

# Cloud computing with Amazon Web Services, Part 1: Introduction

## When it's smarter to rent than to buy

Skill Level: Introductory

[Prabhakar Chaganti \(prabhakar@yelastic.com\)](mailto:prabhakar@yelastic.com)

CTO

Yelastic, LLC.

29 Jul 2008

In this series, learn about cloud computing using Amazon Web Services. Explore how the services provide a compelling alternative for architecting and building scalable, reliable applications. This first article explains the features of the building blocks of this virtual infrastructure. Learn how you can use Amazon Web Services to build Web-scale systems.

## What is cloud computing?

Cloud computing can be loosely defined as using scalable computing resources provided as a service from outside your environment on a pay-per-use basis. You use only what you need, and pay for only what you use. You can access any of the resources that live in the "cloud" at any time, and from anywhere across the Internet. You don't have to care about how things are being maintained behind the scenes in the cloud.

Cloud computing derives from the common depiction in technology architecture diagrams of the Internet, or IP availability, illustrated as a cloud. Cloud computing gained attention in 2007 as it became a popular solution to the problem of horizontal scalability.

The cloud is responsible for being highly available and responsive to the needs of your application. Cloud computing has also been called utility computing, or grid computing.

Cloud computing is a paradigm shift in how we architect and deliver scalable applications. In the past, successful companies spent precious time and resources building an infrastructure that in turn provided them a competitive advantage. It was frequently a case of "You build it first and they will come." In most cases, this approach:

- Left large tracts of unused computing capacity that took up space in big data centers.
- Required someone to babysit the servers.
- Had associated energy costs.

The unused computing power wasted away, with no way to push it out to other companies or users who might be willing to pay for additional compute cycles.

With cloud computing, excess computing capacity can be put to use and be profitably sold to consumers. This transformation of computing and IT infrastructure into a utility, which is available to all, somewhat levels the playing field. It forces competition based on ideas rather than computing resources.

Resources that your applications and IT systems constantly need (to meet growing demands for storage, computing resources, messaging systems, and databases) are essentially commoditized. You can rent this infrastructure from the vendor that provides you with the best price and service. Simple, isn't it? It's a simple but revolutionary idea that is not entirely new. It is now at the forefront of current technology trends because of the groundbreaking cloud computing environment introduced by Amazon.

## Amazon Web Services

Amazon Web Services are a set of services that provide programmatic access to Amazon's ready-to-use computing infrastructure. The robust computing platform that was built and refined over the years by Amazon is now available to anyone who has access to the Internet. Amazon provides several different Web services, but this series will focus only on the basic building block services that fulfill some of the core needs of most systems: storage, computing, messaging, and datasets.

You can architect complex and diverse enterprise applications by layering functions on top of the reliable, cost-effective building block services provided by Amazon. The Web services themselves live in a cloud outside your environment and are highly available.

### Recent Amazon Web Services success stories

[SmugMug](#), an online photo storage application that stores more than half a petabyte of data on S3, estimates cost savings on service and storage to be close to one million dollars. It is a [heavy user](#) of the Elastic Compute Cloud (EC2) computing resources to

meet surges in demand.

[37Signals](#), maker of popular online project management software Basecamp, uses S3 for storage needs.

The New York Times unleashed the [power of EC2](#) to process terabytes of archival data using hundreds of EC2 instances within 36 hours.

[Animoto](#), an online presentation video generator that needs gobs of computing power for video processing, recently [successfully withstood a surge](#) in Web traffic that would kill most companies' systems by scaling up their processing power quickly using EC2. At one point, they were using as many as 3,500 virtual instances running at the same time.

You pay as you go, based only on your usage, with no need for upfront expenditures and capital outlay. There are no maintenance costs for you, because the hardware is maintained and serviced by Amazon.

The virtual infrastructure is the great leveler in today's Web-driven world. Within minutes you can quickly piece together an infrastructure that would potentially take weeks to put together in a real-world IT shop. A key point is that the infrastructure is elastic and can scale up and down based on demand. Companies across the world are putting this elastic computing to use (see sidebar).

Freedom from the shackles of big infrastructure investment and its maintenance opens up great opportunities for innovation. You can now focus on your business ideas instead of fretting over the number of servers you have, worrying about running out of disk space, and so on. According to Amazon's estimates, businesses spend about 70 percent of their time on building and maintaining their infrastructures while using only 30 percent of their precious time actually working on the ideas that power their businesses. Amazon worries about the mundane details of the hardware and infrastructure—and how to make it highly available—while you can concentrate on bringing your ideas to life.

The central elements of this Web-scale infrastructure, which provide the most common building blocks needed for almost any nontrivial application, are:

### **Storage**

Everyone needs storage—for files, documents, user downloads, or backups. Store anything that your application needs in Amazon Simple Storage Service (S3), and take advantage of scalable, reliable, highly available low-cost storage.

### **Computing**

Amazon Elastic Compute Cloud (EC2) provides the ability to scale your computing resources up or down based on demand and makes provisioning new server instances very easy.

## Messaging

Decouple your application components by using the unlimited reliable messaging provided by Amazon Simple Queue Service (SQS).

## Datasets

Amazon SimpleDB (SDB) provides scalable, indexed, zero-maintenance storage, along with processing and querying for datasets.

You can mix and match the services as needed; they're designed to work very well with each other. Because you are running inside the Amazon environment, all communication among these services will usually be quite fast.

Future articles in this series will explore each Web service, and the libraries available for accessing it, in detail.

Entrepreneurs can build scalable and reliable applications by tapping into this virtual infrastructure, which costs far less than the traditional application-hosting platforms that require huge server farms for servicing fluctuations and spikes in demand. It also provides high levels of redundancy.

There are two levels of support available for users of Amazon Web Services:

- Free forum-based support from the Amazon staff who monitor the Amazon forums
- Paid support packages that provide one-on-one and phone support and are a more discreet way to request help

Amazon publishes the health status of all its Web services in a publicly accessible [dashboard](#) that is updated with any issues about the services. During any service outage, the Amazon Web Services team posts updates every 15-30 minutes while they're working on the issue and until it is fixed.

Amazon provides standards-based SOAP and REST interfaces for interacting with each of the services. Developer libraries either from Amazon or third parties are available in multiple languages, including Ruby, Python, Java™, Erlang and PHP, for communicating with these services. Command line tools are also available for managing your computing resources on EC2. The REST interface is easy to use; you can use a client written in any programming language that speaks HTTP to make requests to the Web services.

## Storage with Amazon S3

Amazon Simple Storage Service (S3) provides a Web services interface for the storage and retrieval of data. The data can be of any kind, and can be stored and accessed from anywhere across the Internet. You can store an unlimited number of objects in S3; the size of each stored object can range from 1 byte to 5GB. The storage itself is available either within the United States or the European Union. You

can pick the storage location for your objects when you create *buckets*, which are similar to the concept of folders in your operating system. The data is stored securely using the same data storage infrastructure that Amazon uses to power its worldwide network of e-commerce Web sites.

Access restrictions can be specified for each object that you store in S3, and the objects can be accessed with simple HTTP requests. You can even make your objects available for download using the BitTorrent protocol.

S3 completely frees you from worries about storage space, access to data, or securing the data. You don't even have to deal with the cost of maintaining the storage servers.

Part 2 will discuss S3 in detail.

Amazon ensures high availability for your files so they are available whenever you need them. The service level agreement provided by Amazon for S3 commits to a 99.9 percent uptime, measured on a monthly basis.

## Elastic computing with Amazon EC2

Amazon EC2 is a Web service that lets you requisition virtual machines within minutes and easily scale your capacity up or down based on demand. You pay for only the compute time you use. If you need to increase your computing capacity, you can quickly launch virtual instances and then terminate them once your demand decreases.

These instances are Linux®-based and can run any application or software you want. You are in control of each instance. The EC2 environment itself is built on top of the open source [Xen](#) hypervisor, which was initially developed at the University of Cambridge. Amazon lets you create Amazon machine images (AMIs) that act as the templates for your instances. Access to the instances can be controlled by specifying the permissions. You can do anything you want with them; the only restriction is that they need to be Linux-based images. Recently, Open Solaris support was announced by Amazon in a partnership with Sun, but the vast majority of the free and commercially available pre-built images for EC2 are based on Linux.

Amazon EC2 provides true Web-scale computing, which makes it easy to scale your computing resources up and down. You are completely in control of this computing environment that runs on Amazon's data center. Amazon provides five different types of servers; you can pick the ones that fit your application needs. The servers range from commodity single core x86 servers to eight-core x86\_64 servers. You can place the instances in different geographical locations, or availability zones, to ensure resistance to failure. Amazon also recently introduced the concept of elastic IP addresses that can be dynamically allocated to instances.

## Reliable messaging with Amazon Simple Queue Service

Amazon Simple Queue Service (SQS) provides access to the reliable messaging infrastructure used by Amazon. You can send and retrieve messages from anywhere using simple REST-based HTTP requests. Nothing to install, nothing to be configured. You can create an unlimited number of queues, and send an unlimited number of messages. The messages are stored by Amazon across multiple servers and data centers to provide the redundancy and reliability you need from a messaging system. Each message can contain up to 8KB of text data. The only Unicode characters that are legal in a message are:  
#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFF] | [#x10000 to #x10FFFF].

Each queue can have a configurable visibility timeout, which is used to control access to the queue by multiple readers. Once an application reads a message from the queue, the message will not be visible to any other readers until the timeout period expires. The message will reappear in the queue after the timeout period has expired, and then it can be handled by another reader process.

SQS integrates very well with the other Amazon Web Services. It provides a great way to build a decoupled system where your EC2 instances can communicate with each other by sending messages to SQS and coordinate the workflow. You can also use the queues for building a self-healing, auto-scaling EC2-based infrastructure for your application. You can secure the messages in your queue against unauthorized access by using the authentication mechanisms provided by SQS.

## Dataset processing with Amazon SimpleDB

Amazon SDB is a Web service for storing, processing, and querying structured datasets. It is not a relational database in the traditional sense, but it is a highly available schema, with a less structured data store in the cloud, and which you can use to store and retrieve keyed values. Each set of keyed values needs a unique item name; the items are themselves partitioned into domains. Each item can hold up to 256 key-value pairs of data. You can perform queries against your datasets within each domain. Cross-domain queries are not currently supported by SDB.

SDB is simple to use and provides most of the functions of a relational database. The maintenance is much simpler than a typical database, because there is nothing to set up or configure. Amazon takes care of all the administrative tasks. The data is automatically indexed by Amazon, and is available to you anytime from anywhere. A key advantage of not being constrained to schemas is the ability to insert data on the fly, and add new columns or keys dynamically.

SDB is part of the Amazon infrastructure, and the scaling is done automatically for you behind the scenes. You're free to focus your attention on more important things. Once again, you pay for only the dataset resources you use.

## Scalable architecture

Amazon Web Services can help you architect scalable systems by providing:

**Reliability**

The services run within Amazon's battle-tested, highly available data centers that run Amazon's own business.

**Security**

Basic security and authentication mechanisms are available out-of-the-box, and you can enhance them as needed by layering your application-specific security on top of the services.

**Cost benefits**

No fixed costs or maintenance costs. You pay for services as you go, and scale your resources and budget as needed.

**Ease of development**

Simple APIs let you harness the full power of this virtual infrastructure and libraries, available in most widely used programming languages.

**Elasticity**

Scale your computing resources up or down based on demand. You can go from one to any number of servers quickly to serve your application needs.

**Cohesiveness**

The four core building block services ([storage](#), [computing](#), [messaging](#), and [datasets](#)) are designed from the ground up to work extremely well together, and provide a complete solution across a wide variety of application domains.

**Community**

Tap into the vibrant and dynamic user community that's driving the widespread adoption of these Web services and is creating unique applications built on this infrastructure.

## Get ready

To start exploring the services in more detail in future articles, you will first need to sign up for an Amazon Web Services account (see [Resources](#)). It will give you the public and private security access keys, along with the x.509 security certificate, that are required when we start using the various libraries and tools in the next article.

The tools and libraries available for interacting with these Web services are written in a wide variety of languages. Articles in this series will try to stay language-agnostic and work with examples in multiple languages, but it would be helpful if you're familiar with Java, Ruby or Python.

## Conclusion

In this article, you got an introduction to Amazon's cloud computing environment and a brief overview of the four main parts of this infrastructure. Forthcoming articles in this series will examine each of the Amazon Web Services in greater detail and the various libraries and tools available for leveraging this virtual infrastructure to build your applications.

# Resources

## Learn

- Learn about specific Amazon Web Services:
  - [Amazon Simple Storage Service \(S3\)](#)
  - [Amazon Elastic Compute Cloud \(EC2\)](#)
  - [Amazon Simple Queue Service \(SQS\)](#)
  - [Amazon SimpleDB \(SDB\)](#)
  - The Service Health [Dashboard](#) is updated by the Amazon team regarding any issues with the services.
- [Sign up](#) for an Amazon Web Services account.
- The Amazon Web Services [Developer Connection](#) is the gateway to all the developer resources.
- The latest happenings in the world of Amazon Web Services are on the [blog](#).
- Read case studies and [success stories](#) from a wide variety of companies that are using Amazon Web Services to power their businesses.
- “[Cloud Computing. Available at Amazon.com Today](#)” (*Wired* magazine, Apr 2008) discusses Amazon’s cloud computing initiatives and their game changing nature.
- The New York Times article “[Cloud Computing: So You Don’t Have to Stand Still](#)” (May 2008) discusses the buzz surrounding cloud computing.
- Learn about [Xen](#), a virtual machine monitor for x86 that supports execution of multiple guest operating systems with unprecedented levels of performance and resource isolation.
- In the [Architecture area on developerWorks](#), get the resources you need to advance your skills in the architecture arena.
- Browse the [technology bookstore](#) for books on these and other technical topics.

## Get products and technologies

- Download [IBM product evaluation versions](#) and get your hands on application development tools and middleware products from IBM® DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

## Discuss

- Check out [developerWorks blogs](#) and get involved in the [developerWorks](#)

[community](#).

## About the author

Prabhakar Chaganti

Prabhakar Chaganti is the CTO of Ylastic, a start-up that is building a single unified interface to architect, manage, and monitor a user's entire Amazon Web Service cloud computing environment. He is the author of two recent books, *Xen Virtualization* and *GWT Java AJAX Programming*. Mr. Chaganti is also the winner of the Community Choice award for the most innovative virtual appliance in the VMware Global Virtual Appliance Challenge.

## Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.