Element	Task	Goal	First Date	Second Date
		Date	Completed	Date
			Comments	
Introduction	Read Introduction			
2. The AWS Well-Architected Topics	Read Foundation Topics			

Framework

2. The AWS Well-Architected Framework

Review Key Topics

2. The AWS Well-Architected Framework

Define Key Terms

2. The AWS Well-Architected Framework

Complete Q&A section

Practice Test

Take practice test in study mode using Exam Bank 1 questions for Chapter

2 in practice

test

software

3. Designing Read
Secure Foundation
Access to Topics
AWS
Resources

3. Designing Review Key
Secure Topics
Access to
AWS
Resources

3. Designing Define Key
Secure Terms
Access to
AWS
Resources

3. Designing Complete
Secure Q&A section

Access to AWS Resources

Practice Test

Take practice test in study mode using Exam Bank 1 questions for Chapter 3 in practice

test software

4. Designing Secure Workloads and Applications

Read Foundation Topics

4. Designing Secure Workloads and Applications

Review Key Topics

Workloads and Applications

4. Designing Define Key
Secure Terms
Workloads
and
Applications

4. Designing Complete
Secure Q&A section
Workloads
and
Applications

Practice Test Take
practice test
in study
mode using
Exam Bank
1 questions
for Chapter
4 in practice

test software	
5.	Read
Determining	Foundation
Appropriate	Topics
Data Security	
Controls	
5.	Review Key
Determining	Topics
Appropriate	
Data Security	
Controls	
5.	Define Key
Determining	Terms
Appropriate	
Data Security	
Controls	
5.	Complete
Determining	Q&A section

Appropriate Data Security Controls

Practice Test Take
 practice test
 in study
 mode using
 Exam Bank
 1 questions
 for Chapter
 5 in practice

test
software

6. Designing Read
Resilient Foundation
Architecture Topics

6. Designing Review Key
Resilient Topics
Architecture

6. Designing Resilient Architecture

venne key
Terms

6. Designing Resilient Architecture

Complete
Q&A section

Practice Test

Take
practice test
in study
mode using
Exam Bank

1 questions
for Chapter
6 in practice
test
software

7. Designing Highly Available and Fault-Tolerant

Read
Foundation
Topics

Architecture

7. Designing
Highly Available
and Fault-
Tolerant
Architecture

Review Key

Topics

7. Designing
Highly Available
and Fault-
Tolerant
Architecture

Define Key

Terms

7. Designing
Highly Available
and Fault-
Tolerant
Architecture

Complete

Q&A section

Practice Test

Take

practice test
in study
mode using
Exam Bank
1 questions
for Chapter
7 in practice
test
software

8. High-
Performing
and Scalable
Storage
Solutions

8. High-
Performing
and Scalable
Storage
Solutions

8. High-
Performing
Terms

and Scalable

Storage

Solutions

8. High-
Performing
and Scalable
Storage
Solutions

Practice Test Take
 practice test
 in study

mode using
Exam Bank
1 questions
for Chapter
8 in practice
test
software

9. Designing Read
High-
Performing Foundation
Topics

and Elastic Compute Solutions

9. Designing
High-
Performing
and Elastic
Compute
Solutions

Review Key
Topics

9. Designing
High-
Performing
and Elastic
Compute
Solutions

Define Key
Terms

9. Designing
High-
Performing
and Elastic
Compute
Solutions

Complete
Q&A section

9. Designing
High-
Performing
and Elastic
Compute
Solutions

Practice Test Take
 practice test
 in study
 mode using
 Exam Bank
 1 questions
 for Chapter
 9 in practice
 test
 software

10. Read

Determining Foundation
High- Topics

Performing
Database
Solutions

10. Review Key

Determining Topics
High-
Performing
Database
Solutions

Solutions

10. Define Key
Determining Terms
High-
Performing
Database
Solutions

10. Complete
Determining Q&A section
High-
Performing
Database
Solutions

Practice Test Take
practice test
in study
mode using
Exam Bank
1 questions
for Chapter
10 in

practice test
software

11. High-
Performing
and Scalable
Networking
Architecture

11. High-
Performing
and Scalable

Networking
Architecture

11. High-
Performing
and Scalable
Networking
Architecture

11. High-
Performing

1. 2. 3. 4.

Define Key
Terms

Complete
Q&A section

and Scalable Networking Architecture

Practice Test Take practice test in study mode using Exam Bank 1 questions for Chapter 11 in

practice test software

12. Designing Cost-Optimized Storage Solutions Read Foundation Topics

12. Designing Cost-Optimized Storage Solutions Review Key Topics

Optimized Storage Solutions

12. Designing Define Key
Cost- Terms
Optimized
Storage
Solutions

12. Designing Complete
Cost- Q&A section
Optimized
Storage
Solutions

Practice Test Take
 practice test
 in study
 mode using
 Exam Bank
 1 questions
 for Chapter
 12 in

practice test
software

13. Designing Read
Cost-Effective Foundation
Compute Topics
Solutions

13. Designing Review Key
Cost-Effective Topics
Compute
Solutions

13. Designing Define Key
Cost-Effective Terms
Compute
Solutions

13. Designing Complete
Cost-Effective Q&A section
Compute
Solutions

Practice Test Take

practice test
in study
mode using
Exam Bank
1 questions
for Chapter
13 in
practice test
software

14. Designing Read
Cost-Effective Foundation
Topics

Database
Solutions

14. Designing Review Key
Cost-Effective Topics
Database
Solutions

14. Designing Define Key
Cost-Effective Terms
Database
Solutions

Solutions

14. Designing Complete
Cost-Effective Q&A section
Database
Solutions

Practice Test Take
practice test
in study
mode using
Exam Bank
1 questions

for Chapter
14 in
practice test
software

15. Designing Read
Cost-Effective Foundation
Network Topics
Architectures

15. Designing Review Key
Cost-Effective

Cost-Effective Topics

Network
Architectures

15. Designing Cost-Effective Network Architectures

Define Key

Terms

Network
Architectures

15. Designing Cost-Effective Network Architectures

Complete

Q&A section

Network
Architectures

Practice Test

Take practice test in study mode using Exam Bank 1 questions for Chapter 15 in practice test software

16. Final Preparation

16. Final Preparation	Take practice test in study mode for all book questions in practice test software
16. Final Preparation	Review all Key Topics in all chapters
16. Final Preparation	Take practice test in practice exam mode using Exam Bank #1 questions

for all
chapters

16. Final Preparation Take practice test in practice exam mode using Exam Bank #2 questions for all chapters



Exclusive Offer – 40% OFF

Pearson IT Certification Video Training

livelessons®

pearsonitcertification.com/video

Use coupon code **PITCVIDEO40** during checkout.



Video Instruction from Technology Experts



Advance Your Skills

Get started with fundamentals,
become an expert.



Train Anywhere

Train anywhere, at your
own pace, on any device.



Learn

Learn from trusted author
trainers published by

or get certified.

Pearson IT Certification.

Try Our Popular Video Training for FREE!

pearsonitcertification.com/video

Explore hundreds of **FREE** video lessons from our growing library of Complete Video Courses, LiveLessons, networking talks, and workshops.

PEARSON
IT CERTIFICATION

ALWAYS LEARNING

pearsonitcertification.com/video

PEARSON



REGISTER YOUR PRODUCT at PearsonITcertification.com/register
Access Additional Benefits and SAVE 35% on Your Next Purchase

- Download available product updates.
- Access bonus material when applicable.
- Receive exclusive offers on new editions and related products.
(Just check the box to hear from us when setting up your account.)
- Get a coupon for 35% for your next purchase, valid for 30 days. Your code will be available in your PITC cart. (You will also find it in the Manage Codes section of your account page.)

Registration benefits vary by product. Benefits will be listed on your account page under Registered Products.

PearsonITcertification.com—Learning Solutions for Self-Paced Study, Enterprise, and the Classroom

Pearson is the official publisher of Cisco Press, IBM Press, VMware Press, Microsoft Press,

HP Press, O'Reilly Media Press, and Training & Technology Press.

and is a Platinum CompTIA Publishing Partner—CompTIA's highest partnership accreditation.

At PearsonITcertification.com you can

- Shop our books, eBooks, software, and video training.
- Take advantage of our special offers and promotions (pearsonitcertification.com/promotions).
- Sign up for special offers and content newsletters (pearsonitcertification.com/newsletters).
- Read free articles, exam profiles, and blogs by information technology experts.
- Access thousands of free chapters and video lessons.

[Connect with PITC – Visit PearsonITcertification.com/community](#)

Learn about PITC community events and programs.



PEARSON IT CERTIFICATION

Addison-Wesley • Cisco Press • IBM Press • Microsoft Press • Pearson IT Certification • Prentice Hall • Que • Sams • VMware Press

ALWAYS LEARNING

PEARSON

AWS Certified Solutions Architect – Associate (SAA-C03) Cert Guide

ISBN: 978-0-13-794158-2

See inside ▶▶▶

for your Pearson Test Prep activation code and special offers

Complete Video Course

To enhance your preparation, Pearson IT Certification also sells Complete Video Courses for both streaming and download. Complete Video Courses provide you with hours of expert-level instruction mapped directly to exam objectives.

Special Offer—Save 70%

This single-use coupon code will allow you to purchase a Complete Video Course at a 70% discount. Simply go to the product URL below, add the Complete Video Course to your cart, and apply the coupon code at checkout.

AWS Certified Solutions Architect Associate (SAA-C03)

Complete Video Course

www.pearsonitcertification.com/title/9780138057411

Coupon Code:

AWS Certified Solutions Architect – Associate (SAA-C03) Cert Guide

Premium Edition eBook and Practice Test

To enhance your preparation, Pearson IT Certification also sells a digital Premium Edition of this book. The Premium Edition provides you with two eBook files (PDF and EPUB) as well as an enhanced edition of the Pearson Test Prep practice test software. The Premium Edition includes four practice exams with links for every question mapped to the PDF eBook.

Special Offer—Save 80%

This single-use coupon code will allow you to purchase a copy of the Premium Edition at an **80% discount**. Simply go to the URL below, add the Premium Edition to your cart, and apply the coupon code at checkout.

www.pearsonITcertification.com/title/9780137941568

Coupon Code:

DO NOT DISCARD THIS NUMBER

You will need this activation code to activate your practice test in the Pearson Test Prep practice test software.

To access the online version, go to www.PearsonTestPrep.com.

Select **Pearson IT Certification** as your product group. Enter your email/password for your account. If you don't have an account on PearsonITCertification.com or CiscoPress.com, you will need to establish one by going to

PearsonITCertification.com/join. In the My Products tab, click the **Activate New Product** button. Enter the access code printed on this insert card to activate your product. The product will now be listed in your My Products page.

If you wish to use the Windows desktop offline version of the application, simply register your book at

www.pearsonITcertification.com/register, select the **Registered Products** tab on your account page, click the **Access Bonus Content** link, and download and install the software from the companion website.

This activation code can be used to register your exam in both the online and the offline versions.

Activation Code:



Pearson

Where are the companion content files?

Register this digital version of

[AWS Certified Solutions Architect – Associate \(SAA-C03\) Cert Guide](#)

to access important downloads.

Register this eBook to unlock the companion files. Follow these steps:

1. Go to [**pearsonITcertification.com/account**](#) and log in or create a new account.
2. Enter the ISBN: **9780137941582** (NOTE: Please enter the print book ISBN provided to register the eBook you purchased.)
3. Answer the challenge question as proof of purchase.
4. Click on the “Access Bonus Content” link in the Registered Products section of your account page, to be taken to the page where your downloadable content is available.

This eBook version of the print title does not contain the practice test software that accompanies the print book.

You May Also Like—Premium Edition eBook and Practice Test.
To learn about the Premium Edition eBook and Practice Test
series, visit pearsonITcertification.com/practicetest

The Professional and Personal Technology Brands of Pearson



Cisco Press

informIT

PEARSON IT Certification

QUE

SAMS

Special Offer

Save 80% on Premium Edition eBook and Practice Test

The *AWS Certified Solutions Architect - Associate (SAA-C03) Cert Guide Premium Edition and Practice Test* provides PDF and EPUB eBook files to read on your preferred device and an enhanced edition of the Pearson Test Prep practice test software. You also receive two additional practice exams with links for every question mapped to the PDF eBook.

**AWS Certified
Solutions
Architect - Associate
(SAA-C03) Cert Guide**

Access interactive study tools on this book's companion website, including practice test software, review questions, Key Term flash card application, a study planner, and more!

To access the companion website, simply follow these steps:

1. Go to [**www.pearsonITcertification.com/register**](http://www.pearsonITcertification.com/register).
2. Enter the print book ISBN: **9780137941582**.
3. Answer the security question to validate your purchase.
4. Go to your account page.
5. Click on the **Registered Products** tab.
6. Under the book listing, click on the Access Bonus Content link.

If you have any issues accessing the companion website, you can contact our support team by going to
[**http://pearsonitp.echelp.org**](http://pearsonitp.echelp.org).

Pearson Test Prep online system requirements:

Browsers: Chrome version 73 and above; Safari version 12 and above; Microsoft Edge 44 and above.

Devices: Desktop and laptop computers, tablets running Android v8.0 and above or iPadOS v13 and above, smartphones running Android v8.0 and above or iOS v13 and above with a minimum screen size of 4.7". Internet access required.

Pearson Test Prep offline system requirements:

Windows 10, Windows 8.1; Microsoft .NET Framework 4.5 Client; Pentium-class 1 GHz processor (or equivalent); 512 MB RAM; 650 MB disk space plus 50 MB for each downloaded practice exam; access to the Internet to register and download exam databases.

Code Snippets

Many titles include programming code or configuration examples. To optimize the presentation of these elements, view the eBook in single-column, landscape mode and adjust the font size to the smallest setting. In addition to presenting code and configurations in the reflowable text format, we have included images of the code that mimic the presentation found in the print book; therefore, where the reflowable format may compromise the presentation of the code listing, you will see a “Click here to view code image” link. Click the link to view the print-fidelity code image. To return to the previous page viewed, click the Back button on your device or app.

```
1.{  
2. "Version": "2012-10-17",  
3. "Statement": {  
4. "Effect": "Allow",  
5. "Action": "s3>ListBucket",  
6. "Resource": "arn:aws:s3:::graphic_bucket"  
7. }  
8. }
```

```
"Statement": [
{
  "Sid": "AllowUsersToPerformUserActions",
  "Effect": "Allow",
  "Action": [
    "iam>ListPolicies",
    "iam>GetPolicy",
    "iam>UpdateUser",
    "iam>AttachUserPolicy",
    "iam>ListEntitiesForPolicy",
    "iam>DeleteUserPolicy",
    "iam>DeleteUser",
    "iam>ListUserPolicies",
    "iam>CreateUser",
    "iam>RemoveUserFromGroup",
    "iam>AddUserToGroup",
    "iam> GetUserPolicy",
    "iam>ListGroupsForUser",

    "iam>PutUserPolicy",
    "iam>ListAttachedUserPolicies",
    "iam>ListUsers",
    "iam> GetUser",
    "iam>DetachUserPolicy"
  ]
},
```

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:*",  
        "ec2:*"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "iam:DeleteUser",  
         "Resource": "*"},  
        {"Condition": {"StringLike": {"iam:ResourceTag/temp_user": "can_terminate"}}}  
    ]  
}  
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:*",  
            "Resource": "*"},  
        {"Condition": {"StringEquals": {"aws:PrincipalTag/useradmin": "true"}}}  
    ]  
}
```

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Principal": {"AWS": "arn:iam::123456789:root"},  
        "Action": "sts:AssumeRole",  
    }  
}
```

```
Statement": [
  {
    "Effect": "Allow",
    "Action": "s3>ListAllMyBuckets",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3>ListBucket",
      "s3>GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::corpdocs"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3>GetObject",
      "s3>PutObject",
      "s3>DeleteObject"
    ],
    "Resource": "arn:aws:s3:::corpdocs/*"
  }
]
```

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "sts:AssumeRole",  
        "Resource": "arn:aws:iam::::PRODUCTION-AWS-ACCT-ID:role/  
get-access"  
    }  
}
```

```
"Version": "2012-10-17",
"Statement": [
{
    "Sid": " Controlled Admin Tasks",
    "Effect": "Allow",
    "Action": [
        "rds>CreateDBSnapshot",
        "rds:StopDBInstance",
        "rds:StartDBInstance"
    ],
    "Resource": [
        "arn:aws:rds:[AWS_region]:[_AWS_account_id]:snapshot:*",
        "arn:aws:rds:[AWS_region]:[_AWS_account_id]:db:demoDB"
    ]
},
{
    "Sid": "DescribeInstances",
    "Effect": "Allow",
    "Action": "rds:DescribeDBInstances",
    "Resource": "*"
}
]
```

```
{  
    "Version": "2012-10-17",  
    "Id": "S3PolicyId1",  
    "Statement": [  
        {  
            "Sid": "IPAllow",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3:::2021232reports",  
                "arn:aws:s3:::2021232reports/*"  
            ],  
            "Condition": {  
                "NotIpAddress": {"aws:SourceIp": "54.242.144.0/24"}  
            }  
        }  
    ]  
}
```

```
{  
    "AWSTemplateFormatVersion" : "2022-09-09",  
    "Description": "EC2 instance",  
    "Resources": {  
        "EC2Instance" : {  
            "Type" : "AWS::EC2::Instance",  
            "Properties": {  
                "ImageId" : "ami-0ff8a91497e77f667",  
                "InstanceType" : "t1.micro"  
            }  
        }  
    }  
}
```

```
AWSTemplateFormatVersion: '2022-09-09'  
Description: EC2 instance  
Resources:  
EC2Instance:  
Type: AWS::EC2::Instance  
Properties:  
ImageId: ami-0ff8a91497e77f667
```

```
"AWSTemplateFormatVersion": "version date",  
"AWSTemplateFormatVersion": "2022-09-09"  
<TemplateFormatVersion: Defines the current CF template version>  
  
"Description": "Here are the additional details about this template  
and what it does",  
<Description: Describes the template: must always follow the version  
section>  
  
"Metadata": {
```

```

"Metadata" : {
    "Instances" : {"Description : "Details about the instances"},
    "Databases": {"Description: "Details about the databases"}
}
},
<Metadata: Additional information about the resources being deployed by the template>

"Parameters": {
    "InstanceTypeParameter" : {
        "Type": "String",
        "Default" : "t2.medium",
        "AllowedValues" : ["t2.medium", "m5.large", "m5.xlarge"],
        "Description" : "Enter t2.medium, m.5large, or m5.xlarge. Default is t2.medium."
    }
},
<Parameters: Defines the AWS resource values allowed to be selected and used by your template>

"Mappings": {
    "RegionMap" : [
        "us-east-1      : { "HVM64 : "ami-0bb8a91508f77f868"},,
        "us-west-1      : { "HVM64 : "ami-0cdb828fd58c52239"},,
        "eu-west-1      : { "HVM64 : "ami-078bb4163c506cd88"},,
        "us-southeast-1 : { "HVM64 : "ami-09999b978cc4dfc10"},,
        "us-northeast-1 : { "HVM64 : "ami-06fd42961cd9f0d75"}]
}

```

```
<Mappings: Defines conditional parameters defined by a "key"; in this example, the AWS region and a set of AMI values to be used>
```

```
    "Conditions": {  
        "CreateTestResources": {"Fn::Equals" : [{"Ref" : "EnvType"},  
        "test"]}  
    },
```

```
<Conditions: Defines dependencies between resources, such as the order when resources are created or where resources are created. For example, "test" deploys the stack in the test environment>
```

```
AWSTemplateFormatVersion: 2022-09-09
```

```
Description: EC2 Instance Template
```

```
 "Resources": {  
     "EC2Machine": {  
         "Type": "AWS::EC2::Instance",  
         "Properties": {  
             "ImageId": "i-0ff407a7042afb0f0",  
             "NetworkInterfaces": [{  
                 "DeviceIndex": "0",  
                 "DeleteOnTermination": "true",  
                 "SubnetId": "subnet-7c6dd651"  
             }]  
             "InstanceType": "t2.small"  
         }  
     },  
     "EIP": {  
         "Type": "AWS::EC2::EIP",  
         "Properties": {  
             "Domain": "VPC"  
         }  
     },  
 },
```

```
"VpcIPAssoc": {  
    "Type": "AWS::EC2::EIPAssociation",  
    "Properties": {  
        "InstanceId": {  
            "Ref": "EC2Machine"  
        },  
        "AllocationId": {  
            "Fn::GetAtt": ["EIP",  
                "AllocationId"]  
        }  
    }  
}
```

```
curl https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_
linux/amd64/latest/amazon-cloudwatch-agent.rpm -O
sudo yum install -y ./amazon-cloudwatch-agent.rpm
sudo /opt/aws/amazon-cloudwatch-agent/bin/
amazon-cloudwatch-agent-config-wizard
AWS autoscaling put-lifecycle-hook --lifecycle-hook-name <lifecycle
code> --auto-scaling-group-name <ASG here > --lifecycle-transition
autoscaling:EC2_INSTANCE_LAUNCHING
```

```
{  
    "Statement": [  
        {  
            "Resource": "http://*",  
            "Condition": {  
                "IpAddress": {  
                    "AWS:SourceIp": "192.0.4.0/32"  
                },  
                "DateGreaterThan": {  
                    "AWS:EpochTime": 1367034400  
                },  
                "DateLessThan": {  
                    "AWS:EpochTime": 1367120800  
                }  
            }  
        ]  
    }  
}
```