

Amazon EC2 Beta



Innovation never takes a break, and neither do I. From the steaming hot beaches of Cabo San Lucas I would like to tell you about the Amazon Elastic Compute Cloud, or Amazon EC2, now open for limited beta testing, with more beta slots to open soon.

Amazon EC2 gives you access to a virtual computing environment. Your applications run on a “virtual CPU”, the equivalent of a 1.7 GHz Xeon processor, 1.75 GB of RAM, 160 GB of local disk and 250 Mb/second of network bandwidth. You pay just 10 cents per clock hour (billed to your Amazon Web Services account), and you can get as many virtual CPUs as you need. You can learn more on the [EC2 Detail Page](#). We built Amazon EC2 using a virtual machine monitor by the name of [Xen](#).

Amazon EC2 works in terms of AMIs, or Amazon Machine Images. Each AMI is a pre-configured boot disk — just a packaged-up operating system stored as an [Amazon S3](#) object. There are web service calls to create images, and to assign them to virtual CPUs to run your application. If your application consists of the usual web server, business logic, and database tiers, you can build distinct AMIs for each tier, and then spawn one or more instances of each type based on the load.

In a previous post, [Sometimes You Need Just a Little...](#), I alluded to the new world of scalable, on-demand web services. In that post I talked about the fact that sometimes a little bit of storage is all you need.

Sometimes you need a lot of processing power, and sometimes you need just a little. Sometimes you need a lot, but you only need it for a limited amount of time. Perhaps you are doing some number crunching, some in-depth text processing, some scientific research, or your end-of-month accounting. Or perhaps you want to experiment with some radical new parallel processing algorithm for a week or two. In any of these situations, acquiring sufficient hardware to accommodate the high-water mark of your usage would definitely not be economical. There are already some interesting examples of this in the [Amazon EC2 Discussion Forums](#). For example:

- Daniel Drucker [says](#) “We’re planning on using it for functional MRI analysis. We have large datasets which, when they’re being processed, require a cluster of 15-20 machines... but we only need those machines for a couple hours every few weeks.”
- In the same thread, a user by the name of spanglu says “Let’s say your back-office inventory app is web-based, but that you’re only using it from 7am to 7pm. You can cut your server costs by 50%. Take this to its logical conclusion – only start up an instance when you actually need your inventory app...”

Put another way, time is another interesting axis of scalability.

Before the advent of Amazon EC2, you had to buy or rent sufficient servers to cover your present needs, and you also had to be able to anticipate, forecast, and pay (in advance) for enough hardware, storage, and network bandwidth to accommodate organic growth as well as bursts of traffic brought upon by popular sites such as [Digg](#) or [Slashdot](#). If you are too generous with your planning, hardware sits idle. Too frugal, and your chance at fame and fortune may very well pass, as thousands of would-be users are greeted with a “site too busy” message.

With Amazon EC2, you don’t need to acquire hardware in advance of your needs. Instead, you simply turn up the dial, spawning more virtual CPUs, as your processing needs grow. During the beta you can run up to 20 virtual servers per account, or more by special arrangement.

Returning to our hypothetical three-tier application, Amazon EC2 gives you the ability to control network access on a very fine-grained basis. For example, you can allow the outside world to talk to the web servers, but not to the business logic or to the database server. You can allow the web server to talk to the business logic, and the business logic to talk to the database, and that’s it. You also get free, fast-path access to [Amazon S3](#), making S3 a natural place to store your raw data and your results.

Ok, developers, now that you have access to this computing resource, how can you take advantage of it?

For starters I would recommend that you get your hands on a copy of Cal Henderson’s new book, [Building Scalable Web Sites](#). Cal was one of the engineers behind [FlickrR](#), and his real-world experience in building a high-traffic site are aptly recounted and generalized in his book.

Second, start looking at higher level packages that will let you decompose immense computational tasks into a form suitable for parallel processing. It is no secret that one of [Google](#)’s tricks is a software framework that they call MapReduce. This framework simplifies the task of performing similar processing steps on millions or even billions of pieces of data. Doug Cutting’s open source version of MapReduce is probably worth investigating, as is Starfish, an implementation of MapReduce in [Ruby](#).

Third, consider what you can do to help other developers use Amazon EC2. What about building specialized AMIs and then selling them to other developers? Preload an AMI with a popular open source stack (being careful to respect any and all redistribution prohibitions in the software licenses). What about an advanced monitoring system that spools up additional machine images in times of heavy load, then safely winds them down after the load goes away? There are a lot of ways that you can add value on top of what’s already there.

Finally, the phrase “leveling the playing field” is used quite often with regard to the full line of Amazon’s Web Services, but I think that it is truly appropriate here. I find it worthwhile to imagine that there is a developer (maybe its you) sitting in a dorm room or a spare bedroom somewhere, burning the midnight oil and creating the next killer web site or search [algorithm](#). Without services like Amazon EC2, Amazon SQS, and Amazon S3, you’d need to invest a lot of capital, max out your credit cards, re-mortgage your house, and take a pretty big financial risk just to see if your ideas will work in the real world. With these new services your risk is a lot smaller, but your potential reward is in no way diminished.

That’s about all I have to say. You might find Jon Udell’s S3 screencast interesting, and you can always find something entertaining on the requisite [Slashdot thread](#).

— Jeff;

