Using Amazon Web Services

Amazon.com is one of the most important and heavily trafficked Web sites in the world. It provides a vast selection of products using an infrastructure based on Web services. As Amazon.com has grown, it has dramatically grown its infrastructure to accommodate peak traffic times. Over time the company has made its network resources available to partners and affiliates, which also has improved its range of products.

Starting in 2006, Amazon.com made its Web service platform available to developers on a usage-basis model. Through hardware virtualization on Xen hypervisors, Amazon. com has made it possible to create private virtual servers that you can run worldwide. These servers can be provisioned with almost any kind of application software you might envisage, and they tap into a range of support services that not only make distributed cloud computing applications possible , but make them robust. Some very large Web sites are running on Amazon. com's infrastructure without their client audience being any the wiser.

Amazon Web Services is based on SOA standards, including HTTP, REST, and SOAP transfer protocols, open source and commercial operating systems, application servers, and browser-based access. Virtual private servers can provision virtual private clouds connected through virtual private networks providing for reasonable security and control by the system administrator.

AWS has a great value proposition: You pay for what you use. While you may not save a great deal of money over time using AWS for enterprise class Web applications, you encounter very little barrier to entry in terms of getting your site or application up and running quickly and robustly. AWS has much to teach us about the future of cloud computing and how virtual infrastructure can be best leveraged as a business asset.

Understanding Amazon Web Services

The Amazon is the world's largest river. Amazon.com is the world's largest online retailer with netsales in $24.51 billion, according to their 2009 annual report. The company is a long way past selling books and records. While Amazon.com is not the earth's biggest retailer (that spot is reserved for Wal-Mart), Amazon.com offers the largest number of retail product SKUs through a large eco- system of partnerships. By any measure, Amazon.com is a huge business. To support this business,Amazon.com has built an enormous network of IT systems to support not only average, but peak customer demands. Amazon Web Services (AWS) takes what is essentially unused infrastructure capacity on Amazon.com's network and turns it into a very profitable business.

AWS is having enormous impact in cloud computing. Indeed, Amazon.com's services represent the largest pure Infrastructure as a Service (IAAS) play in the marketplace today. It is also one of the best examples of what is possible using a Service Oriented Architecture (SOA). The structure of Amazon.com's Amazon Web Services (AWS) is thereforehighly educational in understanding just how disruptive cloud computing can be to traditional fixed asset IT deployments, how virtualization enables a flexible approach to system rightsizing, and how dispersed systems can impart reliability to mission critical systems.

Amazon Web Service Components and Services

Amazon Web Services is comprised of the following components, listed roughly in their order ofimportance:

1 **Amazon Elastic Compute Cloud** (EC2; `http://aws.amazon.com/ec2/`), is the central application in the AWS portfolio. It enables the creation, use, and management of virtual private servers running the Linux or Windows operating system over a Xen hypervisor. Amazon Machine Instances are sized at various levels and rented on a computing/ hour basis. Spread over data centers worldwide, EC2 applications may be created that are highly scalable, redundant, and fault tolerant. EC2 is described more fully the nextsection. A number of tools are used to support EC2 services:

Amazon Simple Queue Service (SQS; `http://aws.amazon.com/sqs/`) is a message queue or transaction system for distributed Internet-based applications. See "Examining the Simple Queue Service (SQS)" later in this chapter for a description ofthis AWS feature. In a loosely coupled SOA system, a transaction manager is requiredto ensure that messages are not lost when a component isn't available.

Amazon Simple Notification Service (SNS; `http://aws.amazon.com/sns/`) isa Web service that can publish messages from an application and deliver them to other applications or to subscribers. SNS provides a method for triggering actions, allowing clients or applications to subscribe to information (like RSS), or polling for new or changed information or perform updates.

EC2 can be monitored by **Amazon CloudWatch** (`http://aws.amazon.com/ cloudwatch/`), which provides a console or command line view of resource utiliza- tion, site Key Performance Indexes (performance metrics), and operational indicators for factors such as processor demand, disk utilization, and network I/O. The metrics obtained by CloudWatch may be used to enable a feature called **Auto Scaling** (`http://aws.amazon.com/autoscaling/`) that can automatically scale an EC2 site based on a set of rules that you create. Autoscaling is part of Amazon Cloudwatch and available at no additional charge.

Amazon Machine Instances (AMIs) in EC2 can be load balanced using the **Elastic Load Balancing** (`http://aws.amazon.com/elasticloadbalancing/`) feature. The Load Balancing feature can detect when an instance is failing and reroute traffic to a healthy instance, even an instance in other AWS zones. The Amazon CloudWatch metrics request count and request latency that show up in the AWS console are used to support Elastic Load Balancing.

Amazon Simple Storage System (S3; `http://aws.amazon.com/s3/`**)** is an online backup and storage system. A high speed data transfer feature called AWS Import/Export (`http://aws.amazon.com/importexport/`) can transfer data to and from AWS using Amazon's own internal network to portable storage devices.

Amazon Elastic Block Store (EBS; `http://aws.amazon.com/ebs/`**)** is a system for creating virtual disks (volume) or block level storage devices that can be used for AmazonMachine Instances in EC2.

Amazon SimpleDB (`http://aws.amazon.com/simpledb/`) is a structured data store that supports indexing and data queries to both EC2 and S3. SimpleDB isn't a full database implementation, as you learn in "Exploring SimpleDB (S3)" later in this chapter;it stores data in "buckets" and without requiring the creation of a database schema. This design allows SimpleDB to scale easily. SimpleDB interoperates with both Amazon EC2and Amazon S3.

Amazon Relational Database Service (RDS; `http://aws.amazon.com/rds/`) allows you to create instances of the MySQL database to support your Web sites and the many applications that rely on data-driven services. MySQL is the "M" in the ubiquitous LAMP Web services platform (for Linux, APACHE, MySQL, and PERL), and the inclusion of this service allows developers to port applications, their source code, and databases directly over to AWS, preserving their previous investment in these technologies. RDS provides features such as automated software patching, database backups, and automated database scaling via an API call.

Amazon Cloudfront (`http://aws.amazon.com/cloudfront/`) is an edge-storage or content-delivery system that caches data in different physical locations so that user access to data is enhanced through faster data transfer speeds and lower latency. Cloudfront is similar to systems such as Akamai.com, but is proprietary to Amazon.com and is set up to work with Amazon Simple Storage System (Amazon S3).

While the list above represents the most important of the AWS offerings, it is only a partial list—a list that is continually growing and very dynamic. A number of services and utilities support Amazon partners or the AWS infrastructure itself. These are the ones you may encounter:

Alexa Web Information Service (`http://aws.amazon.com/awis/`) and **Alexa TopSites** (`http://aws.amazon.com/alexatopsites/`) are two services that collect and expose information about the structure and traffic patterns of Web sites. This infor- mation can be used to build or structure Web sites, access related sites, analyze historical patterns for growth and relationships, and perform data analysis on site information. Alexa Top Sites can rank sites based on their usage and be used to structure awareness ofsite popularity into the structure of Web service you build.

Amazon Associates Web Services (A2S) is the machinery for interacting with Amazon'svast product data and eCommerce catalog function. This service, which was called Amazon E-Commerce Service (ECS), is the means for vendors to add their products to theAmazon.com site and take orders and payments.

Amazon DevPay (`http://aws.amazon.com/devpay/`) is a billing and account mangement service that can be used by businesses that run applications on top of AWS. DevPay provides a developer API that eliminates the need for application developers to build order pipelines, because Amazon does the billing based on your prices and then uses Amazon Payments to collect the payments.

Amazon Elastic MapReduce (`http://aws.amazon.com/elasticmapreduce/`) is an interactive data analysis tool for performing indexing, data mining, file analysis, log file analysis, machine learning, financial analysis, and scientific and bioinformatics research. Elastic MapReduce is built on top of a Hadoop framework using the Elastic Compute Cloud (EC2) and Simple Storage Service (S3).

Amazon Mechanical Turk (`http://aws.amazon.com/mturk/`) is a means for accessing human researchers or consultants to help solve problems on a contractual or temporary basis. Problems solved by this human workforce have included object identifi- cation, video or audio recording, data duplication, and data research. Amazon.com calls this type of work Human Intelligence Tasks (HITs). The Mechanical Turk is currently in beta.

AWS Multi-Factor Authentication (AWS MFA; `http://aws.amazon.com/mfa/`**)** isa special feature that uses an authentication device you have in your possession to provideaccess to your AWS account settings. This hardware key generates a pseudo-random six- digit number when you press a button that you enter into your logon. This gives you two layers of protection: your user id and password (things you know) and the code from yourhardware key (something you have). This multifactor security feature can be extended to Cloudfront and Amazon S3.

Amazon Flexible Payments Service (FPS; `http://aws.amazon.com/fps/`) is a payments-transfer infrastructure that provides access for developers to charge Amazon's customers for their purchases. Using FPS, goods, services, donations, money transfers, and recurring payments can be fulfilled. FPS is exposed as an API that sorts transactions into packages called Quick Starts that make this service easy to implement.

Amazon Fulfillment Web Services (FWS; `http://aws.amazon.com/fws/`) allows merchants to fill orders through Amazon.com fulfillment service, with Amazon handlingthe physical delivery of items on the merchant's behalf. Merchant inventory is prepositioned in Amazon's fulfillment centers, and Amazon packs and ships the items.

Amazon Virtual Private Cloud (VPC; `http://aws.amazon.com/vpc/`) provides a bridge between a company's existing network and the AWS cloud. VPC connects your network resources to a set of AWS systems over a Virtual Private Network (VPN) connection and extends security systems, firewalls, and management systems to include their provisioned AWS servers. Amazon VPC is integrated with Amazon EC2, but Amazon plans to extend the capabilities of VPC to integrate with other systems in the Amazon cloud computing portfolio.

AWS Premium Support (`http://aws.amazon.com/premiumsupport/`) is Amazon's technical support and consulting business. Through AWS Premium Support, subscribers to AWS can get help building or supporting applications that use EC2, S3, Cloudfront, VPC, SQS, SNS, SimpleDB, RDS, and the other services listed above. Service plans are available on a per-incidence, monthly, or unlimited basis at different levels of service. With this overview of AWS components complete, let's look at the central part of Amazon Web Service's value proposition, the creation and deployment of virtual private servers using the Elastic Compute Cloud (EC2) service.

Working with the Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2) is a virtual server platform that allows users to create and run virtual machines on Amazon's server farm. With EC2, you can launch and run server instancescalled Amazon Machine Images (AMIs) running different operating systems such as Red Hat Linux and Windows on servers that have different performance profiles. You can add or subtract virtual servers elastically as needed; cluster, replicate, and load balance servers; and locate your different servers in different data centers or "zones" throughout the world to provide fault tolerance. The term *elastic* refers to the ability to size your capacity quickly as needed.

The difference between an instance and a machine image is that an instance is the emulation of a hardware platform such as X86, IA64, and so on running on the Xen hypervisor. A machine image is the software and operating system running on top of the instance. A machine image may be thought of as the contents of a boot drive, something that you could package up with a program such as Ghost, Acronis, or TrueImage to create a single file containing the exact contents of a volume. A machine image should be composed of a hardened operating system with as few features and capabilities as possible and locked down as much as possible.

Consider a situation where you want to create an Internet platform that provides the following:

- A high transaction level for a Web application
- A system that optimizes performance between servers in your system
- Data driver information services
- Network security
- The ability to grow your service on demand

Implementing that type of service might require a rack of components that included the following:

- An application server with access to a large RAM allocation
- A load balancer, usually in the form of a hardware appliance such as F5's BIG-IP
- A database server
- Firewalls and network switches
- Additional rack capacity at the ISP

A physical implementation of these components might cost you something in the neighborhood of $25,000 depending upon the scale of your application. With AWS, you might be able to have an equivalent service for as little as $1,000 and have a high level of availability and reliability to boot. This difference may surprise you, but it is understandable when you consider that AWS can run its services with a much greater efficiency than your company would alone and therefore amortize its investment in hardware over several customers. That is the promise and the potential of cloud computing realized and why large Web sites such as Recovery.gov have moved to AWS.

Amazon Machine Images

AMIs are operating systems running on the Xen virtualization hypervisor. Each virtual private server is accorded a size rating called its *EC2 Compute Unit,* which is pegged to the equivalent of a 1.0–1.2 GHz 2007 Opteron or 2007 Xeon processor. Table 9.1 shows the current set of Instance types, which broadly fall into the following three classes:

1. **Standard Instances:** The standard instances are deemed to be suitable for standard server applications.
2. **High Memory Instances:** High memory instances are useful for large data throughputapplications such as SQL Server databases and data caching and retrieval.
3. **High CPU Instances:** The high CPU instance category is best used for applications thatare processor- or compute-intensive. Applications of this type include rendering, encod-ing, data analysis, and others.

TABLE 9.1

Amazon Machine Image Instance Types

Type	Compute Engine	RAM (GB)	Storage (GB)1	Platform	I/O Performance	API Name
Micro instance	Up to 2 EC2 Compute Units (1 virtual core) in short bursts	0.613	EBS (Elastic Block Storage) storage only	32-bit or 64-bit	Low	T1.micro
Standard instance - small (default)	1 EC2 Compute Unit (1 virtual core)	1.7	160	32-bit	Moderate	m1.small
Standard instance - large	4 EC2 Compute Units (2 virtual cores X 2 EC2 Units)	7.5	850	64-bit	High	m1.large
Standard instance - extra Large	8 EC2 Compute Units (4 virtual cores X 2 EC2 Units)	15	1,690	64-bit	High	m1.xlarge
High Memory Double Extra Large Instance	13 EC2 Compute Units (4 virtual cores X 3.25 EC2 Units)	34.2	850	64-bit	High	m2.2xlarge
High Memory Quadruple Extra Large Instance	26 EC2 Compute Units (8 virtual cores X 3.25 EC2 Units)	68.4	1,690	64-bit	High	m2.4xlarge
High CPU Medium Instance	5 EC2 Compute Units (2 virtual cores X 2.5 EC2 Units)	1.7	350	32-bit	Moderate	c1.medium

High CPU Extra Large Instance	20 EC2 Compute Units (8 virtual cores X 2.5 EC2 Units)	7	1,690	64-bit	High	c1.xlarge

1. Storage is not persistent. All assigned storage is lost upon rebooting. To store data on AWS, you need to create a Simple Storage Service (S3) bucket or an Elastic Block Storage (EBS) volume.

System images and software

You can choose to use a template AMI system image with the operating system of your choice or create your own system image that contains your custom applications, code libraries, settings, and data. Security can be set through passwords, Kerberos tickets, or certificates.

These operating systems are offered:

- Red Hat Enterprise Linux
- OpenSuse Linux
- Ubuntu Linux
- Sun OpenSolaris
- Fedora
- Gentoo Linux
- Oracle Enterprise Linux
- Windows Server 2003/2008 32-bit and 64-bit up to Data Center Edition
- Debian

Most of the system image templates that Amazon AWS offers are based on Red Hat Linux, Windows Server, Oracle Enterprise Linux, and OpenSolaris from the list above. Table 9.2 lists some of the more common enterprise applications that are available from AWS either as part of its canned templates or for use in building your own AMI system image. Hundreds of free and paid AMIs can be found on AWS.

TABLE 9.2

EC2 Enterprise Software Types

Application Type	Software
Application Development Environments	IBM sMash, JBoss Enterprise Application Platform, and Ruby on Rails
Application Servers	IBM WebSphere Application Server, Java Application Server, and Oracle WebLogic Server
Batch Processing	Condor, Hadoop, and Open MPI
Databases	IBM DB2, IBM Informix Dynamic Server, Microsoft SQL Server Standard 2005, MySQL Enterprise, and Oracle Database 11g
Video Encoding and Streaming	Windows Media Server and Wowza Media Server Pro

Web Hosting	Apache HTTP, IIS/ASP.Net, IBM Lotus Web Content Management, and IBM WebSphere Portal Server

When you create a virtual private server, you can use the Elastic IP Address feature to create what amounts to a static IP v4 address to your server. This address can be mapped to any of your AMIs and is associated with your AWS account. You retain this IP address until you specifically release itfrom your AWS account. Should a machine instance fail, you can map your Elastic IP Address to fail over to a different AMI. You don't need to wait until a DNS server updates the IP record assignment, and you can use a form to configure the reverse DNS record of the Elastic IP address change.

There are currently four different EC2 service zones or regions:

- US East (Northern Virginia)
- US West (Northern California)
- EU (Ireland)
- Asia Pacific (Singapore)

Working with Amazon Storage Systems

When you create an Amazon Machine Instance you provision it with a certain amount of storage. That storage is temporal, it only exists for as long as your instance is running. All of the data contained in that storage is lost when the instance is suspended or terminated, as the storage is reassigned to the pool for other AWS users to use. For this and other reasons you need to have access to persistent storage. The Amazon Simple Storage System provides block storage, but is set up in away that is somewhat unique from other storage systems you may have worked with in the past.

Amazon Simple Storage System (S3)

Amazon S3's cloud-based storage system allows you to store data objects ranging in size from 1 byte up to 5GB in a flat namespace. In S3, storage containers are referred to as buckets, and buckets serve the function of a directory, although there is no object hierarchy to a bucket, and you save objects and not files to it. It is important that you do not associate the concept of a filesystem with S3, because files are not supported; only objects are stored. Additionally, you do not "mount"a bucket as you do a filesystem.

The S3 system allows you to assign a name to a bucket, but that name must be unique in the S3 namespace across all AWS customers. Access to an S3 bucket is through the S3 Web API (either withSOAP or REST) and is slow relative to a real-world disk storage system. S3's performance limits its use to non-operational functions such as data archiving and retrieval or disk backup. The REST API is preferred to the SOAP API, because it is easier to work with large binary objects with REST.

You can do the following with S3 buckets through the APIs:

- Create, edit, or delete existing buckets
- Upload new objects to a bucket and download them
- Search for and find objects and buckets
- Find metadata associate with objects and buckets
- Specify where a bucket should be stored
- Make buckets and objects available for public access

One tool commonly used to manage data for Amazon S3 is the s3cmd command line client (`http://s3tools.org/s3cmd`).

The S3 service is used by many people as the third level backup component in a 3-2-1 backup strategy. That is, you have your original data (1), a copy of your data (2), and an off-site copy of

your data (3); the latter of these may be S3. In this regard, S3 acts as a direct competitor to Carbonite's backup system. One of the options available to you is versioning for Amazon S3. With versioning, every version of an object stored in an S3 bucket is retained, provided you enable the versioning feature. Any HTTP or REST operation such as PUT, POST, COPY, or DELETE creates a new object that is stored along with the older version. A GET operation retrieves the newest ver- sion of the object, but the ability to recover and undo actions is available. Versioning also can be used for preserving data and for archiving purposes.

Amazon S3 provides large quantities of reliable storage that is highly protected but to which you have low bandwidth access. S3 excels in applications where storage is archival in nature. For example, you find S3 in use by large photo sharing sites. In the next section you'll see Amazon's Elastic Block Storage or EBS. In EBS you create virtual drives that you can use with your machine instances in the same way that you would use a hard drive with a physical system. EBS tends to be used in transactional systems where high-speed data access is required.

Amazon Elastic Block Store (EBS)

The third of Amazon's data storage systems are devoted to Amazon Elastic Block Storage (EBS), which is a persistent storage service with a high operational performance. Advantages of EBS are that it can store file system information and its performance is higher and much more reliable than Amazon S3. That makes EBS valuable as an operational data storage medium for AWS. The cost ofcreating an EBS volume is also greater than creating a similarly sized S3 bucket.

An EBS volume can be used as an instance boot partition. The advantages of an EBS boot partitionare that you can have a volume up to 1TB, retain your boot partition separately from your EC2 instance, and use a boot partition volume as a means for bundling an AMI into a single package. EBS boot partitions can be stopped and started, and they offer fast AMI boot times.

EBS is similar in concept to a Storage Area Network or SAN; you create block storage volumes varying in size from 1GB to 1TB and make those volumes available to your machine instances. Theperformance of a volume is dependent upon the network I/O and therefore varies as a function of the size of your instance (see Table 9.3), as well as the type of disk I/O operations (random, sequential, request size, and READS or WRITE) that are in progress.

When you create volumes, they appear first as raw block storage devices that must be formatted for use. A volume is mounted on a particular instance and is available to that instance alone; that is, volumes may not be shared between instances. Volumes may be located in the same zone as the AMI to which they are attached. Volumes appear as if they are devices (physical drives) when attached to an instance. You can mount multiple volumes on a single instance, if desired, and create striped RAID volumes for faster performance. The filesystem for mounted volumes appears when you open the volume, and you can install applications or copy data to mounted volumes as you would any physical disk.

Table 9.3 summarizes the various properties of the three different forms of EC2 data storage devices.

TABLE 9.3

EC2 Storage Type Properties

Property	AMI Instance	Amazon Simple Storage Service (S3)	Amazon Elastic Block Storage (EBS)	Amazon CloudFront
Adaptability	Medium	Low	High	Medium
Best usage	Transient data storage	Persistent or archival storage	Operational data stor- age	Data sharing and large data object streaming
Cost	Low	Medium	High	Low
Ease of use	Low	High	High	High
Data protection	Very Low	Very High	High	Low
Latency	Medium	Low	High	High
Least best used as	Persistent storage	Operational storage	For small I/O transfers	Operational data
Reliability	High	Medium	High	Medium
Throughput	Variable	Slow	High	High

CloudFront

Amazon CloudFront is referred to as a content delivery network (CDN), and sometimes called *edgecomputing.* In edge computing, content is pushed out geographically so the data is more readily available to network clients and has a lower latency when requested. You enable CloudFront through a selection in the AWS Management Console.

You can think of a CDN as a distributed caching system. CloudFront servers are located through- out the world—in Europe, Asia, and the United States. As such, CloudFront represents yet another level of Amazon cloud storage. A user requesting data from a CloudFront site is referred to the nearest geographical location. CloudFront supports "geo-caching" data by performing static data transfers and streaming content from one CloudFront location to another.

At the time this chapter was written CloudFront was in beta, but it has been well received. Direct competitors for CloudFront include Akamai Technologies (`http://www.akamai.com/`), Edgecast Networks (`http://www.edgecast.com/`), and Limelight Networks (`http://www. limelightnetworks.com/`). CloudFront's aggressive pricing model is expected to put pressure on these other services over time. Pricing for CloudFront is based on how much data is transferred to clients, and it doesn't require a service contract.

When you create a CloudFront implementation, a CloudFront domain name is registered for your domain name in the form `<domainname>.cloudfront.net`, and objects in the CloudFront domain can be mapped to your own domain. You store your source files on CloudNet servers in Amazon S3 buckets and then use the CloudFront API to register the S3 bucket with the CloudNet distribution. Then in your applications, Web pages, and links, you reference the distribution location.

CloudFront represents the last of the Amazon Web Services that store and serve objects and files.To store data in a way that makes it searchable and organizes it,

Understanding Amazon Database Services

Amazon offers two different types of database services: Amazon SimpleDB, which is non-relational, and Amazon Relational Database Service (Amazon RDS), both of which were in beta at the time of this writing. Dynamic data access is a central element of Web services, particularly "Web 2.0" ser- vices, so although AMIs support several of the major databases, it isn't surprising that they would create their own databases as part of the AWS Service Oriented Architecture.

Amazon SimpleDB

Amazon SimpleDB is an attempt to create a high performance data store with many database features but without the overhead. This is analogous to the goals used to create the Amazon Simple Storage System (S3). The service is meant to be low touch, in that it abstracts many of the common concerns of database administrators for hardware requirements, software maintenance, indexing, and performance optimization.

To create a high performance "simple" database, the data store created is flat; that is, it is non-relational and joins are not supported. Data stored in SimpleDB domains doesn't require maintenances of a schema and is therefore easily scalable and highly available because replication is built into the system. Data is stored as collections of items with attribute-value pairs, and the system is akin to using the database function within a spreadsheet. To support replication, a set of two consistency functions are part of SimpleDB that check data across the different copies. Transactions are performed as a set of conditional `PUTS` and `DELETES`, and you can `INSERT`, `REPLACE`, or `DELETE` values for item attributes. These transaction capabilities do not enable features like `ROLLBACK`, butthey allow you to create solutions that maintain optimistic concurrency control and will perform an `INSERT` based on the value of a counter or timestamp.

You grow a SimpleDB database by scaling out and creating additional data domains, and SimpleDBintegrates with EC2 instances and S3 storage. Data objects stored in S3 can be queried in SimpleDB, returning information about the objects' metadata and pointers to the objects' location.

Data in SimpleDB is automatically indexed and may be queried as needed. The API is relatively simple, consisting of domain creation, put and get attributes, and `SELECT` statements. Accordingto Amazon, query performance is near the level you would see for a database on a LAN, as accessthrough a browser. Although a SimpleDB database is replicated and therefore made highly available and fault tolerant, the service lacks many of the speed enhancements available to relational systems. A data domain may be located geographically in any of AWS's regions.

The design goal was to remove as much of the database system maintenance as possible. In a Webservices architecture, many applications don't require the performance level of a relational database. Among the featured uses of SimpleDB are data logging, online gaming, and metadata indexing. SimpleDB would not be the best choice for a high-volume transaction system. Data transfers within regions between SimpleDB and other AWS services are free. Service charges accrue basedon SimpleDB Machine Hours and inter-regional data transfers.

Amazon Relational Database Service (RDS)

Amazon Relational Database Service is a variant of the MySQL5.1 database system, but one that is somewhat simplified. The purpose of RDS is to allow database applications that already exist to beported to RDS and placed in an environment that is relatively automated and easy to use. RDS automatically performs functions such as backups and is deployable throughout AWS zones usingthe AWS infrastructure.

In RDS, you start by launching a database instance in the AWS Management Console and assigning the DB Instance class and size of the data store. The DB Instance is then connected to your MySQL database. Any database tool that works with MySQL 5.1 will work with RDS. Additionally, you canmonitor your database usage as part of Amazon CloudWatch. Table 9.4 shows the different Instance Classes for an Amazon RDS database. Pricing is based on machine hour rates by class, byamount of storage per month, and per million requests.

TABLE 9.4

Amazon Relational Database Service Instance Class

Type1	Compute Engine	RAM (GB)	Platform	Price2
Small DB Instance (default)	1 EC2 Compute Unit (1 virtual core)	1.7	64-bit	$0.11
Large DB Instance	2 EC2 Compute Units (2 virtual cores X 2 EC2 Units)	7.5	64-bit	$0.44
Extra Large DB Instance	8 EC2 Compute Units (4 virtual cores X 2 EC2 Units)	15	64-bit	$0.88
Double Extra Large DB Instance	13 EC2 Compute Units (4 virtual cores X 3.25 EC2 Units)	34	64-bit	$1.55
Quadruple Extra Large DB Instance	26 EC2 Compute Units (8 virtual cores X 3.25 EC2 Units)	68	64-bit	$3.10

1. Storage available is from 5GB to 1TB.

2. Price for U.S. N. Virginia deployment for database machine; storage price is $0.10 per GB-month; and I/O rate price is $0.10 per 1 million requests for the same location. Data transfer rates also apply.

Among the important features of RDS is the automated point-in-time backup system for data in thedatabase as well as for the MySQL transaction logs. Backups can be saved for up to eight days. In addition to backup, RDS supports database snapshots. A DB Snapshot is stored as a full database backup and is retained until you specifically delete it from your storage container. Snapshots may be scheduled or may be manually initiated by an administrator.

The deployment of RDS databases can be spread among multiple availability zones for increased fault tolerance and data availability. These so-called "Multi-AZ Deployments" can be automatically replicated and maintain a standby replica in another availability zone, with automatic failover when a database disruption is detected. The conversion of a single location RDS database to a Multi-DB deployment may be accomplished with a single API call. Other API calls support instancecreation and maintenance, snapshots, and restores.

Choosing a database for AWS

In choosing a database solution for your AWS solutions, consider the following factors in making your selection:

- Choose SimpleDB when index and query functions do not require relational database support.
- Use SimpleDB for the lowest administrative overhead.
- Select SimpleDB if you want a solution that autoscales on demand.
- Choose SimpleDB for a solution that has a very high availability.
- Use RDS when you have an existing MySQL database that could be ported and you want to minimize the amount of infrastructure and administrative management required.
- Use RDS when your database queries require relation between data objects.
- Chose RDS when you want a database that scales based on an API call and has a pay-as-you-use-it pricing model.
- Select Amazon EC2/Relational Database AMI when you want access to an enterprise relational database or have an existing investment in that particular application.
- Use Amazon EC2/Relational Database AMI to retain complete administrative control over your database server.