



Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut,  
Kevin E. Kelly, Sean Senior, John Stamper

# AWS Certified Solutions Architect

## OFFICIAL STUDY GUIDE

**ASSOCIATE EXAM**

Covers exam objectives, including designing highly available, cost efficient, fault tolerant, scalable systems, implementation and deployment, data security, troubleshooting, and much more...

Includes interactive online learning environment and study tools with:

- + 2 custom practice exams
- + More than 100 electronic flashcards
- + Searchable key term glossary



# Object Storage versus Traditional Block and File Storage

In traditional IT environments, two kinds of storage dominate: *block storage* and *file storage*. Block storage operates at a lower level—the raw storage device level—and manages data as a set of numbered, fixed-size blocks. File storage operates at a higher level—the operating system level—and manages data as a named hierarchy of files and folders. Block and file storage are often accessed over a network in the form of a Storage Area Network (SAN) for block storage, using protocols such as iSCSI or Fibre Channel, or as a Network Attached Storage (NAS) file server or “filer” for file storage, using protocols such as Common Internet File System (CIFS) or Network File System (NFS). Whether directly-attached or network-attached, block or file, this kind of storage is very closely associated with the server and the operating system that is using the storage.

Amazon S3 object storage is something quite different. Amazon S3 is cloud *object storage*. Instead of being closely associated with a server, Amazon S3 storage is independent of a server and is accessed over the Internet. Instead of managing data as blocks or files using SCSI, CIFS, or NFS protocols, data is managed as objects using an Application Program Interface (API) built on standard HTTP verbs.

Each Amazon S3 object contains both data and metadata. Objects reside in containers called *buckets*, and each object is identified by a unique user-specified key (filename). Buckets are a simple flat folder with no file system hierarchy. That is, you can have multiple buckets, but you can’t have a sub-bucket within a bucket. Each bucket can hold an unlimited number of objects.

It is easy to think of an Amazon S3 object (or the data portion of an object) as a file, and the key as the filename. However, keep in mind that Amazon S3 is not a traditional file system and differs in significant ways. In Amazon S3, you GET an object or PUT an object, operating on the whole object at once, instead of incrementally updating portions of the object as you would with a file. You can’t “mount” a bucket, “open” an object, install an operating system on Amazon S3, or run a database on it.

Instead of a file system, Amazon S3 is highly-durable and highly-scalable object storage that is optimized for reads and is built with an intentionally minimalistic feature set. It provides a simple and robust abstraction for file storage that frees you from many underlying details that you normally do have to deal with in traditional storage. For example, with Amazon S3 you don’t have to worry about device or file system storage limits and capacity planning—a single bucket can store an unlimited number of files. You also don’t need to worry about data durability or replication across availability zones—Amazon S3 objects are automatically replicated on multiple devices in multiple facilities within a region. The same with scalability—if your request rate grows steadily, Amazon S3 automatically partitions buckets to support very high request rates and simultaneous access by many clients.



If you need traditional block or file storage in addition to Amazon S3 storage, AWS provides options. The Amazon EBS service provides block level storage for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon Elastic File System (AWS EFS) provides network-attached shared file storage (NAS storage) using the NFS v4 protocol.