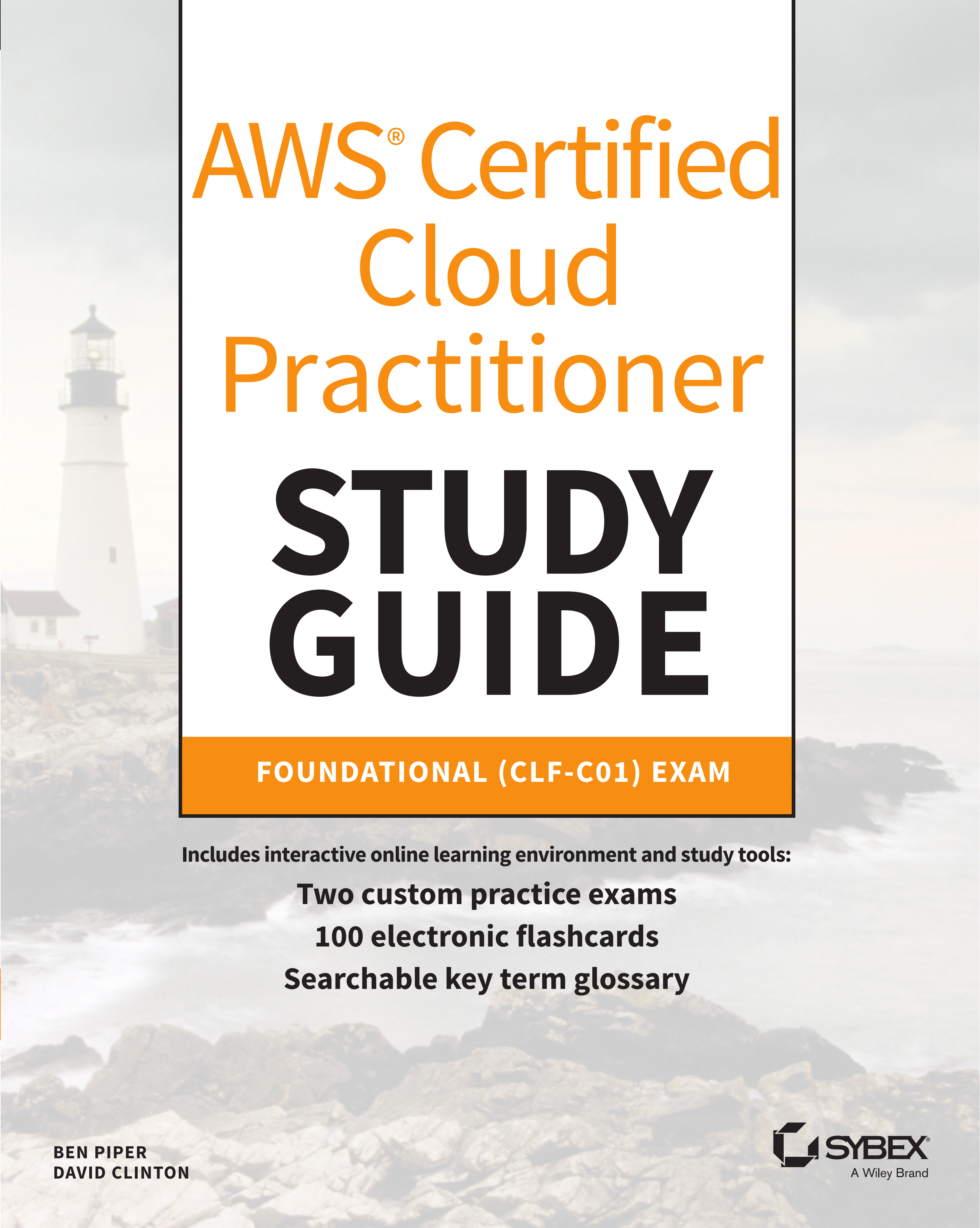

AWS® Certified Cloud Practitioner

STUDY GUIDE

FOUNDATIONAL (CLF-C01) EXAM

Includes interactive online learning environment and study tools:

Two custom practice exams

100 electronic flashcards

Searchable key term glossary

BEN PIPER
DAVID CLINTON

SYBEX
A Wiley Brand

define in advance is a primary key to uniquely identify each item. For example, to store customer data in a table, you might use a unique customer ID number as the primary key.

In exchange for the flexibility of storing unstructured data, the types of queries you can perform against that data are more limited. Nonrelational databases are designed to let you query items based on the primary key. Because the rest of the data doesn't follow a predictable structure, a particular piece of data could be anywhere in the database. Hence, trying to query against any other data requires searching through every item in the entire table—a process that gets slower as the table grows. Nonrelational databases are best suited for applications that need to perform just a few well-defined queries.

Amazon Relational Database Service

The Amazon Relational Database Service (RDS) is Amazon's managed relational database service. RDS lets you provision a number of popular relational database management systems (RDBMSs) including Microsoft SQL Server, Oracle, MySQL, and PostgreSQL.

You can always install and configure your own database server on an EC2 instance. But RDS offers several advantages over this. When you create an RDS database instance, Amazon sets up one or more compute instances and takes care of installing and configuring the RDBMS of your choice. These compute instances are not EC2 instances that you can secure shell (SSH) into, but they are connected to a virtual private cloud (VPC) of your choice, allowing your applications running on AWS or on-premises to take full advantage of an RDS-hosted database. Like EC2 instances, RDS instances use Elastic Block Service (EBS) volumes for storage.

To achieve the level of performance and availability you need, you can choose a multi-Availability Zone (multi-AZ) deployment to have database instances in multiple Availability Zones. RDS can also perform manual or automatic EBS snapshots that you can easily restore to new RDS instances. RDS can also handle the hard work of installing patches and upgrades during scheduled maintenance windows.

Database Engines

When you create an RDS instance, you must choose a database engine, which is the specific RDBMS that will be installed on your instance. You can have only one database engine per instance, but you can provision multiple instances if need be. Amazon RDS supports the following six database engines:

- MySQL
- MariaDB
- Oracle
- PostgreSQL
- Microsoft SQL Server
- Amazon Aurora

With the exception of Amazon Aurora, these database engines are either open source or commercially available products found in many data center environments. Amazon Aurora is a proprietary database designed for RDS, but it's compatible with existing MySQL and PostgreSQL databases. Being able to use RDS to deploy an RDBMS that you're already familiar with makes migrating such databases from on-premises to RDS much easier.

Licensing

Depending on the database engine you choose, you must choose one of two licensing options: *license included* or *bring your own license* (BYOL):

License included The license is included in the pricing for each RDS instance. The Microsoft SQL Server and Oracle database engine options offer this license model. The free database engines—MariaDB, MySQL, and PostgreSQL—exclusively use the license included model.

Bring your own license In this model, you must provide your own license to operate the database engine you choose. Unlike the license-included option, licensing costs are not built into RDS pricing. This model is currently available only for Oracle databases.

Instance Classes

Implementing a relational database—even one backed by RDS—requires some capacity planning to ensure the database gives you the level of availability and performance your application needs.

When you deploy an RDS instance, you must choose a database instance class that defines the number of virtual CPUs (vCPU), the amount of memory, and the maximum network and storage throughput the instance can support. There are three instances classes you can choose from: Standard, Memory Optimized, and Burstable Performance.

Standard

The Standard instance class will meet the requirements of most applications. The latest-generation Standard instance class offers the following specs:

- Between 2 and 96 vCPU
- 8–384 GB memory

Memory Optimized

The Memory Optimized instance class is for applications with the most demanding database requirements. This class offers the most disk throughput and network bandwidth. The latest-generation instance class provides the following:

- Between 4 and 128 vCPU
- 122–3,904 GB memory

Database instances use EBS storage. Both the Standard and Memory Optimized instance class types are EBS-optimized, meaning they provide dedicated bandwidth for transfers to and from EBS storage.

Burstable Performance

The Burstable Performance instance class is for nonproduction databases that have minimal performance requirements, such as those for test and development purposes. The latest-generation Burstable Performance instance class has the lowest network bandwidth and disk throughput and offers the following:

- Between 2 and 8 vCPU
- 1–32 GB memory

It can be difficult to predict exactly how many RDS instances you need and how much compute power, memory, and network and storage throughput each of those instances needs. Thankfully, RDS makes it easy to right-size your database deployments in two ways: scaling vertically and scaling horizontally.

Scaling Vertically

Scaling vertically refers to changing the way resources are allocated to a specific instance. After creating an instance, you can scale up to a more powerful instance class to add more memory or improve computing or networking performance. Or you can scale down to a less powerful class to save on costs.

Storage

The level of performance an RDS instance can achieve depends not only on the instance class you choose but also on the type of storage. New RDS instances use EBS volumes, and the maximum throughput a volume can achieve is a function of both the instance class and the number of *input/output operations per second* (IOPS) the EBS volume supports. IOPS measure how fast you can read from and write to a volume. Higher IOPS generally means faster reads and writes. RDS offers three types of storage: general-purpose SSD, provisioned IOPS SSD, and magnetic.

General-Purpose SSD

General-purpose SSD storage is good enough for most databases. You can allocate a volume of between 20 GB and 32 TB. The number of IOPS per volume depends on how much storage you allocate. The more storage you allocate, the better your read and write performance will be.

If you're not sure how much storage to provision, don't worry. General-purpose SSD volumes can temporarily achieve a higher number of IOPS through a process called *bursting*. During spikes of heavy read or write activity, bursting will kick in automatically and give your volume an added performance boost. This way, you don't have to allocate an excessive amount of storage just to get enough IOPS to meet peak demand.

Provisioned IOPS SSD

Provisioned IOPS SSD storage allows you to specify the exact number of IOPS (in thousands) that you want to allocate per volume. Like general-purpose SSD storage, you can allocate up to 32 TB. But unlike general-purpose SSD storage, provisioned IOPS SSD storage doesn't offer bursting, so it's necessary to decide beforehand the maximum number of IOPS you'll need. However, even if your needs change, you can always adjust the number of IOPS later.

Magnetic

Magnetic storage is available for backward compatibility with legacy RDS instances. Unlike the other storage options, it doesn't use EBS, and you can't change the size of a magnetic volume after you create it. Magnetic volumes are limited to 4 TB in size and 1,000 IOPS.

You can increase the size of an EBS volume after creating it without causing an outage or degrading performance. You can't, however, decrease the amount of storage allocated, so be careful not to go overboard.

You can also migrate from one storage type to another, but doing so can result in a short outage of typically a few minutes. But when migrating from magnetic to EBS storage, the process can take up to several days. During this time, the instance is still usable but may not perform optimally.

Scaling Horizontally with Read Replicas

In addition to scaling up by choosing a more powerful instance type or selecting high-IOPS storage, you can improve the performance of a database-backed application by adding additional RDS instances that perform only reads from the database. These instances are called *read replicas*.

In a relational database, only the master database instance can write to the database. A read replica helps with performance by removing the burden of read-only queries from the master instance, freeing it up to focus on writes. Hence, read replicas provide the biggest benefit for applications that need to perform a high number of reads. Read replicas are also useful for running computationally intensive queries, such as monthly or quarterly reports that require reading and processing large amounts of data from the database.

High Availability with Multi-AZ

Even if you use read replicas, only the master database instance can perform writes against your database. If that instance goes down, your database-backed application won't be able to write data until it comes back online. To ensure that you always have a master database instance up and running, you can configure high availability by enabling the multi-AZ feature on your RDS instance.

With multi-AZ enabled, RDS creates an additional instance called a *standby database instance* that runs in a different Availability Zone than your primary database instance. The primary instance instantly or synchronously replicates data to the secondary instance, ensuring that every time your application writes to the database, that data exists in multiple Availability Zones.

If the primary fails, RDS will automatically fail over to the secondary. The failover can result in an outage of up to two minutes, so your application will experience some interruption, but you won't lose any data.

With multi-AZ enabled, you can expect your database to achieve a monthly availability of 99.95 percent. It's important to understand that an instance outage may occur for reasons other than an Availability Zone outage. Routine maintenance tasks such as patching or upgrading the instance can result in a short outage and trigger a failover.

If you use the Amazon Aurora database engine—Amazon's proprietary database engine designed for and available exclusively with RDS—you can take advantage of additional benefits when using multi-AZ. When you use Aurora, your RDS instances are part of an Aurora cluster. All instances in the cluster use a shared storage volume that's synchronously replicated across three different Availability Zones. Also, if your storage needs increase, the cluster volume will automatically expand up to 64 TB.

Backup and Recovery

Whether or not you use multi-AZ, RDS can take manual or automatic EBS snapshots of your instances. Snapshots are stored across multiple Availability Zones. If you ever need to restore from a snapshot, RDS will restore it to a new instance. This makes snapshots useful not only for backups but also for creating copies of a database for testing or development purposes.

You can take a manual snapshot at any time. You can configure automatic snapshots to occur daily during a 30-minute backup window. RDS will retain automatic snapshots between 1 day and 35 days, with a default of 7 days. Manual snapshots are retained until you delete them.

Enabling automatic snapshots also enables point-in-time recovery, a feature that saves your database change logs every 5 minutes. Combined with automated snapshots, this gives you the ability to restore a failed instance to within 5 minutes before the failure—losing no more than 5 minutes of data.

Determining Your Recovery Point Objective

Do you need snapshots *and* multi-AZ? It's important to understand that although both snapshots and multi-AZ protect your databases, they serve slightly different purposes. Snapshots are good for letting you restore an entire database instance. If your database encounters corruption, such as malicious deletion of records, snapshots let you recover that data, even if the corruption occurred days ago (provided you're retaining the snapshots). Multi-AZ is designed to keep your database up and running in the event of an instance failure. To achieve this, data is synchronously replicated to a secondary instance.

How much data loss you can sustain in the event of a failure is called the *recovery point objective* (RPO). If you can tolerate losing an hour's worth of data, then your RPO would be 1 hour. To achieve such an RPO, simply using automatic snapshots with point-in-time recovery is sufficient. For an RPO of less than 5 minutes, you would also want to use multi-AZ to synchronously replicate your data to a secondary instance.

DynamoDB

DynamoDB is Amazon's managed nonrelational database service. It's designed for highly transactional applications that need to read from or write to a database tens of thousands of times a second.

Items and Tables

The basic unit of organization in DynamoDB is an item, which is analogous to a row or record in a relational database. DynamoDB stores items in tables. Each DynamoDB table is stored across one or more partitions. Each partition is backed by solid-state drives, and partitions are replicated across multiple Availability Zones in a region, giving you a monthly availability of 99.99 percent.

Each item must have a unique value for the primary key. An item can also consist of other key-value pairs called *attributes*. Each item can store up to 400 KB of data, more than enough to fill a book! To understand this better, consider the sample shown in Table 9.2.

TABLE 9.2 A Sample DynamoDB Table

Username (Primary Key)	LastName	FirstName	FavoriteColor
hburger	Burger	Hamilton	
dstreet	Street	Della	Fuchsia
pdrake	Drake	Paul	Silver
perry		Perry	

Username, LastName, FirstName, and FavoriteColor are all attributes. In this table, the Username attribute is the primary key. Each item must have a value for the primary key, and it must be unique within the table. Good candidates for primary keys are things that tend to be unique, such as randomly generated identifiers, usernames, and email addresses.

Exam Essentials

Understand the major differences between relational and nonrelational databases. Relational databases are designed for structured data that contains a defined number of attributes per record. They let you perform complex queries against a variety of dimensions, making them ideal for reporting and analytics. Nonrelational databases are designed for data that doesn't follow a predictable structure. Each item in a nonrelational database must have a primary key, and you can query based on that key.

Know the vertical and horizontal scaling options for RDS. You can scale an RDS instance vertically by upgrading to a larger instance class to give it more processing power, memory, or disk or network throughput. You can also select provisioned IOPS SSD storage to ensure your instance always achieves the storage performance it needs. For horizontal scaling of reads, your only option is to use read replicas.

Be able to describe the components of RDS. An RDS deployment consists of at least one instance. You must select an instance class that defines the vCPUs and memory for the instance. You must also select a database engine. For storage, you must select general-purpose or provisioned IOPS SSD. Magnetic storage is a legacy option that's not available for new deployments. You can also add read replicas to scale horizontally to improve read performance. In a multi-AZ deployment, you can add additional secondary instances that the primary synchronously replicates data to.

Know the backup and recovery options for RDS. You can schedule automatic snapshots for your RDS instance to occur daily during a 30-minute backup window of your choice. Backups are retained between 1 day and 35 days. Enabling automatic backups also enables point-in-time recovery, allowing the restoration of a failed database up to 5 minutes prior to failure. Restoring from a snapshot entails creating a new instance from the snapshot. You can also take a manual snapshot at any time.

Understand how DynamoDB stores data. DynamoDB stores data as items in tables. Each item must have primary key whose values are unique within the table. This is how DynamoDB uniquely identifies an item. The primary key's name and data type must be defined when the table is created. When you create an item, you can also add other attributes in addition to the primary key. DynamoDB uses the primary key to distribute items across different partitions. The number of partitions allocated to a table depends on the number of WCU and RCU you configure.

Be able to identify scenarios for using Redshift. Redshift is a data-warehousing service for storing and analyzing structured data from multiple sources, including relational databases and S3. Redshift can store much more data than RDS, up to 2 PB!