



## The Public Cloud **Megaguide**

CLOUD REVIEW:  
8 PUBLIC  
CLOUD SERVICES  
COMPARED

## Deep Dive

# Cloud review: 8 public cloud services compared

*From Amazon to Windows Azure, IaaS clouds differ widely in complexity, options, and speed*

BY PETER WAYNER

**The message from the** cloud has always been simple: Surrender your cares, IT managers, and we'll handle everything. Forget about skinning your knuckles installing servers, double-checking diesel backups, or fretting about 1,000 or 10,000 things that could go wrong. Give us a credit card number and your data. We'll do the rest.

For the last few months, I've been living the dream, building a vast empire of computers that spanned the globe. Machines everywhere crunched my data into teeny tiny bits, then crunched the numbers even more. Private networks carried my secret scraps of info between the machines so that others could work the data and reform it into pretty graphs. Sure, my desktop is a bit old and it could use more RAM, but with my browser I created a worldwide army of machines with about as much ease as the sorcerer's apprentice in "Fantasia."

The good news is that, unlike the apprentice, the machines more or less disappeared when I asked them to go away. That's the beauty of the cloud. You buy what you want, when you want it. Oh, there is one errant recurring charge for a blob of bits stuck in Microsoft's Windows Azure cloud, but tech support is looking into erasing that. I expect it will be stricken soon, along with those bills for a few pennies that reminded me of the blob when they appeared on my credit card statement.

All the other machines came and went with just a small charge measured in cents. Most dollar stores have been artfully renamed to accommodate goods that cost less than \$5, but in the cloud it's still possible to buy machines as if they were penny candy. Someone should resurrect the old Woolworth name and the five-and-dime slogan.

## Key differences

The most surprising revelation after all of this exploration was that the cloud world is remarkably diverse. Anyone who assumes that the cloud machines are just commodities is latching on to the wrong part of the message. The marketing teams push the idea that the cloud lets you toss around computers and storage like they're interchangeable Lego bricks, but that's not exactly true. All the providers are trying to distinguish their machines and services by offering something a little bit different and a little bit better. Sometimes it takes a few minutes to discover exactly how they are different, but the variations are often significant for anyone doing a large amount of work.

The deviations only begin with the operating system. It's easy to assume that everything's Linux because Linux is everywhere, but that ignores the contrasts among the distributions. While many of the standard distros like Ubuntu are ubiquitous, companies have created their own versions with slight or not-so-slight enhancements. [Amazon Web Services](#) and [Google Compute Engine](#), for instance, have their own Linux for the cloud. [Rackspace](#) users can choose among a number of free versions or pay a monthly fee for Red Hat Enterprise Linux.

Linux is not the only choice. Many of the clouds make Microsoft Windows available for an extra charge, but you won't have to pay an additional fee with [Microsoft's Windows Azure](#) and [Dell Cloud](#). They want to attract Microsoft shops with services that make it easier for them to move more and more computation to the cloud. Anyone with a substantial investment in Microsoft technology will feel a bit more at home with them. Then there's [Joyent Cloud](#), which features a souped-up Open-

## Deep Dive

Solaris derivative called SmartOS.

The deeper differences lie out of sight. While all of these machines are meant to appear to be indistinguishable from buying an Intel box and putting it in your own rack, the reality is that they're often hefty multicore blades that are sliced up into virtual machines for your consumption. You're not renting a single family home that's on its own plot of land -- you're signing up for a condo or a timeshare.

### Benchmarking the machines

The difference among these machine slices become apparent fairly quickly when you start timing them. The companies try to help us along by creating units for measuring CPU power, but it's clear that these are very rough guidelines. I put the computers in my vast empire through the [DaCapo collection of Java routines](#), an especially good set of tests for anyone who's building Java applications. The DaCapo suite tests a variety of different

jobs like creating images in Java and booting up a Tomcat server. Those who don't use Java will still get a good general view of the different capabilities because each of the benchmarks exerts a different kind of stress on the machine. (See my test results in the accompanying table.)

The variation can be so dramatic that it's hard to believe the same machine is running the same test. This is because the different machines have different kinds of virtualization layers handling the different device drivers. All of these uses of the word "different" add up in funny ways, producing wildly diverging results.

Consider Lucene, a common tool used to index large collections of text documents. In the index creation test, the basic [SoftLayer](#) machine is easily more than twice as fast as the basic instance from [Amazon](#). But after the index is built, the SoftLayer machine is only a bit more than 30 percent faster at searching that index.

The results can even vary dramatically among

## Benchmarking the cloud machines

*DaCapo-9.12-bach benchmark suite (times in milliseconds; smaller is better)*

	Amazon EC2 1.7GB RAM Amazon Linux	Dell Cloud 4GB RAM Windows Server 2008 R2	Google Compute Engine Standard 1 Unit Google Compute Engine Linux	HP Cloud 1GB RAM Ubuntu 12.04	Joyent Cloud 512MB RAM SmartOS	Microsoft Windows Azure 768MB RAM Windows Azure	Rackspace Cloud 512MB RAM Ubuntu 12.04	SoftLayer CloudLayer 1GB RAM Ubuntu 12.04
<b>avroa</b>	11338	5897	4836	8135	24273	7811	15578	7154
<b>batik</b>	33675	6068	4735	exception	5231	12936	11594	2526
<b>eclipse</b>	496536	81182	68688	18659	53169	195299	147070	25252
<b>fop</b>	40661	2636	4371	7096	3444	9029	9998	1572
<b>h2</b>	142717	16333	6519	8639	exception	exception	49784	4888
<b>jython</b>	148137	6412	14017	23056	15750	34148	39388	7890
<b>luindex</b>	11786	1872	1790	2738	1770	14780	4055	1059
<b>lusearch</b>	67650	7816	5877	7156	14658	11339	14116	5544
<b>pmd</b>	84856	5164	7713	13605	11044	15918	16662	3257
<b>sunflow</b>	87103	14633	8654	14277	21818	87102	20348	8896
<b>tomcat</b>	77233	50076	64104	16397	37614	58461	23720	8059
<b>tradebeans</b>	122368	16162	11510	17297	22297	20683	21481	9887
<b>tradesoap</b>	347509	32838	36854	51359	51589	72610	75583	21332
<b>xalan</b>	71994	6584	7156	10659	14456	14294	16622	4394

*Instances tested were the smallest available, single-threaded, running the default Java Runtime Engine. For details on the DaCapo benchmark suite, see <http://dacapobench.org>. Source: Peter Wayner, InfoWorld.*

## Deep Dive

machines built in the same cloud. [Google](#), for instance, offers a number of machines that don't behave as you might expect. The high-CPU machine is about the same speed as or just a bit faster than the standard machine on most of the benchmarks. On the Tomcat simulation, it's close to twice as fast. Inexplicably, it's close to three times slower than the standard machine on the Avroca benchmark.

In these tests on Google machines, the old rules of thumb are often true, but rarely as true as you might expect. They're also often wrong. Adding more CPUs helps some of the tests that are multithreaded, but sometimes it slows them down. Adding more RAM often helps but not all of the time. All of these obvious fixes are confused a bit more by the variation in the test speeds. Most of the tests were easy to repeat in the same amount of time, but a few (such as the XALAN parser) varied a fair amount.

All of this means you'll need a doctorate and plenty of experiments to decide this basic question: How much is a machine worth? Bean counters need to factor in these benchmarks when bills come due. Clearly the [SoftLayer](#) machine is worth more if it's building a Lucene index than if it's merely searching it. This variation means it's essential that everyone actually run extensive timing tests on their instances if they want to get the most for their money. You can't assume that the cloud machine priced at 3 cents per hour is a better deal than the one priced at 4 cents.

### Data storage

One place where the effects of virtualization is felt heavily is in data storage. Databases rely heavily on the speed of the I/O channels to the disk drives, and every extra bit of virtualization can slow them down. Some clouds aren't doing much to address this because they probably assume everyone wants to run their own database machines.

Others are creating special data storage services that charge by the byte. [SoftLayer](#), for instance, offers [MongoDB](#) on separate machines that are specially tuned for writing that data. You should get better performance with this service than bringing up MongoDB on your own instance. [HP Cloud](#) and [Rackspace Cloud](#) are following a similar path by delivering [MySQL](#) as a service.

Many companies are trying to do similar things with different database technologies. They're stripping away the virtualization layers and building APIs that let people buy their storage by the bit instead of by the machine. They're counting on the fact that their highly tuned operating systems will outperform what your regular machine will be able to do.

Other services are emphasizing different performance attributes. [Amazon](#) has a wide collection of data storage solutions that will take bags of bits and return them later, but the most interesting may be Glacier, a service designed for when the retrieval times may be "several hours." Not milliseconds, seconds, or even minutes -- hours.

It's still possible to grab another machine and install your favorite storage solution, but these managed solutions can be tempting enough to make the decision for you. If one cloud has the kind of data storage layer you like, you can usually live with the other tools.

### Networking options

Another topic to obsess about is networking. Some clouds -- such as [Dell's](#) and [SoftLayer's](#) -- offer private networks that link the machines. It's easy to create a database machine that only listens to this private network, which leaves it a bit safer from the kinds of assaults that come in through the public Internet. It's not a perfect technique because security in the cloud is still, well, a cloudy subject, but it's a great first start.

Some of the other providers offer more elaborate geographical distinctions regarding the parts of their cloud. Knowing more about where your machine is located can help you make decisions about where you're going to park your data. Especially paranoid staff with especially valuable data can create an empire of machines and arrange for the data to be duplicated in different geographical regions to gain better protection against storms, fires, and other local cataclysms. [Google](#), for instance, is very transparent about the cost of bandwidth between the separate data centers and, thus, prices these transfers differently than the traffic between machines in the same center.

Bandwidth metering has the potential to be confusing. Some, such as [Dell Cloud](#), charge nothing for incoming traffic, a trick that simpli-

## Deep Dive

fies the metering and accounting while building a sort of roach motel for your information; the data moves in but doesn't want to leave. If you're creating a big, number-crunching machine like the one from "A Hitchhiker's Guide to the Galaxy" that will suck up plenty of data but return only one number (42) as the answer, data plans like this are attractive.

### Beyond the basics

The most interesting parts of the cloud are the special-purpose machines. Even if they aren't generally useful for the PHP code you have to get running today, they promise to make it dramatically easier to tackle some challenges in the future. [Amazon](#), for instance, has a set of video cards (GPUs) ready for any experiments that you can dream up and cast into algorithms that are easy for these cards to execute. Physicists, biologists, and computer scientists are already converting their algorithms to run on these cards. It's just one example of how the clouds are making it easier for all of us to experiment with new architectures.

These special stacks don't always require special hardware. A number of the clouds -- including [Amazon](#), [Joyent](#), and [Windows Azure](#) -- offer special Hadoop machines to feed the frenzy. They're tuning the underlying operating system and optimizing the JVM for better performance. [Joyent](#) claims that its machines run "nearly 3x faster." Do they? It depends upon what you're asking the machines to do. [Amazon](#), for its part, has a cloud of machines that take Hadoop jobs directly and let you bid for computer time on spare machines.

There are other sets of features that didn't end up mattering to me. Some of the clouds are doing a better job with performance measurements and fancy graphical dashboards than others. These sucked me in at first, but then I stopped paying much attention to them. Knowing the overall load on the machine is helpful, but most developers are going to need to hack their own statistics to get a better feel for the throughput of their collection

of machines. Your requirements may vary, though, and the extra features may be just what you need.

Another similar set of features may end up becoming more significant. Some of the newest features appearing on the clouds make it easier to automate vast armies of machines, then change the configuration on each of them a small amount. [Amazon](#) lets you create hundreds of new machines from the same image, then pass in configuration information that allows each of them to modify itself. There's no need to log into each machine independently and configure it.

The value of features like this depends heavily on the kinds of jobs you're running. If your stacks are fairly static, this feature won't make much difference. But if you're building up and tearing down big collections of machines, the ability to automate the configuration is essential. Expect more support for features like this to dominate the choices for the people who are working with sporadic bursts of big data to crunch.

### The right cloud for you

If there's one lesson that's emerged from all of this, it's that the answer is never cut and dried. The cheapest machine for you may not be the cheapest machine for me. The best bandwidth price plan for you might be prohibitively expensive for me. The benchmarks vary, as do the prices for data storage. Each of us is forced by the system to spend time crunching numbers and running tests before making a decision.

This is part of the fun. The cloud may appear to smooth over all of the complexity involved in running a bunch of servers, but what the providers are really doing is solving all of the annoying problems while opening up the freedom to choose different architectures. The options are becoming more transparent and easier to make now that we don't have to worry so much about backup generators and rack capacity. After spending a few months playing around with my vast empire, I'm just realizing that I'm not really finished.

## Deep Dive

REVIEW:

# Amazon, mother of all clouds

*Amazon Web Services leads with a luxurious array of options, resources, and services, but trails in performance and price*

BY PETER WAYNER

**Ah, Amazon -- did** Jeff Bezos choose that name to symbolize the largest bookstore in the world or did he realize that he would one day create an enterprise cloud service that was as large and complex as the river basin? After spending some time with his enterprise infrastructure service, I think he saw this coming.

Selling servers by the hour was a bold idea when the Amazon cloud business launched a few years ago, but it seems quaint compared to all the options for sale today. There are currently 21 products available on Amazon Web Services, and only one of them is the classic EC2 machine, an abbreviation of the full name, the Elastic Compute Cloud. The original S3 (Simple Storage Service) now has cousins like the Simple Workflow Service and SimpleDB, a nonrelational data store. Then there are odder innovations like Amazon Glacier, a very cheap storage solution that takes hours to retrieve the data. Yes, hours. Not milliseconds, not seconds, not minutes -- but hours.

It's impossible to summarize it all in a para-

graph or even an article. Amazon Web Services would require a book, but that tome would be out of date by the time it was printed because the service changes quickly. The best news is that Amazon is constantly looking at costs and generally lowering prices as it finds a way to deliver the product for less. Some prices have gone up occasionally over the years, an effort to make the prices reflect reality.

Amazon has also found plenty of supporters. A number of big companies such as Netflix are proud of using Amazon's servers, and plenty of startups are glad they didn't need to set up their own data centers to reach for the gold ring of IPO riches. Some customers brag about spending \$1 million or more a month, an amount that would be more than enough for most companies to justify setting up an in-house facility and team. Clearly, Amazon is delivering a whole lot of value.

## A smorgasbord of possibilities

The vast array of options is probably what keeps

The screenshot shows the Amazon Management Console's 'My Instances' page. On the left is a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main area displays a table of instances with columns for Name, Instance, AMI ID, Root Device, Type, State, Status Checks, Alarm Status, Monitoring, Security Groups, Key Pair Name, and Virtul. Three instances are listed, all in a 'shutting-down' state. Below the table, a summary shows '3 EC2 Instances selected' and lists the instance IDs: HighCPU-medium (i-fe05f980), highCPU-AMazon Linux (i-ec639f92), and i-40728e3e. A 'Description' tab is active, showing details for the three instances, including their IDs and the compute-1.amazonaws.com endpoint.

Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	Security Groups	Key Pair Name	Virtul
HighCPU-medium	i-fe05f980	ami-3d4f254	ebs	c1.medium	shutting-di	none	basic	basic	quicklaunch-1	key3	parav
empty	i-40728e3e	ami-3d4f254	ebs	t1.micro	shutting-di	none	basic	basic	quicklaunch-1	key3	parav
highCPU-AMazon Linux	i-ec639f92	ami-1624987f	ebs	c1.medium	shutting-di	none	basic	basic	quicklaunch-1	key3	parav

3 EC2 Instances selected.

EC2 Instances: HighCPU-medium (i-fe05f980), highCPU-AMazon Linux (i-ec639f92), i-40728e3e

Description | Status Checks | Monitoring | Tags

i-fe05f980: ec2-184-73-120-51.compute-1.amazonaws.com  
 i-ec639f92: ec2-50-16-23-156.compute-1.amazonaws.com  
 i-40728e3e: ec2-50-17-126-106.compute-1.amazonaws.com

The Amazon EC2 control panel tracks the life of the machine and lets you connect directly to the instance via a Java-based version of SSH.

## Deep Dive

people coming back. When I started setting up a few test machines, it was clear Amazon had expanded the options until they no longer seem like commodities. There are at least 16 different sizes of machines. The instances generally bundle more RAM with more CPU cores and more disk space, but you can also choose lopsided versions that are heavier on the RAM, the CPU, or the I/O.

The size is just the first feature you can choose. There's back-end storage that can be mounted and you can fiddle with the amount of disk space. If you like, you can add EBS (Elastic Block Store), which is disk space that lives in the racks near you. This can be faster or slower and backed by more or less RAID protection.

There are so many options that spinning up an Amazon machine is almost as complicated and as flexible as buying a custom server. It's a bit like a toy store because you have to resist the temptation to play with cutting-edge technology -- such as one of the machines jammed full of Nvidia Tesla GPUs ready to run highly parallel algorithms written to Nvidia's CUDA platform. The mind often boggles.

Decoding the pricing table will take some collaboration between the CFO and the CIO. Not only are there 16 different-sized machines, but you can pay to reserve them in advance. If you pay a portion up front, Amazon will cut the hourly price along the way. It's sort of like one of those warehouse clubs where a membership buys you a discount. If you're judicious it's probably worth it, but it will take you some time to predict how much you'll use the machines.

The options aren't just in the size or configuration of the machine. The startup process offers a number of sophisticated options for customizing the distro from the beginning. You can, for instance, set up a "security profile" that controls which ports are open or shut immediately. This saves you the trouble of logging in after creating the machine and configuring the ports manually, a feature that's

essential if you're going to start and stop dozens, hundreds, or thousands of machines.

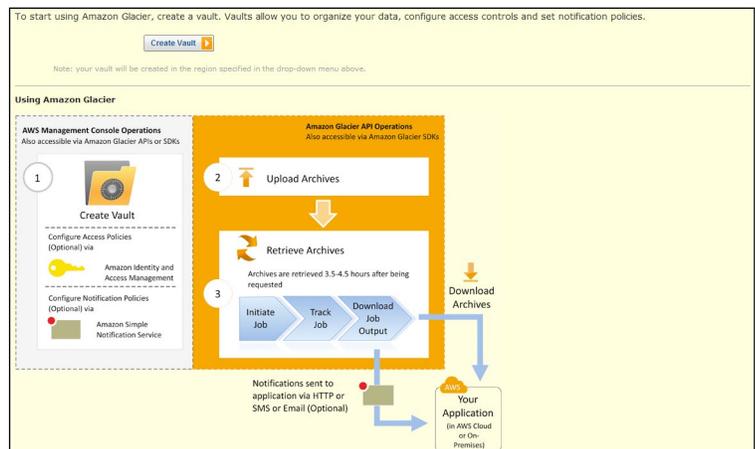
### Benchmarking the cloud

I spent some time running benchmarks on the micro machine, Amazon's low-end model that's supposed to be able to handle bursts of extreme computation. It's intended for people who are either just testing some ideas or building a low-traffic machine. It costs only 2 cents per hour and comes with 613MB of RAM, an odd number that's probably an even fraction of some power of two minus a little overhead.

It was surprisingly hard to find a way to log into the machines. I couldn't get the public/private keys generated by Amazon to work with either PuTTY or the built-in Java-based SSH client. Yet it worked in seconds from my Mac's terminal. I wonder what kind of laptops are popular up at Amazon?

Little issues like this appeared fairly often during my time poking around the cloud. Amazon's Web portal is one of the more sophisticated tools available, offering more extensive diagnostics and hand-holding than the dashboards of competitors, but it is not always foolproof.

For instance, it offers a nice dialog box for helping you connect immediately to your instance with your SSH by formatting the command line. It worked some of the time for me, but it failed when it tried to get me to log into one of my Ubuntu instances as root, a problem that took five seconds to fix once I remembered that I was supposed to log in as "ubuntu." Any Unix user should be able to



**Amazon's instructions for setting up a Glacier vault, a low-cost storage solution that takes hours to find the data.**

## Deep Dive

work around all of these tiny glitches. In fact they're only noticeable because Amazon sets such a high bar with the quality of its portal.

The speed I saw with the machines wasn't very exciting. I tried the [DaCapo Java benchmarks](#), a test suite that includes several computationally intensive tasks, including running a Tomcat server. The results were generally three to five times slower than the low-end machines on [Microsoft's Windows Azure](#) and often six to nine times slower than the low-end machines on [Joyent's cloud](#). However, these numbers weren't perfectly consistent. On the Avro simulation of a sensor network, the EC2 micro machine was faster than Joyent's, and it took only about 45 percent more time to finish than the low-end Azure machine.

The Joyent machines are priced at about 3 cents an hour, a small premium considering the gap in performance. The Azure machines have an introductory price of 1.3 cents per hour -- cheaper than Amazon's micros, though they're dramatically faster.

### Bigger, faster, more

For comparison, I also booted up what Amazon calls a high-CPU machine that offers two virtual cores, each delivering 2.5 (in Amazon parlance) ECUs or Elastic Compute Units. That's five ECUs all together. The micro machine is supposed to offer two ECUs in bursts, while the high-CPU machine offers five ECUs all of the time. The price is dramatically higher -- 16.5 cents per hour -- but that includes 1.7GB of RAM. Again, what happened to our old friends, the powers of two?

The high-CPU machine was usually six to eight times faster than the micro machine, suggesting that the ECUs are just a rough measurement. The results were close in speed to the Joyent machine and often a bit faster, but at more than five times the price. It's worth noting for algorithm nerds that the DaCapo benchmarks used two threads when possible on the Amazon machine but were limited to one thread on the Joyent and Azure boxes.

Once again, this suggests that the algorithm designer, the build master, and the CFO are going to need to sit down and decide whether to buy bigger, faster machines for more money or live with a larger number of slower, cheaper machines.

More fun comes when you start exploring the

other corners of the Amazon toy store. The pay-as-you-go Hadoop cloud, called Elastic MapReduce, lets you upload a JAR file, push a button and start the computational wheels turning. You stick the data in the Amazon's storage cloud, S3, and the results show up there when everything is done.

There's a separate cloud of machines devoted to doing Hadoop processing. At least it looks separate, because you buy the compute cycles through a different Web page, but it could all be running in the same floating network of machines. That's the point.

If you want your Hadoop job to begin as soon as a machine is available, you pay the list prices. If you want to gamble a bit and wait for empty machines, the spot market lets you put in a lower bid and wait until spare machines are available for that price. Amazon is experimenting with constantly running an auction for compute power. This is yet another wrinkle for the engineers and the accountants to spend time discussing.

### Beyond commoditization

My favorite, relatively new feature is the Amazon Glacier, a backup system that takes hours to recover the data. Many people looked at Amazon's first cloud storage solution (S3) and found it was too expensive for backups or other data that wasn't accessed very often. One-size-fits-all solutions are one of the limitations of the cloud. Amazon designed S3 to meet the needs of servers that must access data relatively quickly.

As I mentioned before, there's no easy way to cover all of Amazon Web Services in one article like this. The only solution is to wade in, start booting up machines, and begin testing your application. Amazon offers some very basic services for free to help new customers, but for the most part it costs only a few cents to try out the different sizes. Then you can sit down with your accountant and start pricing out the services.

My impression is that Amazon's cloud has evolved into the high-end Cadillac of the breed. It provides extensive documentation, more hand-holding, and more sophisticated features than rivals, all at a price that is higher than the competition. Perhaps the competition's rates are only temporary and perhaps they're unsustainable, but maybe Amazon's rate is the price you pay for all of the extra features. Amazon's cloud is loaded with them.

## Deep Dive

REVIEW:

# Dell Cloud lets you have it your way

*Dell's VMware-based cloud infrastructure provides all of the flexibility and complexity of the leading enterprise virtualization platform*

BY PETER WAYNER

**In business, you go** where your customers are. If the kids want to listen to that rock and roll music, well, you put it on the jukebox. If the enterprise caretakers want to buy something from a cloud, then you bundle up your server boxes and call them a cloud. That's what Dell is doing. If time is too short to buy your Dell machines with a purchase order and take delivery, you can call up the company and it will start them up in its data center.

Dell's new cloud has a distinctly Dell flavor that's apparent from the beginning. The company has always been very close to Microsoft, and now it's even closer after the [leveraged buyout](#). While other clouds charge a bit more for a Microsoft license, you get one to Windows Server 2008 R2 as part of the whole bundle. The Dell Cloud portal where you control your machines insists that you log in via Internet Explorer or Firefox. Chrome isn't even on the list.

The sales process is also very Dell. You can buy a machine by the hour, but the first options you see ask you to reserve a chunk of hardware for a month -- much as you might if you were leasing a real slab of silicon. Dell's sales team is ready to help at any time. A "small" machine comes with one virtual CPU, 2GB of RAM, and 100GB of storage for a going rate of \$125 a month, averaging about 17.5 cents an hour.

A medium instance -- the size I tested -- has four virtual CPUs, 8GB of RAM, and 400GB of storage for \$500 a month. If you want to buy by the hour, it's 5.5 cents per virtual CPU per hour, 7 cents per gigabyte of RAM per hour, and 30 cents per gigabyte of storage per month. Once you reserve this hardware, you can then split it up into VMware virtual machines, just as if you purchased a real piece of hardware and installed VMware.

## Virtualization and freedom

The biggest difference about Dell may be in the openness to the virtual machine part of the stack. All of the other major cloud companies take your money and give you root on some virtual machine. Then they pretend that much of the virtualization isn't there. The root password makes it look as if you're logging into your very own box, when in reality you're logging into a virtual machine that's sharing one piece of hardware with a bunch of other customers.

With Dell, you open up your Dell Cloud portal and find a VMware vApp, described by one Dell support engineer as the equivalent of a rack where you can stick your own virtual machines. To fill the virtual rack, you can draw on a few standard templates to create an F5 load balancer, a Windows Server 2008 R2 machine, or a Suse Linux 11 box, but of course you're also welcome to upload any VMware or OVF virtual machine.

There's also an option for starting up a machine with a particular ISO file -- useful if you want to boot up a particular LiveCD version of Linux or any other OS. The portal even lets you pretend that you're accessing the CD/DVD drive on your machine though you're just uploading ISO files.

The ability to poke around at this level is liberating. You can mess around with a virtual machine on your desktop using [VMware Workstation](#) or [VMware Fusion](#), then upload it to your virtual rack and start it up in the server farm. Most of the other clouds let you create images of your servers, but usually you end up doing the work to build the image on their machines.

The VMware software layer is also employed when you start communicating with your running

## Deep Dive

machines. You can open up the SSH port if you like, but the simplest path is to use VMware's remote client. One screen on your portal shows a list of all of your virtual machines. If you click on one row, the VMware remote client will start up a connection in another window of your browser. This process was a bit glitchy for me, but I finally got it to work with Firefox. Once it started going, I was able to fiddle with the server from my desktop. The video wasn't as snappy, but that's the price of working across the country from the server.

To get a feel for the speed of Dell's cloud machines, I pushed a Windows Server 2008 R2 virtual machine through the [DaCapo benchmarks](#), a set of Java routines that tests many common Java server applications. As with other virtual machines in other clouds, the results varied greatly. Many of the benchmarks were 50 to 100 percent faster than an Amazon High-CPU instance (14.5 cents per hour). But others, such as the image rendering tests (batik and sunflow), ran neck and neck.

These differences mean you *must* try out your application yourself to see if you're getting the performance you want. For instance, the lucene indexing routines were faster on the Dell medium

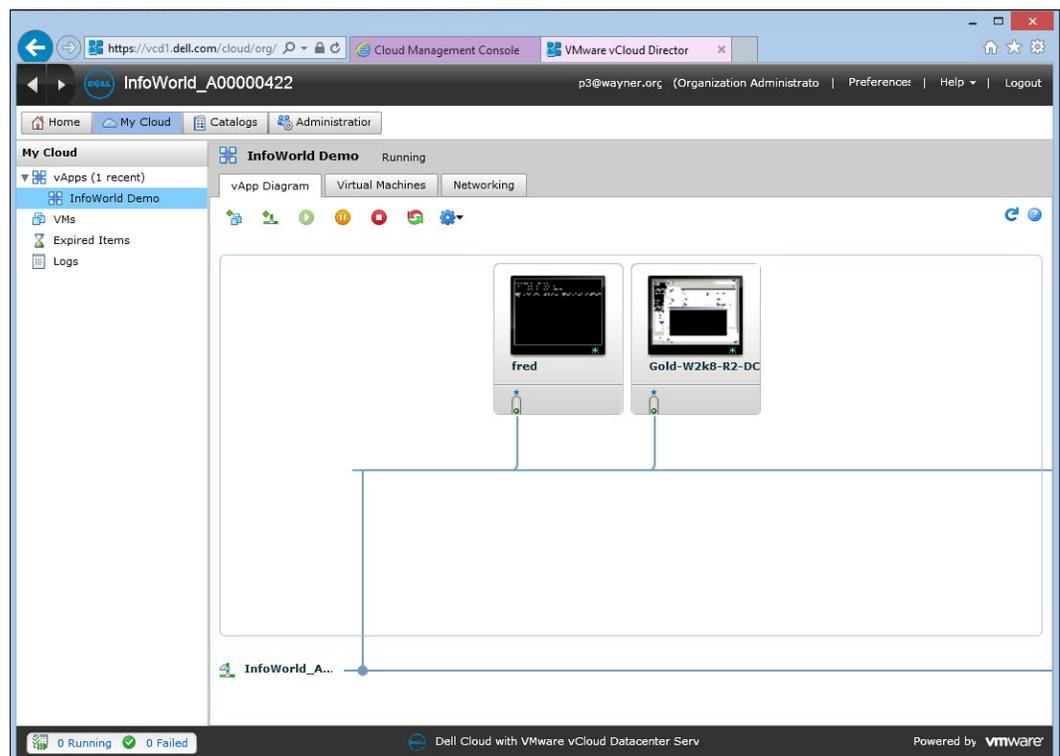
than on the Amazon High-CPU box, but the searching tests ran in the same amount of time. The good news is you can fiddle with the VMware machine on your desktop until you get the right combination of software packages and device routines to improve your performance.

### The network you know

Dell is also offering the same kind of transparency for the network configuration. You can choose between a number of different networking options for your VM once you get it running. You can configure internal and external networks, as well as reconfigure your virtual boxes in much the same way as you would your real servers. When you want your machines to speak to the outside, you can monkey around with NAT and DHCP to pass out the external IP addresses.

I'm a bit torn about this approach. Some of the other clouds sweep all of this under the rug and simply connect your box to the outside Internet. You get a root password and an IP address open to all. It's much easier to get rolling, but of course there's no flexibility.

Other clouds have dedicated internal and



**If you like VMware, you'll like Dell Cloud. Here, the Dell Cloud portal offers a visual depiction of the virtual machines you have running in your VMware vApp.**

## Deep Dive

external networks but still hide the details. Dell's approach will be familiar to anyone running the network in an office or an internal server farm because the steps are similar. The technologies are the same and you can use all of the flexibility if you want to do so. It may be a bit more work, but that's the price for the openness.

Dell, like the others, is not charging for data coming into the system, just for data leaving it (24 cents per gigabyte). Dell Cloud doesn't yet boast an elaborate collection of different regions and services, unlike many of the other clouds. Amazon, for instance, has at least 18 different lines on its data price sheet that govern how much it costs to move a gigabyte from point A to B.

Dell Cloud isn't that complicated yet. But it will probably lose a bit of this simplicity as it grows into the space and starts offering different storage

and database options. For now, Dell Cloud will be most attractive to IT staffs used to buying and configuring Dell or Windows machines in their own networks. Dell Cloud offers a wonderful amount of openness that will be familiar to anyone who's set up a rack of virtual machines with VMware in their own server room.

The big advantage to moving to Dell's cloud, of course, is that Dell handles much of the grungy details like bolting machines into racks and hooking up the air conditioning. But everything else will seem just as familiar as calling up your Dell representative, putting in an order, and installing the software you want. You just won't have to wait for FedEx to deliver and the server room staff to bolt it into a rack.

## Deep Dive

## REVIEW:

# Google Compute Engine rocks the cloud

*Google's new compute cloud offers a crisp and clean way to spin up Linux instances and easily tap other Google APIs*

BY PETER WAYNER

**You're sitting around.** You have some computing to do. Ten years ago, you would ask your boss to buy a rack or two of computers to churn through the data. Today, you just call up the cloud and rent the systems by the minute. This is the market that Google is now chasing by packaging up time on its racks of machines and calling it the Google Compute Engine.

[Google took its sweet time](#) entering this corner of the cloud. While Amazon, Rackspace, and others started off with pay-as-you-go Linux boxes and other "infrastructure" services, Google began with the [Google App Engine](#), a nice stack of Python that held your hand and did much of the work for you. Now Google is heading in the more general direction and renting raw machines too. The standard distro is Ubuntu 12.04, but CentOS instances are also available. And you can store away your own custom image once you configure it.

## Google's big selling point

Why rent machines from Google instead of Amazon or Rackspace or some other IaaS provider? Google claims its raw machines are cheaper. This is

a bit hard to determine with any precision because not everyone is selling the same thing despite claims of computing becoming a commodity. Google sells its machines by the Google Compute Engine Unit (GCEU), which it estimates is about a 1GHz to 1.2GHz Opteron from 2007.

All of Google's machines rent for 5.3 cents per GCEU per hour, but that isn't really what you pay. The smallest machine you can rent from Google today, the so-called n1-standard-1-d, goes for 14.5 cents per hour. That's because the n1-standard-1-d -- which comes with one virtual core, 3.75GB of RAM, and 420GB of disk space -- is equivalent to 2.75 GCEUs, according to Google. You can get machines with two, four, and eight virtual cores all at the same price per GCEU.

These numbers are bound to evolve soon according to a member of the Google Compute Engine team. The product is said to be in "limited preview," and as it grows more polished, the company will probably experiment with adding more options with more or less power.

Is 5.3 cents per GCEU a good deal? It depends upon what you want to do with your machine.

The screenshot shows the Google Compute Engine dashboard. On the left is a navigation menu with options like Overview, Services, Team, API Access, Reports, Quotas, and Google Compute Engine. The main area has tabs for Overview, Instances, Disks, Networks, Zones, and Operations. Below the tabs are buttons for 'Create New Instance', 'Refresh', and 'Load more'. A table lists instances with columns for Name, Machine Type, Zone, External Address, and Status. Two instances are shown: 'teamalpha' (n1-standard-1, us-central1-a, 173.255.113.34, RUNNING) and 'teambeta' (n1-standard-1, us-central1-a, 173.255.113.247, PROVISIONING). A 'Delete' link is present for each instance. At the bottom of the table, it says 'No more resources to list'.

Name	Machine Type	Zone	External Address	Status
teamalpha	n1-standard-1	us-central1-a	173.255.113.34	✓ RUNNING
teambeta	n1-standard-1	us-central1-a	173.255.113.247	⚙️ PROVISIONING

Google Compute Engine gives you a clean, Google-esque Web dashboard to create instances, assign them to zones, and monitor their status. So far, you can choose from Ubuntu and CentOS images.

## Deep Dive

Rackspace prices its machines by the amount of RAM you get. It has stopped selling the anemic 256MB RAM VMs, but rents its 512MB boxes at only 2.2 cents per hour or \$16.06 per month. If you want a machine with 4GB from Rackspace, it will cost you 24 cents each hour, about \$175 per month.

Is that a better deal? If your computation doesn't need the RAM, a basic instance from Rackspace is much cheaper. Even if the CPU might not be as powerful, you would be better off with a cheaper machine. But I suspect many will need fatter machines because modern operating systems suck up RAM like a blue whale sucks up krill.

After you get past the differences over RAM and disk space, the Google machines are meant to be essentially the same as the machines from Amazon or Rackspace -- or even the machines you might buy on your own. Like Amazon and

Rackspace, Google makes it easy to start off with Ubuntu; after that, you're talking to Ubuntu, not Google's code. There are differences in the startup and shutdown mechanisms, but these aren't substantial. More substantial is Google's inability to snapshot persistent storage, as you can in Amazon, but Google promises this is coming soon.

If you're migrating from Amazon or Rackspace, you'll need to rewrite your scripts because the APIs are full of linguistic differences, even if they offer most of the same features.

### Google Compute Engine ins and outs

Another big part of the equation is bandwidth. Google doesn't charge for ingress, but it has a fairly complicated model for egress. Shipping data to a machine in the same zone in the same region is free, but shipping it to a different zone in the same region is one penny per gigabyte. Then the cost for

The screenshot shows the Google APIs console interface. At the top, it says "Google apis" and "My Project" with a dropdown menu. Below that, there are tabs for "All (46)", "Active (1)", and "Inactive (44)". A sidebar on the left contains navigation links: Overview, Services, Team, API Access, Reports, Quotas, and Google Compute Engine. The main content area is titled "All services" and includes the instruction "Select services for the project." Below this is a table with columns for Service, Status, and Notes.

Service	Status	Notes
Ad Exchange Buyer API	OFF	Courtesy limit: 1,000 requests/day
AdSense Host API	Request access...	Courtesy limit: 100,000 requests/day
AdSense Management API	OFF	Courtesy limit: 10,000 requests/day
Analytics API	OFF	Courtesy limit: 50,000 requests/day
Audit API	OFF	Courtesy limit: 10,000 requests/day
BigQuery API	OFF	Courtesy limit: 10,000 requests/day • Pricing
Blogger API v3	Request access...	Courtesy limit: 10,000 requests/day
Books API	OFF	Courtesy limit: 1,000 requests/day
Calendar API	OFF	Courtesy limit: 10,000 requests/day
Custom Search API	OFF	Courtesy limit: 100 requests/day • Pricing
Drive API	OFF	Courtesy limit: 500,000 requests/day
Drive SDK	OFF	
Firebase API	OFF	Courtesy limit: 100,000 requests/day
Fusion Tables API	OFF	Courtesy limit: 25,000 requests/day
Google Affiliate Network API	OFF	Courtesy limit: 1,000 requests/day
Google Cloud Messaging for Android	OFF	
Google Cloud SQL	OFF	Pricing
Google Cloud Storage	OFF	Pricing

**Ready integration with other Google services is one of Compute Engine's main attractions. It's just one of 46 services that you can access through Google's Developer API.**

## Deep Dive

letting the data “egress” to the Internet depends upon whether it’s going to the Americas/EMEA or the APAC (Asia and the Pacific). For what it’s worth, egressing the data to some website visitor from the APAC is almost twice as expensive as egressing it to someone in the United States. The costs are set on a sliding scale with discounts for big egressers.

While the complexity of the pricing table will send the purchasing managers to their calculators, it’s interesting what Google is trying to do with this scheme. By making intermachine communications free, Google is no doubt banking on people using the racks in the same zones to actually work together on solving problems. In other words, Google is giving us the tools for stitching together our own supercomputers.

In general, Google is doing a good job of making some of the dangers of the cloud apparent. Like compute instances in [Amazon, Rackspace, and other IaaS clouds](#), each Google instance comes with “ephemeral disk,” a name that makes the storage sound more fragile than it really is. Keep in mind that the file system that comes with your cloud computer -- be it on Amazon, Rackspace, or Google -- is not backed up in any way unless you code some backup routines yourself. You can [run MySQL](#) on your cloud box, but the database won’t survive the failure of your machine, so you better find a way to keep a copy somewhere else too.

Calling the storage “ephemeral” makes it obvious that the data might go elsewhere during a real failure or even a “maintenance window.” If anything, the name might overstate the dangers, but it all becomes a gamble of some form or another. The solution is to purchase separate “persistent disk” space and store your information there. Or you might want to put it in Google Cloud SQL, the [BigQuery data store](#), or one of the other services offered by Google.

If words like “ephemeral” still sound off-putting, the documentation says Google will negotiate service-level agreements for enterprise customers that begin with promises of 99.95 percent uptime.

Google is also making the dangers of location apparent. One section of the documentation addresses just how you should design your architecture around potential problems. The various zones and regions may go down from time to time, and it’s your responsibility to plan ahead for these

issues. Google makes the costs of shipping the data transparent, so you can come to intelligent decisions about where to locate your servers to get the redundancy you need.

### A Compute Engine with a view

Google Compute Engine is just one part of the Google APIs portal, a grand collection of 46 services. These include access to many of Google’s biggest databases such as Books, Maps, and Places, as well as to some of Google’s lesser-known products like the Web Fonts Developer API.

I suspect many developers will be most interested in using Google Compute Engine when they want to poll these Google databases fairly often. While I don’t think you’re guaranteed to be in the same zone as the service you want, you’re still closer than when traveling across the generic Web. Google offers “courtesy” limits to many of these APIs to help out new developers, but you will end up paying for the best services if you use them extensively. These prices are changing frequently as Google and the developers try to figure out what they’re really worth.

Google says some experimenters are already pairing the Compute Engine with the App Engine to handle expensive computations. In one of the experiments, [Google worked with a biology lab to analyze DNA \[PDF\]](#). The data was uploaded through an App Engine front end, then handed over to a block of Compute Engine cores to do the work. The Compute Engine machines were started up when the data arrived, and they were shut down and put back in the pool as soon as their work was done.

You can start and stop your machines by hand and track them with the Web portal, but I suspect many will end up using the command-line tool. Google distributes some Python code that handles most of the negotiations for reserving, starting up, and stopping servers. While the Web portal is OK for small jobs, the ability to easily write scripts makes the command-line version more useful.

The command-line tool is also more powerful. You can create instances through the Web GUI, but there’s a limit to how far you can go. I couldn’t figure out how to log in with SSH through the portal, then I switched back to the command line. Perhaps Google should check out some of the

# Deep Dive

[HTML5-based tools like FireSSH](#) that integrate SSH with a Web page. The only real challenge is finding a good way to hold the SSH keys.

One of the more interesting features is the way to bind metadata to each computer. Google is clearly intending for people to write their own automatic routines for bringing machines online and off. If you want your software to be self-aware, it can look at the metadata for each instance, and the instance can also read the metadata about itself. This lets you pass in configuration information so that each new machine is not born with a clean slate.

If you want to build your own collection of Linux boxes, Google Compute Engine offers a nice, generic way to buy servers at what -- depending on the size of compute instance you need -- can be a great price. The most attractive feature will probably be the proximity to the other parts of the Google infrastructure. Google is as much a data vendor as an advertising company, and the collection of APIs is growing nicely. I can see how some companies will want to run their computational jobs in the Google cloud just to be closer to these services.

The screenshot shows the Google Cloud Platform console for provisioning a new Compute Engine instance. The 'REST Request' tab is active, displaying the following JSON payload:

```
{
  "kind": "compute#instance",
  "disks": [],
  "networkInterfaces": [
    {
      "kind": "compute#instanceNetworkInterface",
      "accessConfigs": [
        {
          "name": "External NAT",
          "type": "ONE_TO_ONE_NAT"
        }
      ],
      "network": "https://www.googleapis.com/compute/v1beta2/projects/iw-test/networks/default"
    }
  ],
  "serviceAccounts": [
    {
      "kind": "compute#serviceAccount",
      "email": "default",
      "scopes": [
        "https://www.googleapis.com/auth/userinfo.email",
        "https://www.googleapis.com/auth/compute",
        "https://www.googleapis.com/auth/devstorage.full_control"
      ]
    }
  ],
  "machineType": "https://www.googleapis.com/compute/v1beta2/projects/iw-test/machine-types/n1-standard-1",

```

**The Web GUI for provisioning instances and disk shows the REST packet that will actually do the work, in case you want to automate the task later.**

## Deep Dive

REVIEW:

# HP Cloud aims for the enterprise

*HP's OpenStack-based IaaS cloud blends openness and portability with nice proprietary extras and welcome hand-holding*

BY PETER WAYNER

**Hewlett-Packard may be best** known for its ubiquitous printers and laptops, but in the enterprise world, it is just as recognized for its servers. Now that the idea of the cloud is taking over, HP is joining the marketplace and renting some of its servers in the HP Cloud.

The servers are priced by the hour just like everyone else's, but though these machines may be commodities like hamburgers, there are differences, just as there are differences between Burger King and McDonald's.

For instance, HP offers a longer list of Linux distributions for your new server slice than some of the others. You can get the classics such as Ubuntu, Debian, and CentOS in many of the best-known versions. If you know what you're going to do with the machine, you can start right up with a Bitnami distribution sporting a number of pre-installed applications like Drupal.

Not everything is on the list. Windows Server and Red Hat Enterprise Server, two operating systems offered for a bit more money by Rackspace and other clouds, aren't anywhere to be seen. This promises to be temporary. Marc Padovani, HP Cloud Services director of product management, suggests that Windows will arrive sooner rather than later. And a long list of [solution partners](#) shows that HP Cloud is embracing a broad commercial ecosystem along with the open source software.

The HP Cloud is built on OpenStack, which should be attractive for any enterprise manager worried about being locked into a cloud. Sure, all of the cloud vendors talk as if the machines are really commodities, but if it takes you several months to recode your scripts, mobility is extremely limited. HP's embrace of OpenStack is clearly a push to

attract managers who want the flexibility to outgrow the HP Cloud or just move on. The wide range of Linux distributions feels like a part of that plan.

As with the other clouds, the price list for HP Cloud machines largely follows the amount of RAM. The smallest offering delivers 1GB of RAM for 4 cents per hour. This price is cut in half now during a special "public beta" promotion. By comparison, Rackspace charges 6 cents per hour for a 1GB machine.

HP starts tossing in additional virtual CPUs with more RAM. A 2GB machine comes with two virtual CPUs, not one. Some cloud providers don't speak about the number of CPUs, but this may be an advantage for certain applications.

There isn't a lockstep connection between the amount of RAM and the number of CPUs as you march down the price list, but it's roughly correlated. By the time you reach a 32GB machine, you also get eight virtual CPUs at a price of \$1.28 per hour (or 64 cents during the private beta sale).

The machines toss in additional disk space. I don't know if this is as important because HP offers other ways of storing information. The virtual machines are disposable, and you shouldn't plan on doing much besides using the disk on them as a cache. You have to back up everything -- HP has a number of options for that.

## Ephemeral machines and persistent storage

The traditional idea has always been to separate data storage into a Web service that stands apart from your machine. Amazon S3 pioneered the idea of creating a storage service where you can park your bags of bits. The system is meant to be inde-

## Deep Dive

pendent from the compute nodes. Any computer can request copies of any bag of bits whenever it's needed.

HP has its version of Amazon S3 that it calls HP Cloud Object Storage. Like Amazon, HP provides a RESTful API for storing and retrieving files.

As part of the Web management GUI, HP also provides a Web interface for organizing these objects into containers and controlling who can see them. You can upload your files directly from here, then turn on public access. The browser computes the URL for you. It's a nice feature that makes developing and debugging a bit easier than in Amazon, for example. You're not always scrounging around for an FTP password.

HP also included a few tools that make building bigger networks easier. For instance, there's a way to predefine which ports will be open on your new machine, saving you the time of logging into each machine to run a script. HP will load up the right public keys so that you can log in quickly if needed.

The HP Cloud comes with a CDN (content delivery network) built into this object store. If you tap the right button on the Web GUI, the data in a particular container is pushed out into what looks like Akamai's network, at least judging from the URLs. The data is available via an HTTP or HTTPS URL, both nicely added to the Web interface.

The prices begin at 12 cents per gigabyte per month for basic storage and drop as you store more, although the price breaks don't start until you squirrel away at least 10 terabytes. You'll also be billed a penny per gigabyte for every 10,000 requests to store, fetch, or copy your information.

The bandwidth going into the storage cloud is free, but it will cost you to get your data out. After the first (free) gigabyte, it's 12 cents per gigabyte. If you turn on the CDN, the prices jump to 16 cents in North America and more elsewhere.

The object storage is nice, but the most intriguing feature to me is the persistent block storage, a feature that's in a private beta testing phase. The plan is to create virtual persistent disks you can mount on your virtual machines just like a real disk.

With HP Cloud Block Storage, you write to the file system and HP's machines will do the rest. You don't need to write new storage code that only works in the cloud. You just take the code that works with your software and point it toward the block storage partition. HP touts this as a way to easily move information between your instances or keep the data handy when there's no machine using it. You might put your database in the block storage and only connect it to a running instance when you need it.

**Compute — US West Availability Zone 1**

Servers Floating IPs

Create Server(s)

Flavor standard.xsmall

Install Image 1234 - CentOS 6.4

Get Compute API Keys

SecurityGroups

Create

**Instance: 415757**

To access via SSH:

1. Locate the private key for the key pair named **doug1**
2. For SSH to work, ensure your key is protected:  
`chmod 400 doug1.pem`
3. Open your SSH client (such as Terminal).
4. Connect via your server's IP address:  
15.185.101.66

**Example**  
`ssh -i doug1.pem root@15.185.101.66`

**Running Instances**

Status	Instance	Flavor	Image	Key Pair	Private IP	Fixed Public IP	Age
Active	415757	standard.xsmall	1234 - CentOS 6.4-bit20111207 (deprecated)	doug1	10.5.59.9	15.185.101.66	less than a minute

Connect

HP Cloud shows close attention to the details with a clean Web-based GUI that pops up helpful pointers.

## Deep Dive

### MySQL as a service

Also in private beta is an OpenStack-optimized version of MySQL as a service. HP promises to handle the backups and replication for you. You just pay per “use.” I think that offering actual MySQL as the service is better than the generic versions of SQL you get in Amazon and some other clouds. The HP Cloud’s MySQL will behave like the MySQL on your test machines. It will also be portable. If you need to move your code somewhere else, it’s bound to have MySQL. A proprietary SQL service just locks you in.

What happens if any of these fail? HP is offering a fair amount of security. The object store, for instance, makes three copies of every object and pushes them into different zones, each with backup power and dual Internet connections. HP Cloud handles the work for you. Some other services, such as [Google’s Compute Engine](#), leave the replication across zones up to you -- then bill you for bandwidth in between.

There is a big difference in the tone of HP’s marketing. Google’s literature might have been written by engineers, lawyers, or -- an even more careful and paranoid group -- engineers who went on to law school. Google’s documentation continues to play up the potential failures in data centers and talks about potential failures and downtime for maintenance. It’s your job to plan ahead.

HP Cloud seems more optimistic. There aren’t as many options for customizing how your data is protected, and that’s probably a good thing for users who just want to store their data and trust HP to replicate it three times.

If you want reassurance, you can sign a service-level agreement with HP, which offers [basic terms](#) on the HP Cloud website. If some data can’t be found for a few minutes, HP will start offering service credits up to 30 percent of the bill.

HP is adding some other services to round out the product line. For starters, it has [well-structured libraries](#) of Ruby, Java, Clojure, PHP, and .Net routines, and these are finding connections in other places. There’s a Drupal plug-in for dumping your Drupal data directly into HP Cloud’s object store. HP is also partnering with companies such as [New Relic](#), the makers of a performance monitoring toolkit already integrated with HP Cloud.

HP Cloud is playing catch-up. The more established players such as Amazon and Google have much more elaborate lists of options. But HP Cloud offers much of what a growing firm might want: OpenStack machines at an introductory beta price that’s quite good. HP is counting on open standards and easy portability being the carrots that attract people looking for well-priced machines without any lock-in. That’s a good mix.

## Deep Dive

REVIEW:

# Joyent Cloud is built for speed

*Joyent's very smart OS proves that some cloud servers are better than others*

BY PETER WAYNER

**The sales pitch** for servers in the cloud has always leaned on the word “commodity.” You push a button, and voilà, a root password is yours within minutes. The machines in the cloud may not be exactly what you would order if you were filling out a P.O. for your own metal box, but they’re probably close enough. The number of options may not be great, but you can choose among enough standard sizes and enough standard operating systems to get approach the ideal. In return for limiting your options, you can click a few buttons and run a machine in less than 180 seconds.

Joyent is a commodity cloud provider, but with a twist. Joyent Cloud offers many of the same basic machine options and standard distro choices you’ll find on Amazon and other clouds, but you can also try what Joyent has waiting behind door number two. In addition to Linux and Windows VMs, the company builds custom machines with not-as-common operating systems and calls them appliances. If you decide to go with Joyent’s machines, you might find they can be dramatically faster than the commodity server next door, at least when performing standard tasks like answering requests for basic Web pages or data from a database.

The not-so-secret sauce that Joyent is pitching is called SmartOS. You may know it as Illumos or more likely as Solaris, its most prominent name. Around the time that Sun merged with Oracle, a number of engineers from the Solaris team decamped for Joyent. They forked some code from the OpenSolaris project, called it SmartOS, and began renting cloud servers running the new operating system. The name changes aren’t sticking very well because the SmartOS documentation will occasionally refer to the operating system as Solaris. But it’s not worth getting caught up in the

name. If you liked Solaris and mourned the death of Sun, this is your chance to keep the flame for just a few cents an hour. Nostalgia has never been cheaper.

But the reasons for choosing SmartOS have little to do with your fond memories of the ‘90s and the heady days of Sparc chips. Many continue to believe that Solaris is a great choice for an enterprise server, especially one with multiple cores. The file system, ZFS, offers a better transactional system that only locks in the disk changes when the data is completely written. The OS also offers a feature to segregate users, called “zones,” that dramatically lowers the resource requirements behind the good fences that make good neighbors.

If you choose a Joyent SmartOS server from the cloud, you get all these features and a few more like DTrace, a debugging tool that can help you pinpoint what’s slowing down your software. DTrace will let you watch the latency of writes to disk and make sure you’re getting your money’s worth from the platform.

Turning Solaris into a cloud operating system has paid off. Joyent emphasizes that its virtualization technology is so tightly integrated with SmartOS that SmartOS guests are connected directly to the device drivers of the machine. There’s no virtualization layer in between.

Joyent’s virtualization tool also boasts a few other neat tricks. You can, for instance, resize your virtual machine without rebooting it -- assuming the underlying server has room to spare. Joyent also promises that the virtual machine can use the extra capacity temporarily in a burst mode.

## Virtual bare-metal performance

Joyent crows that its machines often perform

## Deep Dive

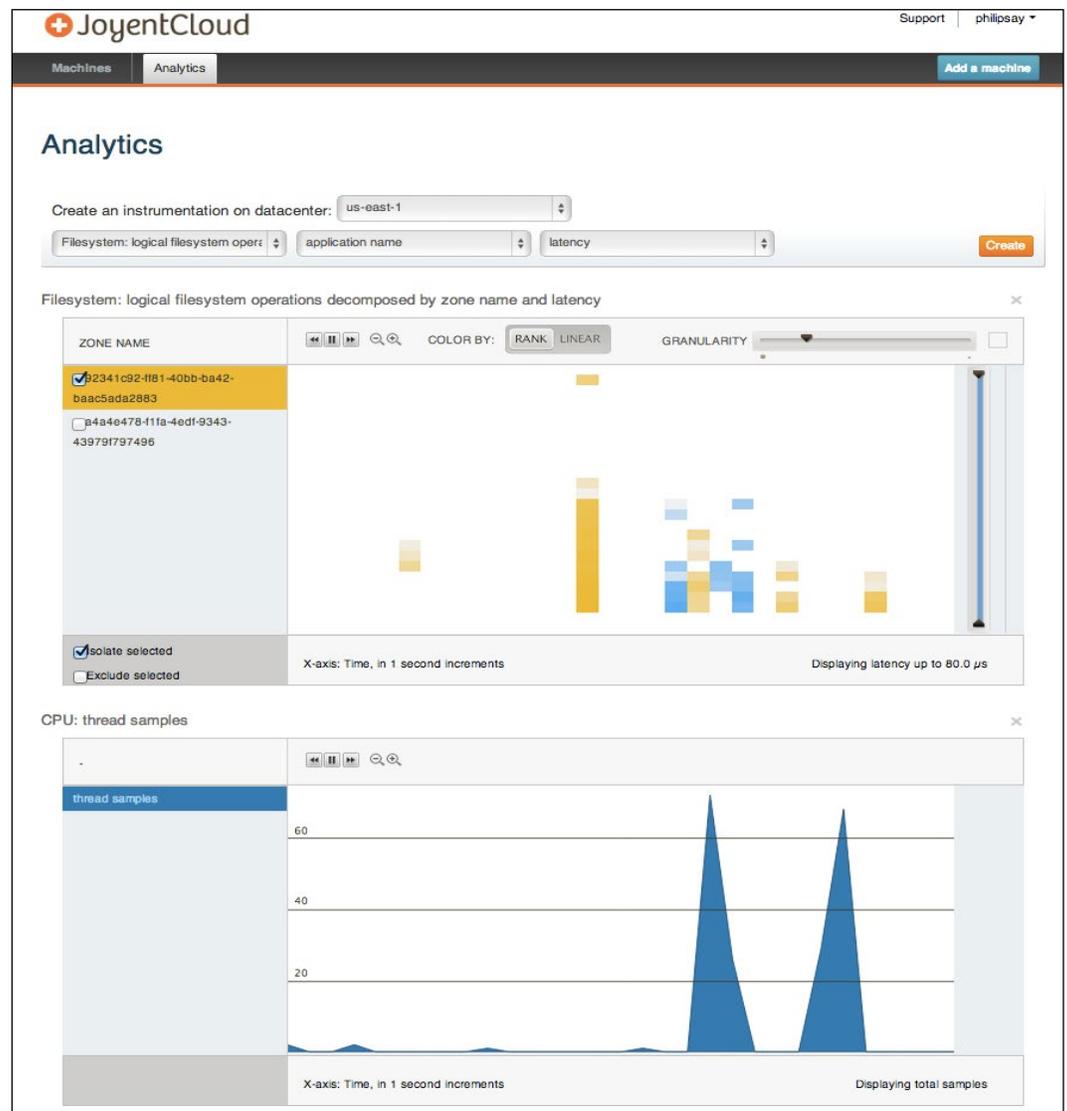
better than others because of the thinness of the virtualization layer, which it calls “bare-metal performance.” The company suggests it’s especially useful when systems are running virtual machines such as Java, PHP, and Node.

To test this out, I created a low-end, 512MB machine at both Joyent and Rackspace. Joyent charges 3 cents per hour for such a machine, while Rackspace charges 2.2 cents. I ran a number of [DaCapo benchmarks](#) for Java. The SmartOS came with Java 1.6 already installed. I had to install OpenJDK 1.6 on the CentOS box I built at Rackspace.

Most of the benchmarks ran between two and three times faster on the Joyent box. Some that used a heavy amount of IO -- such as the luindex

benchmark that builds a Lucene index -- ran slightly more than three times faster. The avrora benchmark that simulates microcontrollers ran at about the same speed on both servers.

At the beginning, the Joyent box was often slower than the Rackspace box on the multi-threaded benchmarks because it created 24 threads. Why? SmartOS poked around in the machine and decided it could run 24 threads because it was running on a 24-processor machine. When I limited the benchmark to one thread, the Joyent machine sped up dramatically (see table below). Even for the multithreaded tests, the Linux-based Rackspace machine automatically chose one thread.



Joyent’s Analytics page makes it easy to configure graphical views of system performance.

## Deep Dive

These results show that for all of the talk about commodity servers, there are plenty of differences between the off-the-shelf machines. It pays to spend some time testing your particular application, then tuning it -- the operating system, the level of disk access, and the level of network access can affect the speeds dramatically. What may be good for one application may not benefit another.

### Fast ride in a Web machine

Joyent is not relying on SmartOS alone. The company also offers pre-loaded "appliances." When you click through the options for your brand-new box, you'll see a number of machines that come ready to run packages such as Node.js, Percona, Riak, and MongoDB. You push the button and get a machine with all of the files in the right places, so you can start running the services immediately. These can certainly save you time starting up, but they're not as simple as they could be. After you push the start button for a Node box, for instance, you still get to log in and do everything with the command line.

Joyent is pushing Node.js with as much fervor as it's pushing SmartOS. The company cites the incredible performance that some folks see with putting lightweight jobs in the hands of the JavaScript server. A basic Node appliance can easily handle thousands of clients because well-written Node code can juggle the incoming requests with little overhead.

You don't need to run Node.js and SmartOS to want to use Joyent. Joyent Cloud offers all the standard distros at prices that seem, on the face, to be 15 to 25 percent cheaper than Amazon Web Services. On paper, at least, the machines appear to be basically the same with equivalent specs.

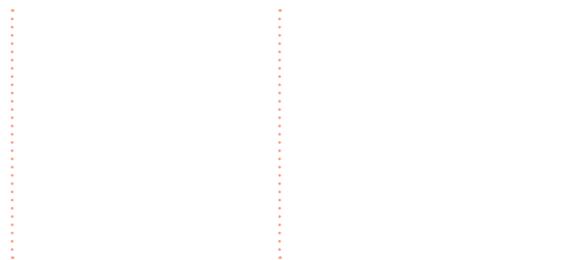
But as the simple tests above show, the same

basic box with the same specs can perform wildly better or worse depending upon how it interacts with the hypervisor. The amount of RAM, the number of CPUs, and the amount of disk space can't capture the effect of using SmartOS as a virtualization layer. Joyent's engineers claim their machines will run some software dramatically faster. They even like to claim they run Windows faster than Windows. Of course you should test your code before you believe.

Like some other clouds, Joyent offers open source to assuage the worries of the IT manager who fears being locked into the technology. You can download [SmartOS](#) and [Node.js](#) if you want to move your servers from the Joyent Cloud. Joyent will also host a "private cloud" for you in its data center. Joyent does the hosting and the provisioning of the SmartOS. You write the checks.

In the end, Joyent is offering a different kind of commodity than most of the IaaS cloud providers. Joyent is selling you a machine, just like they are, but Joyent's appliances are engineered to deliver Web pages and database storage as a service. Joyent has worked extensively on streamlining the connection between the incoming HTTP request and the bare metal. All that stands in between are Node and SmartOS, and both can be incredibly efficient in the right hands.

Joyent is essentially commodifying at a higher level: the HTTP processing level. If you're looking to process lots of basic requests, then you'll be attracted to a simple stack like the Node.js appliance. If you're looking on a cleanly engineered machine built on one of the more secure and well-engineered operating systems, you'll want to look at SmartOS. They're not the standard commodity tools, but that's not bad if you're looking for something more.



## Deep Dive

REVIEW:

# Windows Azure shoots the moon

*Microsoft's cloud wows with great price-performance, Windows toolchain integration, and plenty of open source options*

BY PETER WAYNER

**A long time ago** in a century slipping further and further away, Bill Gates compared MSN with the exploding World Wide Web, saw the future, and pivoted nicely to embrace the Internet. A few decades later, someone at Microsoft looked at the cloud and recognized that the old days of selling Windows Server OS licenses were fading. Today we have Windows Azure, Microsoft's offering for the cloud.

Azure is a cloud filled with racks and racks of machines like other clouds, but it also offers a wider collection of the building blocks enterprise managers need to assemble modern, flexible websites. There are common offerings such as virtual machines, databases, and storage blocks, along with not-so-common additions such as service buses, networks, and connections to data farms address verifiers, location data, and Microsoft's own Bing search engine. There are also tools for debugging your code, sending emails, and installing databases like MongoDB and ClearDB's version of MySQL.

All of these show that Microsoft is actively trying to build a system that lets developers easily produce a working website using the tools of their choice. Azure is not just delivering commodity Microsoft machines and leaving the rest up to you -- it's starting to make it simpler to bolt together all of the parts. The process still isn't simple, but it's dramatically more convenient than the old paradigm.

## Not-only-Windows Azure

The Azure service is a godsend for those who are heavily invested in Microsoft's operating systems. Many of the big clouds offer only Linux or BSD machines. Rackspace charges 33 percent more to

build out a Microsoft Windows server, but Azure rents a Windows machine at the same bargain rate as Linux.

Did I say the same as Linux? Yes, because Microsoft is fully embracing many open source technologies with Azure. You can boot up a virtual machine and install a few of the popular Linux distros like Ubuntu Server 12.04 or OpenSuse 12.1. There aren't many choices of open source distros, but Microsoft has chosen a few of the more popular ones. They cost the same as the standard Windows Server 2008 R2 and Windows Server 2012 offerings.

Microsoft's embrace of open source is on full display with Azure. The company is pushing PHP, Node.js, Python, Java (if you consider Java open source), and even [MySQL](#). Well, that's not exactly true. You can create running versions of Drupal or WordPress, and Azure will set up MySQL back ends for you. If you go to the SQL tab to start up your own SQL database, you can provision an instance of Microsoft SQL Server, but there's no mention of MySQL. That's because Microsoft is letting a third party, ClearDB, deliver MySQL. It's one of a dozen or so extras you can buy.

The websites with Drupal or WordPress are among many options available. Microsoft will let you have up to 10 free ones with your account. Then you push your HTML or PHP to them with Git, and the server does the rest. (Notice the [embrace of Git](#) too.)

These free options are come-ons. If your website takes off and you start getting traffic, you can upgrade to shared services or full, managed machines that can be load balanced. The documentation is a bit cagey about what happens as you start fiddling with the Scale control panel, but you get better guarantees of service and less throt-

## Deep Dive

ting. If you move over to the Reserved setting, you get dedicated virtual machines with resource guarantees. This is a pretty simple way to build and test a website before deploying it for production.

### Tapping the Microsoft toolchain

Building a simple website tests only a small part of what's available. Microsoft offers the largest smorgasbord of treats for enterprise programmers of any of the clouds I've seen -- no doubt because the company is also one of the dominant programming tool creators. Microsoft has connected the services with tools in the Azure platform so that you can create, for instance, a "mobile service" just like you're booting up a virtual server. While the other clouds are selling a commodity machine, Microsoft is trying to take it a bit further.

The integration isn't as seamless as it could be. But while it leaves much of the work up to you, it's neat to click a few buttons and download a Visual Studio project with the scaffolding for storing data in Azure. You must still use Visual Studio to create the app, but Microsoft is trying to make it simpler to create a Windows 8 smartphone app that stores its data with Azure.

What's even more surprising is the iOS tab next to the Windows 8 tab. If you click there, you can download an Apple Xcode project file for your Mac. Your Windows back-end service can also support an iOS app. Microsoft is not just embracing open

source code, but the new Borg, the iPhone.

Despite this detour, most of the options and much of the fun are reserved for Windows and Microsoft SQL programmers. You can create a cloud service wrapped around a database with Microsoft's SDK. Then you can deploy it first to the staging server, then push it into production. It's a pretty nice infrastructure. There's quite a bit of depth to the SDK too. It's not possible to cover all of it here, a feeling I often had while experimenting with Azure.

When you create your own Microsoft SQL databases with Azure, you pay by the gigabyte. The databases start at \$5 per month per 100MB, but the cost per byte drops quickly. A 10GB database is only \$45.96.

It's worth mentioning that you don't need to work with Microsoft's pay-as-you-go plan. Microsoft gives fairly substantial discounts (20 to 32 percent) if you make a commitment and pay in advance. Should you go over what you promised to buy, Microsoft bills you for the rest at the standard rates. The monthly commitment deals kick in only when you promise to use \$500 worth of services per month.

SQL isn't the only option -- Microsoft remains current here too. If you like, you can house your info in blobs, tables, or queues in a data store that can either be locally redundant or "geo redundant." In other words, it can be replicated in one data

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
portalvhds453t5s19nvj7d	Storage Account	✓ Online	Pay-As-You-Go	West US
frogdog2	Web Site	✓ Running	Pay-As-You-Go	West US
bobster	Web Site	✓ Running	Pay-As-You-Go	West US
frogdog	Web Site	✓ Running	Pay-As-You-Go	West US
fred33	Web Site	✓ Running	Pay-As-You-Go	West US
frogdog2_db	Database	✓ Online	Pay-As-You-Go	West US
frog	Mobile Service	✓ Ready	Pay-As-You-Go	West US
frog_db	Database	✓ Online	Pay-As-You-Go	West US
fredsky	Cloud service	✓ Created	Pay-As-You-Go	West US
froggy	Storage Account	✓ Online	Pay-As-You-Go	West US

Your list of Azure virtual machines and services can grow pretty quickly.

## Deep Dive

center or across the country. The sizes can be pretty big. As an example, 1,025GB of geo-redundant storage costs \$128.12 per month. Microsoft will also charge you for each “storage transaction,” but these seem fairly cheap. The price calculator quoted me 90 cents for 9 million. Does anyone else offer pricing in cents any longer?

### Benchmarking Azure performance

Are the Azure prices good? I decided to take the bargain-basement machine out for a spin with some Java benchmarks. While Java may or may not be the most popular language used on these machines, Microsoft’s extensive documentation includes instructions on building a Java machine for doing computation. They list it alongside C#, Node, and PHP.

The “extra small” machine with 768MB of RAM and a shared core sells for just 1.3 cents per hour or \$9.36 per month. That “preview” pricing is cheaper than that of any other cloud machines I’ve seen -- including Rackspace’s 256MB “first generation” machine that sells for 1.5 cents per hour. This isn’t an exact comparison because that price only gets you a Linux box. Rackspace’s Windows machines

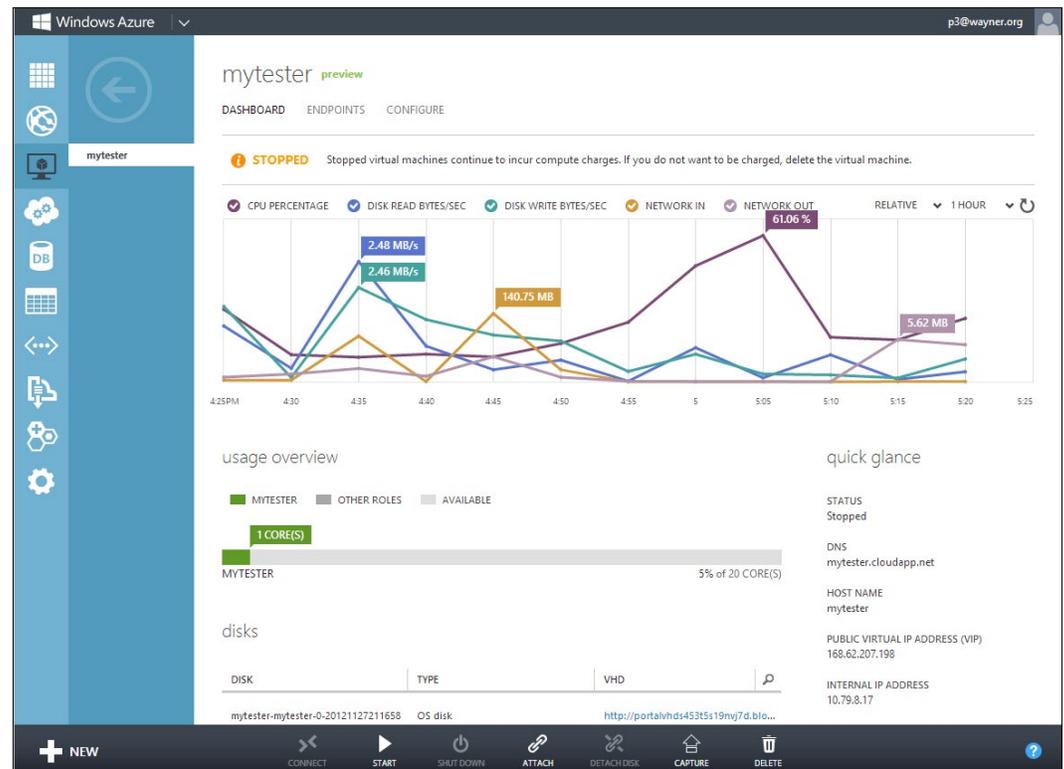
are more expensive.

I loaded the latest Oracle JVM onto a Windows Server 2008 R2 image and started up the [DaCapo benchmarks](#). The performance was wildly different from other low-end servers I built at [Rackspace](#) and [Joyent](#). The differences between so-called commodity machines continue to surprise me.

The avrora simulation of parallel events, for instance, was more than twice as fast on the Azure machine than on the Joyent or Rackspace boxes, but the Tomcat server benchmark was two to five times slower. The Lucene indexing benchmark was dramatically slower on Azure, but the Lucene search of an index was dramatically faster. All of this proves you have to test your own code on each platform if you want to squeeze out optimal performance.

Should you purchase? The tested speed of the machine may vary according to the individual benchmarks, but the overall results are in the same ballpark. If you’re looking for a cheap machine, it’s a good price.

I don’t know how long Microsoft plans to maintain these preview prices, but they’re extremely tempting, especially for anyone who’s looking to



Azure provides an easy way to track the load on your server.

## Deep Dive

move code that needs Microsoft's Windows or SQL platform. You can spin up what you need quickly and move your software over without rewriting anything. The prices are also pretty compelling for non-Microsoft users. You can use the most high-profile distributions of Linux and enjoy the same hardware configurations and price points.

### **Beyond the commodity cloud**

The machines and the software running on them may be the heart of the cloud, but Microsoft's Azure platform is more than just Windows machines and Microsoft SQL. You can also purchase various data services through APIs that handle a few of the most common data management jobs that Web developers confront.

The integration is mainly just the bookkeeping. If you purchase the Bing Search Services (the first 5,000 searches a month are free), you get a pointer to a proxy C# class and code that makes the API call. The source code snippet comes with a cute little link that will insert your API key. What you do with the search results is up to you and the C# code

you write.

There are at least a dozen different options, including an address checking service that compares the address your customer entered with the officially recognized addresses from the postal service database. Another looks up sales tax rates. A third offers weather data. It's a nice mix that can save you the trouble of building these services yourself.

Windows Azure shows how a company that built a product like Visual Studio approaches the cloud. Microsoft didn't just create a bunch of APIs and shell scripts, but knitted everything together with a clicky Web interface. Rather than use Vi to knit together Unix scripts, you point and click. It's a nice way to advance the platform and make it simpler for people to create scalable websites that offer a full range of services. This budding integration with the Microsoft toolchain is the frosting on the cake that cast the Azure machines with their enticing prices as more than just commodities.

## Deep Dive

REVIEW:

# Rackspace Cloud keeps IaaS simple

*Rackspace stands apart with familiar tools, open standards, and enterprise-grade support*

BY PETER WAYNER

**Rackspace was one of** the first players in the cloud arena. The company recognized early that enterprises wanted faster, simpler ways to spin up and spin down servers. If the bosses are going to be fickle and impulsive, there will always be a market for companies that make it easy for the people curating the data to pivot. If the corporate vision is going to morph, the IT shops will want a way to morph with it.

At Rackspace, the meaning of “cloud” has always been a bit simpler and more straightforward, and the philosophy a bit more open and pragmatic, than at other cloud providers. While some of the others spun elaborate metaphors, abstracted away the old files, and portrayed the opaqueness of their mechanism as a feature, Rackspace sold real instances that felt more like real computers. From the beginning, Rackspace’s cloud was just a fast way to buy extra machines for an hour, then turn them off.

Lately the company has been adding new products and features to create what it calls the Next Gen cloud. You can still access the First Gen cloud and use the original cloud software, but it won’t offer all of the new features such as better data storage, public IPv6 support, and the ability to change a server’s metadata.

## **Rackspace Cloud: The next generation**

While there are many new features, Rackspace is still largely selling machines (virtual machines, to be more precise), but now you can glue them together in a few additional ways. The data can be squirreled away in either block storage or containers, two abstractions that aren’t permanently glued to the servers. For MySQL users,

Rackspace has built a stripped-down and tuned machine image that delivers better performance. The company has also provided off-the-shelf load balancers and backup; adding these features has become much simpler.

Rackspace has also upgraded the machine choices and in one sense lowered the price. The First Gen cloud offered (and still offers) old, anemic machines with just 256MB of RAM and a monthly cost of \$10.95, but that bargain-basement offering is gone from the Next Gen cloud. The low-end machines in the Next Gen cloud now come with 512MB of RAM, 20GB of disk space, one virtual CPU, and a monthly cost of \$16.06. Bargain users are grouching about the increased cost of running very lightly taxed machines, but those using the larger machines will see prices dropping. A first-generation machine with 512MB of RAM costs 3 cents per hour, while a second-generation machine with 512MB of RAM costs only 2.2 cents per hour.

You’re not just paying for RAM with the next generation. The larger, more-expensive machines with 2GB of RAM or more now come with additional virtual CPUs that go along with the extra storage. By the time you’re purchasing 30GB of RAM for \$876.60 per month, Rackspace is tossing in eight virtual CPUs with the package. You get more power for your money too.

The biggest change in the second generation is the file storage that can now live separately from your servers. In the past, the instances were more like real boxes. If you wanted to get the data on and off the machines, it was up to you. Now Rackspace offers storage blocks that are configured separately from your virtual machine. They’re mounted like an external disk drive, and you can use them to read and write data apart from your server image.

## Deep Dive

The block storage API is pretty transparent. After I pushed the button to create the block storage and attach it to my server, I needed to partition, format, and mount the space using the operating system. It's like attaching a storage device to your own physical machine. You get to push the buttons yourself.

Rackspace also offers a very fast SSD option if you want to pay for a bit more speed. The old-fashioned SATA storage is 15 cents per gigabyte per month, while the solid-state storage is 70 cents per gigabyte per month. If you need only a small amount of fast storage, it's a good choice.

There are hints that Rackspace is embracing some of the more amorphous visions for the cloud. If your data needs to reach a wide audience, you can have a slice of Akamai's Content Distribution Network called CloudFiles. You store your data in CloudFiles and Akamai will deliver it faster. Storage is 10 cents per gigabyte per month, while outgoing bandwidth is 18 cents per gigabyte.

### The OpenStack advantage

Rackspace was one of the main forces behind [OpenStack](#), and the company continues to make the flexibility of the open source cloud stack one

of its big selling points. Your enterprise won't be locked into the Rackspace cloud because you can always set up OpenStack in your own data center -- if and when you want to leave Rackspace behind. This flexibility is crucial for many businesses because rewriting code can be quite a pain, especially if it's older code that someone wrote long ago before quitting.

Rackspace brags that it's not just offering you a separate fork of OpenStack, but the actual code running on its cloud machines. "The software is not a separate distribution of OpenStack, so you don't have to worry about a branching dead end," promises the Rackspace website. This is quite an offer and one that's designed to prey on any enterprise manager's worst fears of vendor lock-in.

Rackspace also pushes hybrid architectures that make it possible for you to link up your private cloud with the Rackspace cloud for moments when you need more servers. One customer, for instance, says it turns on servers in the public cloud whenever it runs major ad campaigns. When the traffic surge is over, the website retreats to the private cloud. Running the same OpenStack layer in your data center makes it simpler to do this.

Rackspace is taking a different approach to

The screenshot shows the Rackspace Cloud Control Panel interface for creating a new server. At the top, there's a navigation bar with the Rackspace logo and various service categories like Servers, Load Balancers, Files, DNS, Databases, and Backup. Below this, there's a sub-navigation bar for Cloud Servers, Saved Images, Block Storage, and Storage Snapshots. The main content area is titled 'Create Server' and is divided into sections for 'Identity' and 'Image'. In the 'Identity' section, the 'Server Name' is set to 'gamma' and the 'Region' is set to 'Dallas (DFW)'. The 'Image' section shows a list of available operating system images, including CentOS 6.3, Debian 6 (Squeeze), Fedora 16 (Verne), Fedora 17 (Beefy Miracle), FreeBSD 9, and Gentoo 12.3. On the right side of the page, there's a sidebar with sections for 'Setting Up Your Server', 'HELP ME WITH...' (with links to 'Choosing the Right Image and Size' and 'Managing My Server'), and 'WHAT'S NEXT?' (with links to 'Logging in to Your Server' and 'Configuring Basic Security').

**From CentOS to Ubuntu, all of the usual suspects are available for creating a Linux or BSD machine.**

## Deep Dive

storing data. While Amazon, Google, HP, and a few others are building elaborate NoSQL abstraction layers that promise elastic scalability, Rackspace is sticking with [MySQL, the decades-old, tried-and-true solution](#) to storing information.

This is a great solution for developers with any legacy code. It's easy to fire up a new project and begin with a [fun, new NoSQL storage engine](#) when you're starting with a blank sheet of paper, but it's much harder to take running code and convert it. I was able, for instance, to get Drupal up and running in less than five minutes because Drupal relies upon MySQL to store all of the data. I didn't need to rewrite the Drupal code to work with some new object or document store. There was no weird glue code or translation architecture. I just started up a MySQL database and pointed the Drupal code at the URL.

The separate MySQL option came around because the engineers at Rackspace listened to customer complaints that the performance of databases often wasn't as good as it could be. The virtualization layer used in clouds like these added delays in writing and reading from the I/O, a factor for operations like running a database. The device driver for your virtual machine won't write directly to the disk, but will shove the data into shared memory and wait for the underlying machine OS to actually write it to storage. That may be an

acceptable price to pay for some applications, but not for code that lives to store data to a hard disk.

Rackspace's solution is to eliminate some of the hardware and operating system layers, which the company calls "container-based virtualization." You can't log into your MySQL database server and configure the underlying OS. You get only a URL and a MySQL user password; all of your interaction happens as a MySQL user, not as a regular Unix user.

### Protecting your cloud data

Rackspace has added extra redundancy out of sight. The version of MySQL isn't running on any old machine, but writing to a SAN with RAID hardware. Rackspace then enables further protection by copying the data to other machines on the network. All of this replication happens at the hardware or network level, not with MySQL. Rackspace doesn't currently use the MySQL replication code, although it promises to offer that in the future. The company also promises to offer yet another layer of protection, an automated backup tool for taking snapshots of your database.

The cloud is a bit of a departure for Rackspace, at least given the price. The company built its name on offering great hand-holding support at premium prices. While the cloud instances are priced like commodities, you can still spend money if you need to. If you want to purchase a

The screenshot shows the Rackspace Cloud Control Panel interface. At the top, there's a navigation bar with 'Servers', 'Load Balancers', 'Files', 'DNS', 'Databases', and 'Backup'. Below that, there's a sub-navigation bar with 'Cloud Servers', 'Saved Images', 'Block Storage', and 'Storage Snapshots'. The main content area is titled 'Block Storage Volumes'. A 'Create Volume' modal is open, showing the following configuration:

- Volume Name: InfoWorld Volume 3
- Region: Dallas (DFW)
- Volume Type: Standard (SATA) (Regular hard disk storage)
- High-Performance (SSD) (High-speed solid-state storage) is also visible but not selected.
- Size: 100 GB (100-1024 GB. Can't be resized later.)
- Note: Volumes must be attached to servers in the same region.
- Price: \$15.00 / mo.

In the background, a table shows existing volumes:

Region	Type	Size
Dallas (DFW)	Standard (SATA)	100 GB

**You can choose between disk storage and faster SSD when you create a block of storage that can be mounted on your virtual machine.**

## Deep Dive

Cloud Site, one of the products sitting next to the regular cloud servers, it will cost \$150 a month. That's dramatically more than the \$5 per month that some low-cost providers charge for website hosting, but it includes Rackspace's trademarked "fanatical support." A Cloud Site also comes with fixed limits on bandwidth and storage, which the low-cost servers pretend don't exist when they claim it's all unlimited. Of course the low-cost sites are fibbing -- nothing is unlimited.

There are a number of other ways to buy premium products or premium support. All the cloud machines are available with Rackspace support under a separate tab called Managed Cloud. Almost every product Rackspace offers at a commodity price is also available with hand-holding for more money. If your operation doesn't have the expertise inside or you just want to

arrange for an additional layer of people who can assist, you can sign up for support. Even if you're building your own private cloud using Rackspace's open source set of tools, Rackspace will offer to help you from afar.

This is the corner of the IT world that Rackspace has chosen: high-quality support married with commodity hardware and open systems. The company's sales literature pushes the idea that you should "Stay because you want to, not because you have to." The next generation of the Rackspace cloud offers more features and more options, but it stays true to this basic plan. It's more like running your own servers, the kindly Linux boxes you've grown accustomed to, and less like buying into a newfangled religion.

## Deep Dive

REVIEW:

# SoftLayer's cloud is fast and flexible

*SoftLayer brings fine-grained configuration options, high performance, and interesting extras to the self-service cloud*

BY PETER WAYNER

**The cloud has a** way of hiding much of what we used to fret about. Servers are boxes, and boxes are meant to be interchangeable. You push the button and you log in. It's just a box, and there's no need to spend much time thinking about it because it's a commodity.

SoftLayer is one of the companies fighting the commodification of the servers, at least a bit. SoftLayer is still selling servers by the hour and offering a cloud of machines that starts up on demand, but it's also making the server purchase more like it used to be. You have plenty of options, some of which include getting a raw machine that's yours, all yours.

Amazon and [Google](#), for instance, started selling a few basic models. Although they've expanded the selection over the years by adding higher-powered CPUs or more RAM, the menu of choices is still pretty simple. If you get a small machine, you get a small CPU with a smaller amount of RAM and a smaller bundle of everything else. If you want more, you buy more of everything.

SoftLayer lets you shop for servers the old way. You choose how many cores you want, then choose the RAM independently. You can build a machine with 16 2GHz cores and 1GB of RAM, one core and 16GB of RAM, or any integer in between -- say, 13 cores and 7GB of RAM. The prices slide up and down, and the two parts are priced independently. Sixteen cores will cost 75 cents per hour, while only one core will cost 7 cents per hour. There are price breaks along the list and it's not exactly linear.

## Higher performance, higher price

I ended up playing around with a low-end machine that cost 12 cents per hour. The single core cost 7

cents, the 1GB of RAM cost 3 cents, and the bandwidth (100Mbps) cost 2 cents.

This system was dramatically faster on the set of basic tests I've been running: the [DaCapo Java benchmarks](#) that test raw computation and simulate some common enterprise tools, including Tomcat and Lucene. Most of the tests were two to three times faster than even the better commodity machines from [Joyent Cloud](#) and [Microsoft Windows Azure](#) that I've tested. The Tomcat test was almost 10 times faster than Amazon's EC2 small instance and about 30 to 40 percent faster than Amazon's high-CPU model. There were plenty of variations among the different tests, though. It's impossible to generalize or reduce the speed difference to a single number.

It's clear that a serious customer should take the machines out for a test-drive with production versions of their software. Each machine is surprisingly different for something that's supposed to be a commodity. The comparison should also include basic accounting because the low-end machines I used have big price differences. A low-end Joyent machine is only 3 cents an hour; a low-end [Rackspace](#) machine runs 2.2 cents an hour. The SoftLayer machine is about three to four times more expensive at 10 cents an hour.

## Layering on the options

The hardware is only part of the shopping process with SoftLayer. While a number of cloud providers give you just a few radio buttons of options during the configuration process, SoftLayer takes you through four pages of choices. Four extra public IP addresses are 1 cent per hour. A premium monitoring package is 6 cents per hour. You can add five different disks to your server if you like. You can add

## Deep Dive

more local disk space or storage on a SAN. It's like the old days when the server salesman wanted to fill up all the bays with extras. If you want a license to run Windows, SoftLayer will toss one in for between 5 and 10 cents per hour depending upon which version of Windows you choose.

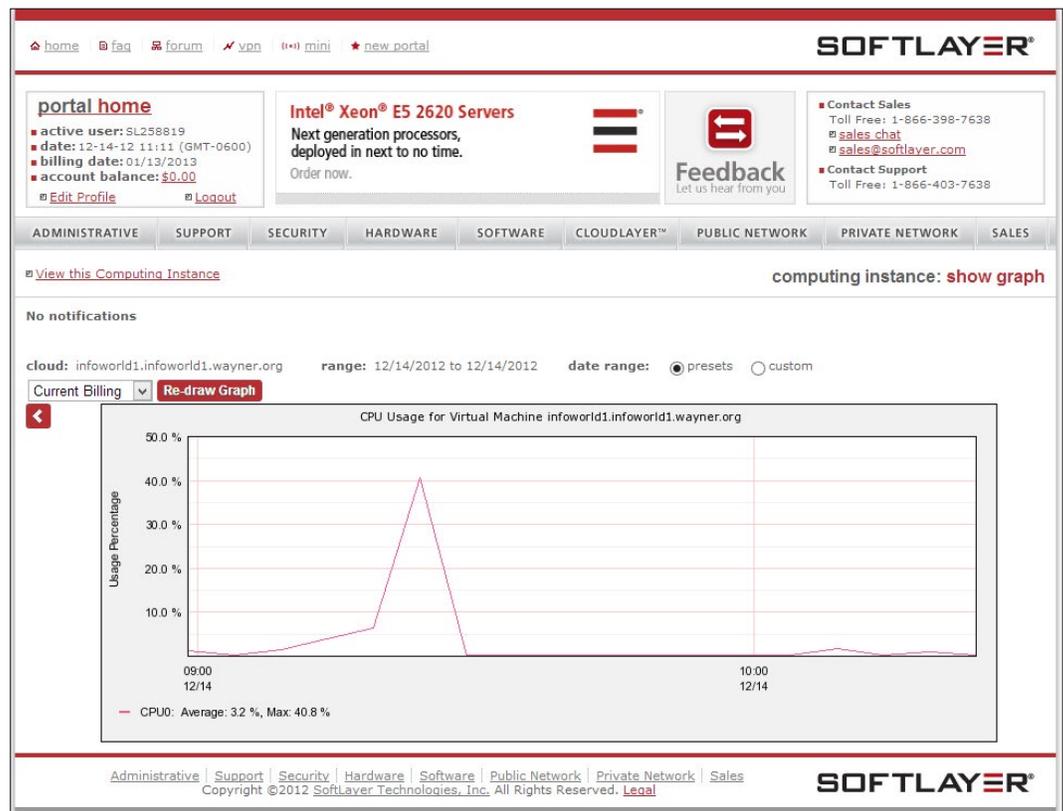
While I had no real problem configuring several machines, SoftLayer is still working through making all of this function smoothly. I asked for [MongoDB](#) on my machine but got a message later that it wouldn't work with Ubuntu 12.04, the OS I happened to choose. There are menu items on the portal for CPanel software licenses, but I wasn't given an option to buy one. SoftLayer is clearly planning on making it easy to buy extras, but not all of the dots are connected yet.

One interesting option is a "bare metal" server, also sold by the hour or by the month. I spun up one of these with two cores running at 2GHz and 2GB of RAM at a price of 50 cents per hour. These don't run in the same seemingly endless stack of virtualization, allowing them to access the I/O

channels faster. This pays off with databases and other disk-bound applications.

The performance of the "bare metal" server was often better, but not in every case. In some the results were largely the same as the "painted metal," for lack of a better term. The times were about the same for the relatively linear, single-process jobs like the Batik vector graphics rendering or the Eclipse test. These are largely computational. But other tests such as the Lucene searching or the Sunflow ray tracing sped up dramatically because the code was able to take advantage of the extra cores and the better disk I/O. The DaCapo benchmarks have an option to limit the number of threads, and when I held the bare-metal machines to one thread, the gains largely disappeared.

Your results, of course, will differ just as the results from the benchmarks do. The so-called bare-metal machines are better at handling I/O operations such as writing to disk because they don't have the hypervisor adding an extra step to the interaction with the device drivers.



**An example of SoftLayer's GUI. This graph shows the CPU load during testing. There are similar graphs for the full range of server performance statistics.**

## Deep Dive

### MongoDB in the cloud

SoftLayer is seeking to bottle this advantage in a different way. The company is creating its own bundles of bare-metal machines and installing MongoDB on them on top of CentOS. A monthly fee of \$359, for instance, buys a four-core machine that's ready to run. You can also purchase a support subscription from 10gen through SoftLayer. You pay for some of this expertise from the beginning because SoftLayer designed the server package with 10gen's guidance.

The MongoDB boxes are one of many offerings; a content delivery network, load balancers, and firewalls are also available. Plus, you can store your data as objects in SoftLayer's object store, which is built using a version of the OpenStack Swift object store. The metadata for the objects are indexed, making it a bit easier to find what you're looking to get. The object store's integration with the content delivery network makes it a bit simpler to serve up the same data again and again throughout the cloud. Storage is 10 cents per gigabyte, as is outbound traffic.

There's yet another interesting feature that's hidden from these endless menus of choices.

SoftLayer gives you your own private network for back-channel communications among your machines. Each server has one address for talking to the Internet at large and one for talking just to the private network. If you want to keep some servers in the background, out of view of the Wild West of the Internet, you can open up the ports on this private network. This channel makes it simpler to enforce some rules by locking out the public Internet in one swoop.

This adds up to a large collection with all the options you'll need to build out your server farm. The flexibility to pick and choose just how much memory and cores you need is much greater than SoftLayer's main cloud competitors, casting the entire process as a bit of a throwback. You're not grabbing a commodity block of computing time that's more or less the same as every other block. You're building out a server and adding extra features, all using prices that are measured by the hour or by the month. It's a welcome reminder of the flexibility the old server sales force used to offer the enterprise customer.