

InfoWorld DeepDive

THE PUBLIC CLOUD MEGAGUIDE

REVIEW: Amazon,
Microsoft, Google, IBM,
and Joyent clouds
compared



Deep Dive

CLOUD REVIEW

Amazon, Microsoft, Google, IBM, and Joyent

The top five public clouds pile on the services and options, while adding unique twists

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 once again 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 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."



Deep Dive

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 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 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.

Billing matters

Figuring out what you're paying, by the way, is growing increasingly complex, thanks to the different ways the cloud providers are offering discounts. In the beginning, you bought your time by the hour, and the cost of one week could be computed by multiplying the hourly rate by 24, then by seven.

Now you have options. Amazon, for instance, offers "reserved instances," a mechanism for rewarding the companies that make a one-year or three-year commitment to AWS. As they say, "The more you pay up front, the more you save."

Not to be outdone, Google offers its own discount mechanism where the price of a machine keeps dropping the more you use it. You don't need to make a commitment or to use the machine constantly. The price simply goes down as your usage goes up.

That's not the hardest part of the equation because the billing models aren't the only difference. Performance varies substantially between machines and between applications. The machines all look alike when you're typing into the command line, but the performance starts to change based on details like how much RAM you choose and whether your app is trying to write to disk.

The architectures and applications are different enough that the smart thing to do is

run benchmarks, then start multiplying out the costs. Maybe that extra RAM is worth it and maybe it isn't. Sometimes you'll want a cluster of N fully loaded machines and sometimes you'll want 4N cheap machines that cost only pennies an hour. It's not an easy decision. You'll need spreadsheets to figure it out.

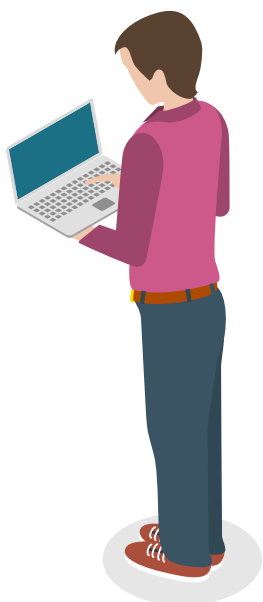
Key differences

Once you start to discover that performance varies, often dramatically, from cloud to cloud and that you don't always get twice the performance for paying twice as much, you start to realize there are many subtle differences behind the curtain.

Anyone who assumes that the cloud machines are merely 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. It's probably not worth your time to fret over saving a few pennies an hour if your job is going to end in a few hours, but the variations are often significant for anyone doing a large amount of work.

The deviations start when you choose 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 Cloud Platform](#), for instance, have their own Linux for the cloud.

Linux is not the only choice. Many of the clouds make Microsoft Windows available for an extra charge, but you often won't have to pay an additional fee with [Microsoft Azure](#), at least with some configurations. Microsoft wants to attract Windows shops with services that make it easier to move more and more computation to the cloud. Anyone with a substantial investment in Microsoft technology will feel more at home with Azure.



Deep Dive

	Amazon Web Services	Google Cloud Platform	IBM Cloud	Joyent Cloud	Microsoft Azure
COMPUTE SERVICES					
Max. CPUs per VM	40	32	16	32	32
Max. RAM per VM	244GB	208GB	64GB	256GB	448GB
Max. storage per VM	48TB	64TB	8.1TB	6TB	1TB
Custom machine types		X	X	X	
Dedicated hosts	X		X	X	
Bare metal			X	X	
GPU instances	X		X		X
Spot instances	X	X			
Reserved instances	X			X	
Autoscaling	X	X	X	X	X
Container services	X	X		X	X

Then there's [Joyent Cloud](#), which features a souped-up OpenSolaris derivative called SmartOS. It can offer faster performance for some jobs like Java, but SmartOS is mainly intriguing now because the Joyent engineers have rewired it to run Docker containers. It's an amazing work of engineering.

Docker is, of course, another big option because developers are starting to toss around Docker containers as they used to ship around virtual machines. All the clouds say they offer plenty of support, but there are subtle and not-so-subtle differences under the hoods. Joyent's option is good for bare-metal performance. Google, of course, makes it easy to use its Kubernetes tool to spin up clusters of containers. Azure lets you choose between Mesos/Marathon and Docker Swarm, and Amazon has its own way of scheduling containers. Running containers is a do-it-yourself project on [IBM Cloud](#) (aka SoftLayer), whereas you'll find container management services on IBM Bluemix.

While all of the cloud machines are meant

to appear to be indistinguishable from buying an Intel box and putting it in your own rack, the reality is they're often hefty multicore blades that are sliced up into virtual machines for your consumption. You're not renting a single family home on its own plot of land — you're signing up for a condo or a timeshare. The container code, the OS, and the hypervisor all enter into the game. The more layers, the slower the machine. The only way to know, though, is to test.

Data storage

One place where the effects of virtualization are 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.

All the clouds offer object storage. You give them a bag of bits, and they promise to return them later. It's not a bad way to store data. It's



Deep Dive

	Amazon Web Services	Google Cloud Platform	IBM Cloud	Joyent Cloud	Microsoft Azure
DATABASE SERVICES					
MySQL	X	X			
MariaDB	X				
PostgreSQL	X		*		
Oracle Database	X				
Microsoft SQL Server	X				X
IBM DB2			*		
MongoDB			*		
Other NoSQL	X	X	*		X
In-memory cache	X		*		X

* Available on IBM Bluemix

highly reliable and extremely flexible.

There are fancier options with more sophistication. IBM Cloud, for instance, offers MongoDB on separate machines that are specially tuned for the workload. Similar options are available in all the clouds with similar databases. Google offers four basic options like Cloud Datastore, a NoSQL product, and Cloud SQL, a version for those who still like SQL. Microsoft, of course, will host its very own SQL Server in Azure. Amazon offers the gamut of database services, including its [high-performance twist on MySQL](#).

Other services emphasize 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 but *hours*.

Then there are the more revolutionary approaches like Joyent’s Manta, a system that brings the computation to the data instead of the other way around. Joyent calls it a “distributed object storage service with integrated compute.” Your objects float out in the cloud. When you want to query them, you can write

code that runs locally. Manta supports R, Python, Node.js, Java, and a few more.

Of course, it’s still possible to grab another machine and install your favorite storage option, 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 IBM Cloud — offer private networks that link the machines. It’s easy to create a database machine that listens only 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, well, a cloudy subject, but it’s a great first start. For even more isolation from the proverbial noisy neighbor, you can go with IBM’s bare-metal instances.

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



Deep Dive

	Amazon Web Services	Google Cloud Platform	IBM Cloud	Joyent Cloud	Microsoft Azure
--	---------------------	-----------------------	-----------	--------------	-----------------

STORAGE SERVICES

CDN	X	X	X		X
Block storage	X	X	X	X	X
Solid state drives	X	X	X	X	X
Object storage	X	X	X	X	X
File storage	X		X		X
Archive storage	X	X	X		X

NOTE: Joyent's object storage service integrates compute, allowing the data to be processed in place.

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 it prices these transfers differently than the flows between machines in the same center.

Bandwidth costs have the potential to be confusing. The clouds typically charge nothing for incoming traffic, a trick that simplifies 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. If your goal is to compete with YouTube, on the other hand, you'll want to pay close attention to the egress rates.

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 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



	Amazon Web Services	Google Cloud Platform	IBM Cloud	Joyent Cloud	Microsoft Azure
--	---------------------	-----------------------	-----------	--------------	-----------------

NETWORK SERVICES

DNS	X	X	X	X	X
Load balancing	X	X	X		X
Virtual private networks	X	X	X	X	X
VPN gateways	X	X	X		X
Dedicated connections	X	X	X	X	X

Deep Dive

	Amazon Web Services	Google Cloud Platform	IBM Cloud	Joyent Cloud	Microsoft Azure
ANALYTICS SERVICES					
BI and visualization	X	X	*		X
Data warehouse	X	X	*		X
Stream processing	X	X	*		X
Elasticsearch	X		*		
Hadoop	X	X	*		X
Spark	X	X	*		X
Machine learning	X	X	*		X
Data pipelines	X	X	*		X

* Available on IBM Bluemix

are easy for these cards to execute. Physicists, biologists, and computer scientists are already converting their algorithms to run on these cards. It's only 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. All of these cloud providers 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 options for the teams doing data analytics. Microsoft offers a number of [big data crunching services](#) that are integrated with the Azure cloud. Once you upload your data, the algorithms are ready to go. You push a few buttons and fancy graphs and deep insights pour out. Similarly, you can tap the power of [Watson and predictive analytics tools](#) on IBM's Bluemix. Amazon offers a narrower set of [machine learning capabilities](#), tailored to developers and business analysts. Google's machine learning service was recently made available in a limited preview.

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 will 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 exactly 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



Deep Dive

	Amazon Web Services	Google Cloud Platform	IBM Cloud	Joyent Cloud	Microsoft Azure
DEVELOPER AND PLATFORM SERVICES					
Source code management	X	X	*		X
Code deployment	X	X	*		X
Continuous integration	X		*		X
Event-driven compute	X	X	*		X
Workflow	X		*		X
Business rules			*		X
API management	X		*		X
Mobile SDKs and back end	X	X	*		X
IoT platform	X	X	*		X
Message queueing	X	X	X		X
Push notifications	X	X	*		X
Email sending	X		X		X
Search	X	X	*		X
Media transcoding	X		X		X

* Available on IBM Bluemix

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 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 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've realized I'm not really finished. ■

Deep Dive

REVIEW

Amazon Web Services is eating the world

Amazon continues to define the cloud with an unrivaled set of services for developers, IT, and data crunchers

BY PETER WAYNER



Is it possible to review Amazon Web Services in one article? Not a chance. What about a book? Perhaps a long one, preferably with several volumes. The reality is that Amazon's cloud business is larger than ever and spawning new features, services, and options faster than any one person could begin to follow. The company is swallowing the Internet by delivering the simplest way to create complex, highly scalable, data-rich applications.

The scope of the project is amazing. There are, in my exploration, at least 10 different ways to store your data and four different ways to buy raw computation. If you need more than raw power, Amazon is moving up the stack by delivering cloud versions of many sophisticated tools for analyzing large data sets, like Hadoop, Spark, and Elasticsearch.

These tools are changing the game for programmers and data analysts, giving them fewer reasons to write fresh code and more reasons to link together different, high-end services from Amazon. While raw computing

power is still the focus, the new tools and services are compelling value propositions that can make good financial sense. Writing your own code gives you the freedom and the power to move elsewhere, but entrusting more and more of the stack to Amazon can be dramatically cheaper and faster. It's a complex decision.

A sea of machines

The core of the Amazon cloud remains the collection of virtual servers known as the Elastic Compute Cloud (EC2). If you want a machine, you can go to the AWS website, click a few buttons, and have root. More and more people, though, are using the API. Did I say "people"? I also meant bots because the cloud is more and more automated. If you're going to do more than start up a single instance for experimentation, you're better off writing code to spin up your machines. There are SDKs for Java, .Net, PHP, Python, and even [Google's Go language](#).

The range of machines that Amazon rents is growing larger and more complex. There are at

Deep Dive

Number of instances <small>(i)</small>	<input type="text" value="1"/>	Launch into Auto Scaling Group <small>(i)</small>
Purchasing option <small>(i)</small>	<input type="checkbox"/> Request Spot instances	
Network <small>(i)</small>	<input type="text" value="vpc-fb8fe49e (172.30.0.0/16)"/>	Create new VPC
Subnet <small>(i)</small>	<input type="text" value="subnet-1ea77535(172.30.1.0/24) us-east-1b"/> 251 IP Addresses available	Create new subnet
Auto-assign Public IP <small>(i)</small>	<input type="text" value="Use subnet setting (Enable)"/>	
IAM role <small>(i)</small>	<input type="text" value="None"/>	Create new IAM role
Shutdown behavior <small>(i)</small>	<input type="text" value="Stop"/>	
Enable termination protection <small>(i)</small>	<input type="checkbox"/> Protect against accidental termination	
Monitoring <small>(i)</small>	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy <small>(i)</small>	<input type="text" value="Shared - Run a shared hardware instance"/> Additional charges will apply for dedicated tenancy.	

You'll find a burgeoning list of options available when configuring an instance in Amazon EC2.

least nine general classes of instances available — and this is only when you consider the machines listed in the “current generation.” You can still rent instances from earlier hardware families, if your software seems to need it for some reason. Each of the general classes of machine is available in various models configured with various amounts of RAM and local disk storage.

There may be too many options in Amazon's list of instance types for mere mortals to examine and debate. The i2.8xlarge, for instance, comes with 244GB of RAM and 32 virtual CPUs that pump out 104 elastic compute units (ECUs), the metric that Amazon uses to measure the power of its machines. The d2.8xlarge comes with 244GB of RAM and 36 virtual CPUs that pump out 116 ECUs.

These instances use different versions of Intel's Xeon, and they're only two among dozens of options. I found myself scratching my head and wondering how to choose. If you have a serious project, you'll want to benchmark your code on a range of instance types and figure out how fast your application happens to run. If you're renting only a few machines for occasional work, it may not make sense to think too much about it. But if you're spinning

up hundreds of machines for larger projects, benchmarking is the best solution. The ECU metric, after all, is merely an average using some standard benchmarks. As they say in the car business, your mileage may vary.

Decisions, decisions

Choosing the hardware is only the beginning. Amazon has gradually increased the number of options as it learns more and more about what can go wrong, and the API reflects this experience. There's now a click box to “enable termination protection,” a kind of safety switch to prevent you from deleting your instance by mistake when you're pruning your list of running machines. Once you enable it, you have to explicitly disable it. I know, I know, you'll probably write a script to automatically disable it, but if you do you'll have only yourself to blame. Amazon tried. This is one of the little options that make life a bit easier — but not simpler — in the Amazon cloud.

The trickier questions come when you decide how to pay. You can rent outright and pay full price or start fishing for lower rates. Amazon offers a spot market that lets you bid for extra capacity, and it can fluctuate as demand ebbs and flows. These flows can be

Deep Dive

quite dramatic because some of the big video streaming services often take over the clouds on Friday and Saturday nights.

The savings can be surprisingly large, but you have no way of knowing what the final price will be. At the moment I'm writing this, an m3.medium instance is going for 0.91 cents per hour, much lower than the list price of 6.7 cents per hour. Everyone puts in their bids to run their jobs and if the winning bid price rises above your maximum bid, your machine shuts down.

Amazon gives software developers tools to automate the build, test, and deployment cycle. It also competes with GitHub by offering storage for the code on the way to deploying it.

You can pay a bit more for some guarantees. If you want your machine to run for at least an hour, the price (at the time I'm writing this) jumps to 3.7 cents, still far lower than the regular price.

The pricing model is not aimed at a continual auction alone. If you want stability instead of flexibility, there are even more options in the pricing charts. If you're willing to commit to one year or even three years, you can save 30 percent, 40 percent, or as much as 60-plus percent by paying up front and locking in a price — no auctions, only lower prices.

All this means your accountants will have as big a part to play as your programmers. You'll

want to sit both of them down to design the architecture because the pricing model is meant to encourage efficient use of resources. The programmers need to ask themselves again and again whether they can push computation into a batch job that can be executed occasionally when the cost of computing drops on the spot market.

Inventing the cloud

It's easy to find examples of cloud services that originated with Amazon. MapReduce, data warehousing, stream processing — AWS had them first. Only now are competitors beginning to emulate [Lambda](#), [Amazon's server-less, event-driven compute service](#). No cloud competitor has a match for Aurora, Amazon's souped-up MySQL service, or the range of databases Amazon offers as a service.

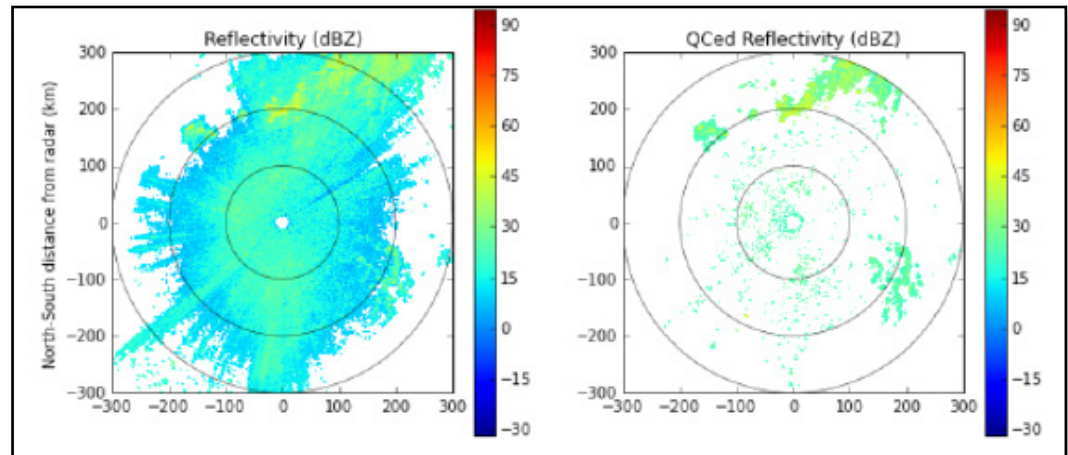
Amazon continues to innovate, as well as emulate its competitors. The Elastic Beanstalk is Amazon's version of the Google App Engine, a collection of software packages that automates the process of building a cluster of machines that grows and shrinks with demand. It's a more general system that supports a number of common server platforms for applications written in Java, .Net, Python, PHP, Node, Ruby, and Go. The scripts automatically configure the load balancer and machines, starting and stopping the basic EC2 instances as needed.

The Beanstalk is more like a concierge than a separate, all-encompassing framework like the Google App Engine. It runs your code on generic EC2 instances that will appear in your list of machines — important to keep in mind if you notice new instances starting up. I once spent several hours killing zombie instances that kept coming back to life before I remembered the experimental Elastic Beanstalk app I created. It kept discovering that something had killed the missing instance and dutifully brought it back to existence.

Elastic Beanstalk is nominally free. You don't pay anything extra for its services, but you pay for the EC2 instances it spins up on your behalf.

Amazon's EC2 Container Service takes the same approach with Docker containers. Amazon has its own little agent that will start up EC2 instances on your container's behalf, then install

Deep Dive



Amazon hosts dozens of public data sets, including climate data, geographic data, census data, Wikipedia traffic statistics, and even Enron emails. The Nexrad radar images shown above come from Amazon's public data set captured from the U.S. government's weather radar systems.

the Docker container. You don't pay for the container service, only the underlying resources.

Higher and higher

Not all of the tools are given away for free to rent out more machines. Cluster instances for Elastic MapReduce, Amazon's package of Hadoop-based tools, cost about 25 percent more over the underlying EC2 list prices. Amazon has created a fairly standard bundle of the major tools (Spark, Hadoop, Presto, Pig, Hive) and integrated them with Amazon's S3 storage system. If you're crunching logs or data in other parts of the Amazon cloud, analyzing them with an Elastic MapReduce cluster can make a great deal of sense. Getting the raw data out of AWS takes time and bandwidth.

Amazon is also moving higher in the stack by providing answers instead of computing resources. While Microsoft's Azure delivers a [more complete set machine learning tools](#) to data scientists, Amazon makes [machine learning more straightforward](#) for developers or business analysts. Ultimately, Amazon will sell you the analytics too.

One of the most intriguing new additions to the cloud is the public data that Amazon is gathering for us. For instance, you can chew on radar

images from [a network of 160 high-resolution Doppler radar sites](#) that provide data on precipitation and atmospheric movement in five-minute intervals. Amazon is putting up a number of big public data sets with the hope they'll attract projects that rent EC2 instances to ask questions. If your business involves the weather — hint, hint commodity traders — it's much easier to play.

Services like these show how Amazon plans to build out the cloud in the future. The company has mastered the process of delivering a commodity product, and it continues to improve the service to address the needs of a broad base of users. There are now so many options that it can take hours of research before you're ready to push the button and rent some machines — or nonmachines.

The newer services like Lambda and Amazon Machine Learning are clean layers meant to abstract away the complexity of renting and configuring EC2 instances. After all, nobody really wants to run servers. We want to deploy code, collect data, and find answers. We can expect Amazon to continue pushing the automation up the stack to deliver higher-level services that sweep the actual servers under the rug. In the meantime, Amazon continues to offer the largest array of instance types, along with the richest set of services and options to make the most of them. ■



Deep Dive

REVIEW

Microsoft Azure reaches beyond Windows

Microsoft's cloud built for Windows and .Net has exploded with open source options and big data services

BY PETER WAYNER

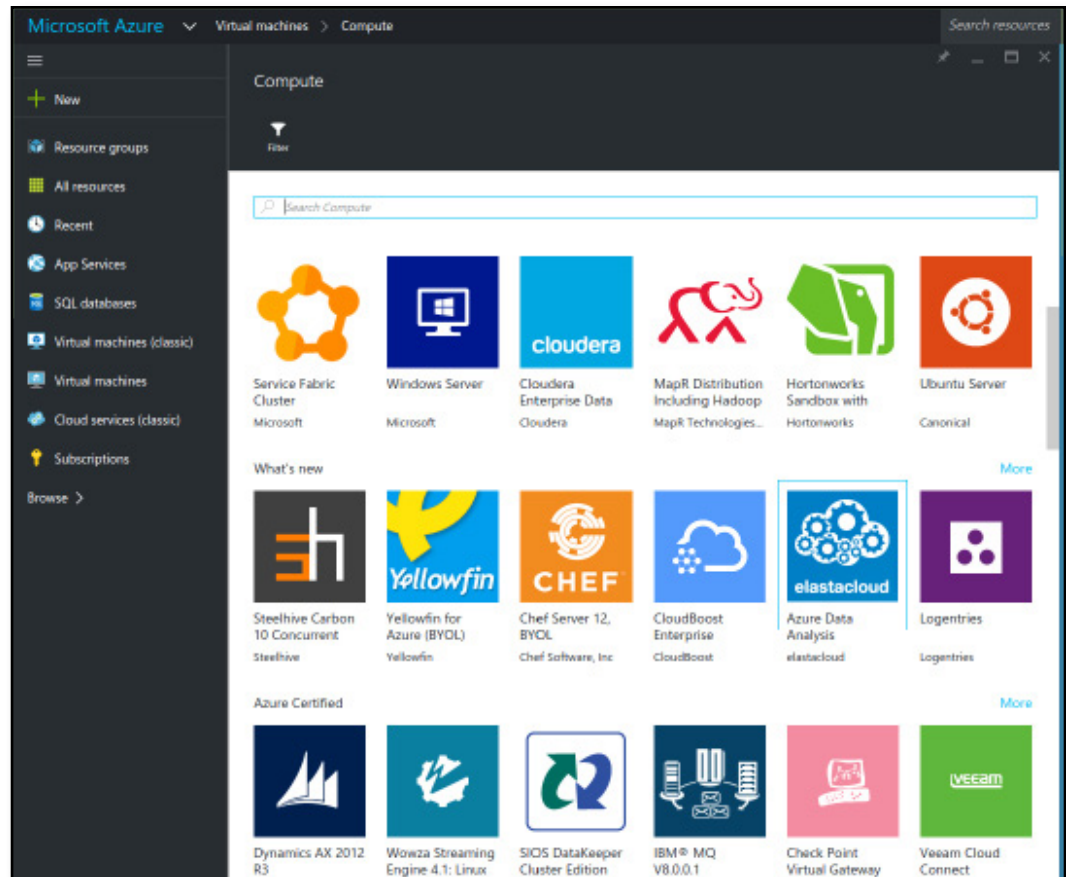
A long time ago in a century getting further and further away, Bill Gates compared MSN with the exploding World Wide Web, saw the future, and pivoted nicely to embrace the Internet. A few decades later, someone at Microsoft looked at the cloud and recognized that the old days of selling Windows Server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

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 that enterprise managers need to assemble modern, flexible websites.

There are common offerings, such as virtual machines, databases, and storage blocks. Then there are some not-so-common additions such as machine learning tools, parallel processing engines, service buses, networks, and connections to data farms. There are also some tools for debugging your code, sending emails, and installing databases like MongoDB or Cassandra or ClearDB's version of MySQL. You can draw on many tools made by companies that aren't Microsoft and many that are open source.

The wealth of options shows that Microsoft is actively trying to build a system that makes it easy for developers to produce a working

Deep Dive



Azure gives you hundreds of options including licensed software when you choose an image for your virtual machine. The price of the software is rolled into the price of the server.

website using the tools of their choice. Azure is not just delivering commodity hardware running Microsoft and leaving the rest up to you. It's starting to make it easier to bolt together all of the parts. In many cases, Microsoft bolted together many of the parts — or at least threaded the nuts — and your only job is to pour in some of your data. The process is rarely as simple as, say, sending an email, but it's dramatically easier than the old paradigm.

Microsoft's melting pot

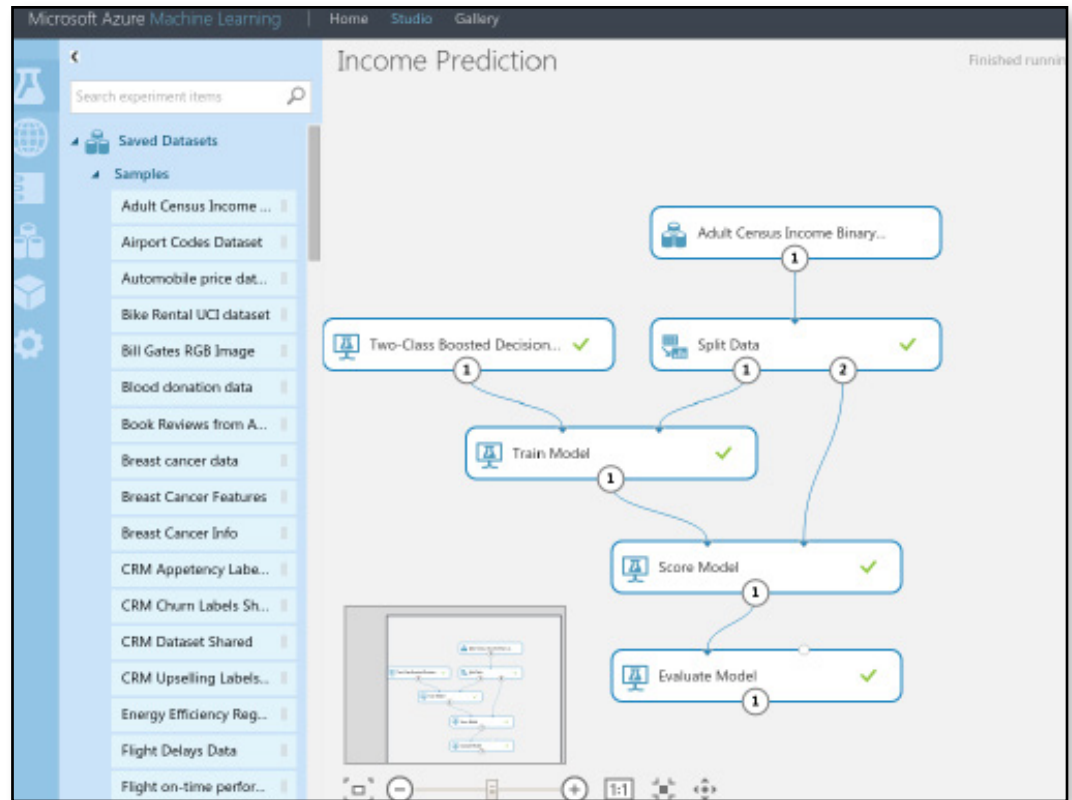
The Azure service is a godsend for those who are heavily invested in Microsoft's operating systems. Many of the other clouds offer only Linux or BSD machines, or charge more to build out a Microsoft Windows server. Azure rents some basic Windows machines at the same bargain rate as Linux. The fatter machines with more power,

though, will cost more if you want Windows running on them instead of Linux.

Did I say Linux? Yes, because Microsoft's embrace of open source is on full display with Azure. You can boot up a virtual machine and install popular Linux distros like Ubuntu Server 14.04 or CentOS 7.1. There is a surprisingly large number of open source distros available — enough so that Azure feels more like an open marketplace than a Windows shop. You can spin up machines from basic distributions built by companies like Canonical or OpenLogic, or draw on hundreds of images that are more finished machines than foundations for your own creation. There are so many options, it's almost impossible to actually make a decision.

There's an Apache Solr installation from Bitnami that's ready to index your documents, or a Chef server complete with a license for

Deep Dive



Analyzing data in Azure Machine Learning can be done by just dragging and dropping some tiles to create a flowchart or pipeline for your data.

maintaining 100 nodes (with more available). Or if you do want to start with a foundation layer, there are some virtual boxes that start with IBM's WebSphere application server. You'll find hundreds, if not thousands, of machine images in the selection box when you're building up a virtual machine.

If you want to build an application on Windows Server and the .Net stack, those options are still there, along with nice integrations with Visual Studio. But so are nice services for Node.js, PHP, Python, Java, and cross-platform mobile app developers.

Beyond virtual machines

It's probably a mistake to spend too much time focusing on the raw machines. Most of the new innovations at Azure are being sold as services, and some of the fanciest are referred to as engagements, machine learning, or insights. Many are quite clever and might be said to

deliver answers instead of raw compute cycles.

The machine learning option, for instance, takes the data from your database and analyzes it. There's no code, just a drag-and-drop interface that links together various algorithms into a pipeline. The data comes in and then gets fed into various machine learning algorithms until you find the signal that you want to find.

Some people might imagine that the tool's interface is so simple that you don't need to be a programmer. That might be stretching it — a non-data-scientist will probably be lost, no matter how easy it can be to drag the tiles into place. Still, the simplicity shows that Microsoft doesn't imagine Azure to be a cloud filled with commodity machines.

Another option is the collection of data lake services that store and analyze your data. Microsoft is bundling together Hadoop, Spark, and Storm while storing the data in HBase or U-SQL. It offers a thin layer of visualization that makes it

Deep Dive

a bit easier to watch what's happening and track your job. All of this runs on basic Microsoft cloud machines accessing data stored in Microsoft's disk farms, but you don't need to worry about that; you can just focus on Hadoop and your higher level questions.

There are many more similar options. One service called Stream Analytics is meant to "perform real-time analytics for your Internet of things solutions." Another offers Event Hubs to track what all of your things are doing when they phone home. Yet another provides some identity management with a directory service. The Web interfaces for these are long on slick graphics and short on the traditional command lines. And I'm only scratching the surface.

Media platform services

One of the biggest new areas is devoted to delivering "media," which largely means video and perhaps some audio. Microsoft has built out a content delivery network to make it simpler to stream whatever you want to the customers, no matter what their screen size or location happens to be. The tools will try to match the content to the screen to save bandwidth using the best set of encoders.

If you want to add some encryption and rights management, Microsoft wants to help. Azure offers top-grade AES encryption and multiple options for managing keys and rights with systems like PlayReady or Widevine.

All of this leads Microsoft to say that Azure provides all of the features you need to build

your own cloud-based DVR for content, both produced and live. It's an ambitious goal that will make it easier for video producers big and small to deliver their own content without relying on legacies like Netflix or Hulu. It's not hard for anyone with a credit card to open up a video shop online and then pay for the bandwidth if and when customers show up.

The scope of Azure and the ambition of Microsoft are both so great that it may be time to look for a new metaphor to replace the "cloud." Microsoft wants to do more than just rent out commodity machines by the boatload. Sure, there are still plenty of powerful machines at good prices. But the interesting stuff are the services that do the real work. You can roll out sophisticated analysis and juggle multiple streams of data, all from a Web interface.

In a sense, Microsoft is doing what it did decades ago when it rolled together a word processor and a spreadsheet into a single package called Office. The company has always found its success when it produced an interface that was simple enough for the average user but sophisticated enough to offer more of the power of the computer.

Microsoft is doing the same now with big data analytics and the Internet of things. It is building a nice, manageable set of tools so the average customer can conjure up amazing mathematical and analytical results that were once achievable only by the most highly skilled with the largest budgets. It's a compelling solution that's bound to find many takers. ■

Deep Dive



REVIEW

Google Cloud flexes flexibility

Google's elegant Cloud Platform makes it easy to spin up instances or simply tap Google APIs only when you need them

BY PETER WAYNER

If one company among all companies is synonymous with cloud-centered computing, it would be Google. From the very beginning, Google built a business located somewhere in the murky depths of the Internet, and its search engine continues to be one of the most formidable engineering marvels of the modern world. When was the last time there was an outage?

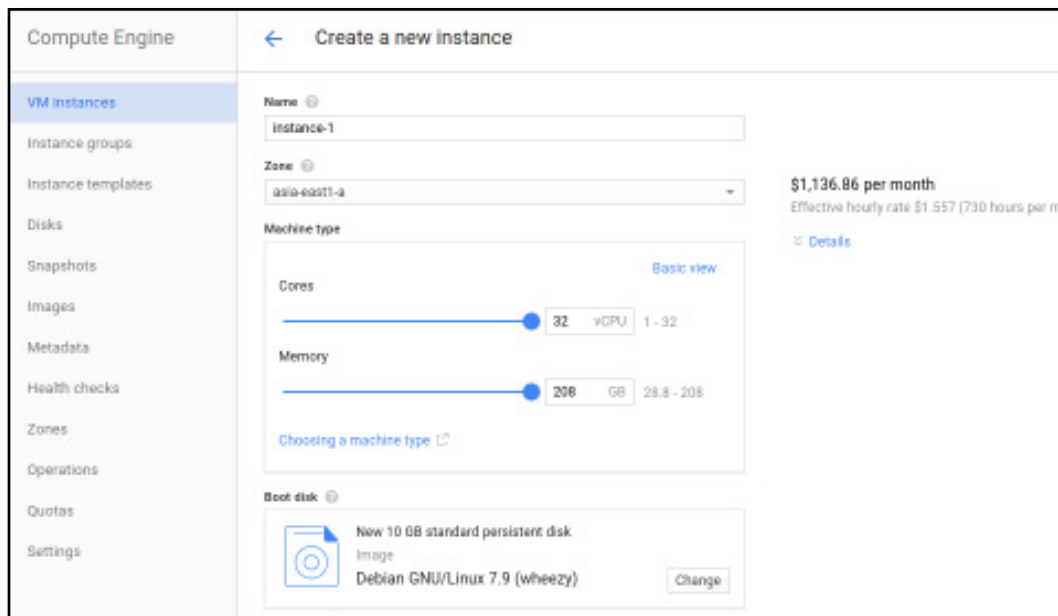
It's only natural that anyone looking to build an information-based business that spans the Internet would turn to Google and leverage all of its experience. As pioneers, if Google needed a technology, Google engineers had to develop it themselves, then deploy it. Now everyone can

profit from Google's skills and build a Google-grade system with Google-grade reliability for pennies per hour or per click.

The simplest path to using the Google Cloud is to start up an instance from the Google Compute Engine. After a few clicks on a Web page or a few calls on an API, you get to be root in your own virtual machine running deep inside Google's racks.

The basic options include 18 of the standard Linux distros (Ubuntu, Debian, Red Hat, CentOS, and so on) or — and this may be a bit of a surprise — two versions of Windows Server (2008 R2 or 2012 R2). Running Windows will

Deep Dive



The Google Compute Engine console lets you spin up instances by choosing the number of CPUs and gigabytes of RAM.

add a bit more to the cost: an extra \$14.60 per month for a standard, single-vCPU machine. The price on the OS goes up as you add cores.

Machines and containers

Google offers a wide range of hardware options and plenty of predefined configurations with as many as 32 virtual cores and as much as 208GB of RAM. If you don't like the standard options, the UI includes a couple of sliders that let you choose more arbitrary amounts like 10GB or 34GB of RAM. However, the amount of RAM is connected to the number of cores, and you can't choose more extreme combinations like one CPU and 180GB of RAM ... not that you might want it.

Useful options for developers are the micro and small instances that have shared CPUs and barely any RAM (0.6GB or 1.7GB). The prices are so low, they're almost nonexistent. The micro is a mere 0.9 cent per hour, which works out to be \$6.57 for a full month.

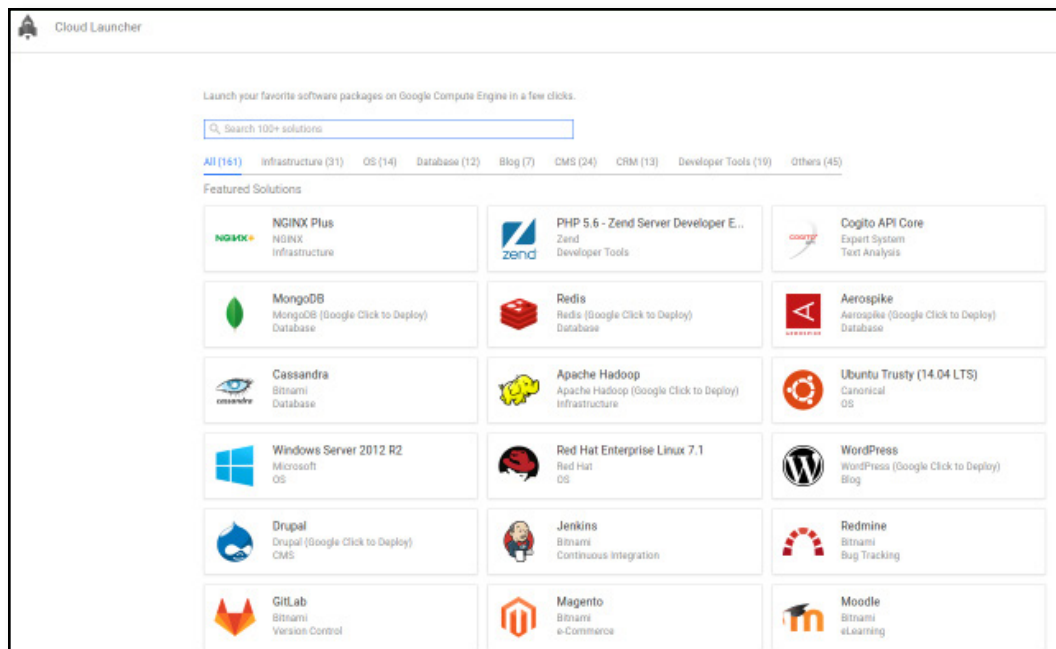
But the final price could be even lower. Google has started rewarding better customers by giving a "sustained use" discount to people who keep their machines running. The first week of the month is full price, but the next week

gets a 20 percent discount. If you leave your machine running for the full month, the last two weeks are discounted 60 and 40 percent respectively. This could average out to be a 30 percent discount off the rack rates for anyone who leaves their machine running.

Note that you don't need to leave the instance up continually to get the better price. Compute Engine bills by the minute and will adjust the rate accordingly. Google's VMs boot fairly quickly, so it's possible to start up instances for a half hour here or an hour there. Shorter times probably aren't that feasible, and in any case Compute Engine has a minimum 10-minute charge for booting up a VM.

There are other ways to save money. Batch processing and nonessential work can be done with pre-emptible instances that may or may not be there for you, working or not working. When they are running, you can pay up to 70 percent less, a substantial discount. Google's prices for pre-emptible instances are fixed, unlike Amazon's rates for spot instances, which are based on an auction market for excess computer cycles. If you want predictability, pre-emptible instances are for you.

Deep Dive



You'll find a good number of prebuilt machines available in the Google Cloud Engine, though the collection pales to those in Amazon and Azure.

It's worth noting that disk space is billed separately, in part because the disk behaves like it's independent of the machine itself. You "attach" the persistent disk to your instance, and when the instance dies, the disk lives on. You can attach it to another machine later. If you really don't need it, you can delete it. You can also juggle the disks by duplicating them with snapshots or even attaching them to multiple machines — although only in read-only mode.

A newer corner of Google's cloud is the Container Engine, an option that lets you spin up a cluster of VMs orchestrated by Google's own Kubernetes, a tool for juggling Docker containers across a collection of machines. The machines, in this case, come from the Compute Engine.

Using the system requires understanding the extra features that Kubernetes brings to the Docker world. You can bundle containers together in pods, which can be useful if the containers are going to share the same resources like a local disk.

The price for all of this depends upon how many Compute Engine instances are running underneath. If you have fewer than five nodes,

you pay only for the instances. If you have six nodes or more, it's an additional 15 cents an hour for the convenience.

Platforms and APIs

The raw instances aren't the only option. The App Engine is a completely different approach that was radical when it first appeared. Instead of building your server from the OS up, using all of the power of root, the App Engine is like a computing servant. You hand over a relatively thin piece of code and App Engine does the rest. Google handles the load balancers, the servers, the OS, and the databases, then bills you by the compute cycles it takes to answer an HTTP request.

It doesn't take much to build out an app. The first version of App Engine accepted only Python code. Today you can also upload Java, PHP, or Google's very own Go. Google's templates and libraries do much of the work. The standard approach is to take an open source framework like Django (Python) or WordPress (PHP), then add a few extensions.

The tricky part is to rethink your database

Deep Dive

access. Many of the basic open source frameworks evolved assuming that they could write to a local disk. App Engine wants you to write to Google's Cloud SQL or Cloud Storage or the NoSQL Datastore. This makes it simpler for App Engine to scale out your app to run on many machines at once because Google doesn't want to worry about juggling a file system.

App Engine is probably best for apps that end up being relatively thin layers of code between the user and the database. (Snapchat is one of the most famous App Engine success stories.) The big fear is that some app will go off on a tangent — say, chewing up cycles to compute pi to the millionth decimal. By forcing you to use the cloud storage services, Google retains more control. Google places tight “safety limits” on your code, although these are negotiable — especially if you're a big paying customer.

The preferred way to store your data with the Cloud Engine is in BigQuery, a set of append-only tables with an SQL-like interface. BigQuery is built for logs and logs of online data, the kind that people associate with the term “big data.” Google is starting to enhance this with more sophisticated tools, like Datalab, which offers a layer of graphics and analysis on top of the BigQuery data storage layer. You store your information in BigQuery, then boot up an App Engine instance running the Datalab code. Datalab is designed to encourage a measure of collaboration between anyone who logs into it through notebooks.

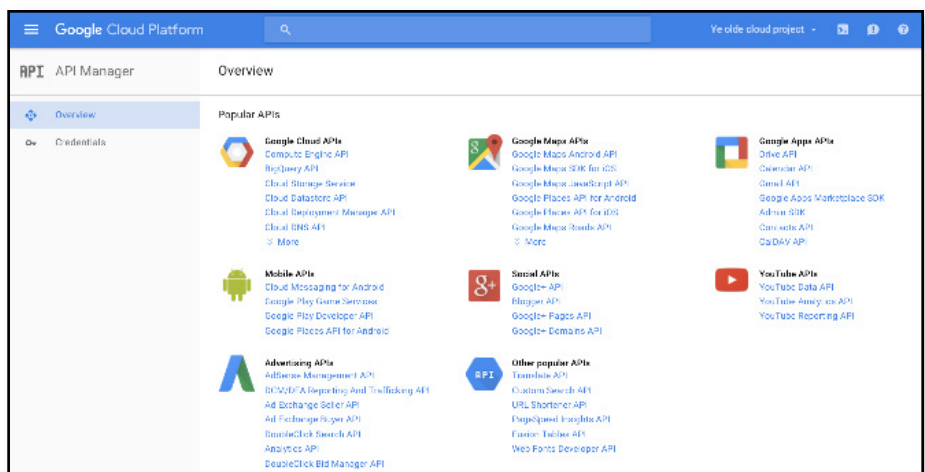
Finally, one of the more interesting options for the Google Cloud user is access to Google infrastructure and specialized services through the many APIs. More than 100 different APIs will let you offload some of your work to Google.

It's relatively simple, for instance, to let users of your application log in with their Google ID. The Translate API will convert text between any of several dozen languages. The Maps API will let you add location information to your website. There are also prediction APIs and even services aimed at genetic researchers in biology labs. It's a long list.

It's probably best to think of these APIs as completely independent of the Compute Engine. You don't need to run your code locally in a Google data center to use the APIs, although your response time might be a bit faster. But this may only be the case when your server-side code needs to access the API. If your client-side code is pulling down a map, there's no reason that hosting the server code at Google would result in better performance.

The choice is yours. Google has generally engineered every service to run independently from every other. If you want a Compute Engine machine but your own hardware for something more sensitive, Google is happy to help by renting you many of its powerful services through its APIs. All of the knowledge and expertise that makes the Google cloud run so smoothly is right at your fingertips. ■

Much of the magic of the Google Cloud Platform, along with many other intriguing services, is available through open APIs.



Deep Dive

**CLOUD REVIEW****IBM Cloud is built to order**

Big Blue lets you have public cloud your way with bare metal, private instances, and even custom-configured hardware options

BY PETER WAYNER

There was a time long ago when IBM ruled the cloud, although no one used that word back then. The company's mainframe line in the 1960s and '70s was the original distant and unseen pile of metal and silicon meant to be shared among all users. Each program got a slice of the big machine's time, and everyone understood they were "time sharing," which sounds a bit more precise than the amorphous word "cloud."

That was then. Today, no one should be surprised that IBM is playing in the modern cloud business because it practically invented the idea decades ago. The current clouds generally use the same slicing and dicing as IBM's original time-sharing 360 architecture, although the modern cloud sales lingo hides this fact behind the metaphor that these are individual "instances" that act like individual machines.

The process is different too. In the 1960s, you bumped into an IBM salesman at the country club who took you for a few rounds

of golf and more than a few drinks. Today you wiggle some sliders on your Web page and the price pops up, like on AWS, Google, and Azure — no drinks, no golf, no country club. But the price is much lower and you're not making a commitment that lasts more than a few minutes. When all is clicked and paid for, these services are pretty similar to the ones offered by the other major clouds. You get an instance that runs some code.

Real servers for rent

But IBM is also offering something a bit different from the Amazon, [Google](#), and [Microsoft](#) clouds. It's willing to rent you "bare-metal servers," which pretty much means a real physical box, the kind we used to buy and install in the rack of that machine room down the hall. In this case IBM installs the box in its own racks in its own server farms, although you can pay a bit more for a private rack if you want.

The sales literature is even a bit humorous

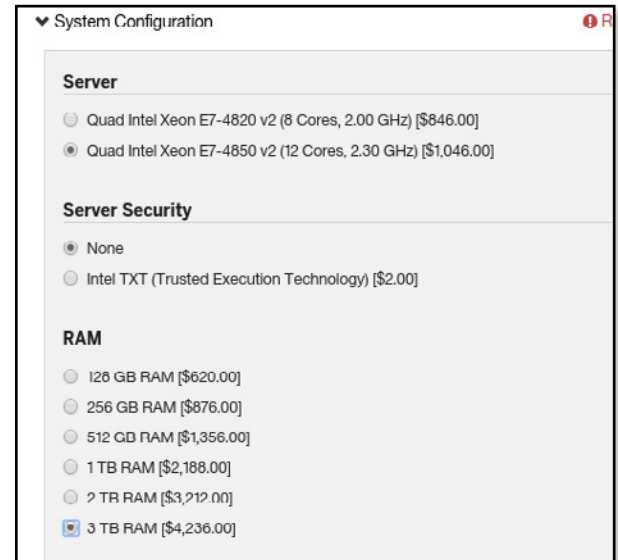
Deep Dive

and nostalgic to anyone who has been lost in the modern cloud. One section warns how it can take several hours for someone to locate your server and actually install RAM, GPU cards, or other real hardware in it. It's not as retro as golf and country clubs, but a real person has to get off their rear end, shake a leg, and wander through racks and racks so that you can have more RAM. Everyone else has tried to leave the physical world behind us by building a software layer that creates virtual machines, but IBM has built a real machine room that builds real machines to order.

No doubt there are advantages to IBM's reality-based approach. The biggest may be a wider range of options because you really can build machines to your exacting specs. IBM will pack up to 3TB of RAM into its machines (\$4,236 per month) if you want. This is at least 10 times bigger than the options I've seen in virtual machines on other clouds. Redundant power supplies? They start at \$50 per month. RAID arrays? The controller starts at \$50 per month. GPU? For \$759 per month, you get an Nvidia card.

This old-school approach is a big departure from the completely virtual, cloudy model of the other major providers. Amazon will rent you a GPU too, but it will be in its own corner of the cloud. You'll probably shuttle the data back and forth between your regular instances using the S3 object store. It sounds superflexible, but the communication time could be brutal. All of the flexibility of virtualization comes with overhead. It's bound to turn some functions into a slog.

After raw speed and flexibility, renting your very own piece of metal must appeal to those who worry about security. No matter how hard the hypervisor people work at sealing the holes between the virtual machines, we can never really be sure this additional layer is perfect. Renting bare-metal servers cuts out another possible source of trouble. It may be a bit of overkill, but there's a reason why some people buy their own limousine instead of calling Uber



Who wouldn't want 3TB of RAM? There's no shortage of hardware options on IBM Cloud.

Black. They don't know who's been using it the rest of the time.

There are costs, though, to this approach. If you want a standard configuration, IBM promises to get your machine up and running in 20 to 30 minutes. It could take several hours if you want something a bit out of the ordinary because real humans have to exert real effort.

That's OK if you're going to leave the machine running for a bit, but it won't work very well for those who want to spin up servers to handle some surge in demand, only to trash them a few hours or even minutes later. While other providers like Joyent are talking about spinning up containers in seconds, IBM's bare-metal business is made for the long haul.

Drawing comparisons

The extra complexity is reflected in the price. A single processor with four cores (Xeon ES-1270 at 3.4GHz) and 8GB of RAM starts at \$235 per month. Four cores with 8GB of RAM from Google's Compute Engine rent for \$99.96 a month, after sustained use discounts. If you don't want to rent by the month, IBM will rent you bare-metal machines by the hour. These hourly rates are steeper than those for virtual cloud machines by about the same multiples.

Of course it's not exactly fair to look at these

Deep Dive

Data Center

Attention! Data Center selection is required.

- AMS01 - Amsterdam
- AMS03 - Amsterdam
- CHE01 - Chennai 1
- DAL01 - Dallas
- DAL05 - Dallas
- DAL06 - Dallas
- DAL09 - Dallas
- FRA02 - Frankfurt
- HKG02 - Hong Kong
- HOU02 - Houston
- LON02 - London
- MEL01 - Melbourne
- MEX01 - Queretaro
- MIL01 - Milan
- MON01 - Montreal
- PAR01 - Paris
- SJC01 - San Jose
- SJC03 - San Jose 3
- SAO01 - Sao Paulo 1
- SEA01 - Seattle
- SNG01 - Singapore
- SYD01 - Sydney
- TOK02 - Tokyo
- TOR01 - Toronto
- WDC01 - Washington, DC

Data Center

Computing Instance
1 x 2.0 GHz Core \$0.025 /hr.

RAM
2 GB \$0.030 /hr.

Operating System
\$0.00 /hr.

First Disk
25 GB (SAN) \$0.000 /hr.

Public Bandwidth
0 GB Bandwidth \$0.000 /hr.

Uplink Port Speeds
100 Mbps Public & Private Network Uplinks

Monitoring
Host Ping \$0.000 /hr.

Response
Automated Notification \$0.000 /hr.

VPN Management - Private Network
Unlimited SSL VPN Users & 1 PPTP VPN User per account

Vulnerability Assessments & Management
Nessus Vulnerability Assessment & Reporting

Primary IP Addresses
1 IP Address

Think global, deploy local. IBM Cloud provides a long list of available data center locations when you configure a server.

prices and jump to any conclusions. The IBM machine comes with 2TB of local disk, while Google wants you to store your data in its cloud and only use the local disks for copies of the OS. And it's hard to compare the computing power of the four cores in both situations without doing real benchmarks. But a bare-metal machine certainly eliminates the overhead of a virtual one. This is the price you pay for the peace of mind of owning — er, renting your very own box.

The options are hardly limited to bare-metal boxes. IBM is also renting virtual servers at prices closer to those of the other clouds, though they're not exactly a steal. A basic machine with four cores and 8GB of RAM will run you 20.5 cents per hour. Compare that to 20 cents per hour for Google's n1-standard-4, a machine that combines four cores with 15GB of RAM. And Google offers discounts if you leave your machine up and running for a big percentage of the month.

As on the other clouds, you can install some of the major distros as images, and some of the images can be fattened with the basic LAMP binaries. Microsoft Windows Server is available for 2.4 cents more per hour, or 18 cents more an hour if you want the Datacenter edition. You can also pay more (8 cents an hour) for Red Hat Linux.

One interesting option on the menu lets you add up to five different disks. The first and second can be "local" to the instance or stored on a SAN, but the last three need to be on a SAN. This approach will be more comfortable to anyone who is still used to storing data in the OS's file system instead of dumping it in some cloud storage layer with an API. It will also be relevant to anyone with legacy software written to use some "local" data. You can run your old software in this environment more easily. If anyone knows dusty deck computing, it's IBM.

IBM also knows upselling. Do you worry

Deep Dive

about viruses? McAfee's Virus Scan for Enterprises is available for a click. IBM also sells different monitoring packages and services, with a long list of agents available for Windows and Linux.

Yet another difference is the private nodes. If you want a bit more isolation, IBM will boot up your virtual machines in a private hardware instance. You get the security of your own bare-metal machine with the flexibility of using virtual machines. You can pack several VMs into your private node, which is bound to keep the developers happy.

A global reach

It's worth noting that IBM's cloud is truly global. I counted 19 different cities from five continents in the list of options for where to spin up an instance. By comparison, I see only nine options in the Amazon menu. Google doesn't even specify cities, offering only four basic locations like US East and Europe West.

IBM's greater experience with global markets will be important for some customers. For all of the talk about global commerce and market unification, some areas of the law are very local. It can matter where the server is

physically located, and more options help.

There's much more to IBM than virtual instances and real servers in real racks. That's only one corner of a big company, a corner with the SoftLayer brand that IBM acquired. The company will be happy to help you build your big data stack or e-commerce solution in its cloud, or you could turn to the IBM Bluemix PaaS to build them in the cloud yourself.

While many of the other cloud companies are starting to sell business intelligence and analytical software in their cloud platforms, IBM has been leading the way with research in this area for years. This is the company, after all, that won at "Jeopardy," and now it's selling all of this experience in the cloud, where you can tap IBM Watson Analytics as a service.

Of course, IBM has separate divisions that are ready to sell the software. Separating the various software pieces into separate divisions makes some sense. The cloud division can concentrate on delivering classic hardware in a stunning variety. IBM may have unbundled the golf and the fancy cocktails from the sales cycle, but the hardware is ready to run faster and more flexibly than ever. ■

Deep Dive

REVIEW

Joyent Cloud runs closer to the metal

Through clever engineering, Joyent gives you faster startups and more speed than VM-based clouds

BY PETER WAYNER

The world of cloud computing is dominated by Tyrannosaurus Rex-sized corporations like [Google](#), [Amazon](#), and [Microsoft](#), big companies that have used their prowess and might to build giant data centers full of racks and racks of machines. Then there's Joyent, a small but nimble competitor that might be called mammal-sized, if only for the sake of a metaphor. It deserves notice for shaking up the cloud marketplace with clever mods of the operating system layer that are producing great results.

The metaphor isn't perfect because the technology here isn't entirely new. Joyent might be called one of the last vestiges of Sun Microsystems. While Oracle ended up with control over Java and the Sun hardware, Joyent embraced OpenSolaris and welcomed some of the experts who knew what made the OS special. The company leveraged this talent to update Solaris and build a cloud that delivers great performance for Java, Node.js, and Docker containers.

Joyent's version of Solaris is called SmartOS. It forms the foundation for Triton, a container hosting service, and for many of Joyent's raw

compute instances. Joyent will also load up images with Ubuntu, CentOS, Debian, and a few other operating systems, but SmartOS is the main attraction.

There aren't as many options as you'll find in the other clouds, but there are plenty of images that are preconfigured to run many standard services — Nginx, MongoDB, Hadoop, and so on — on SmartOS. You can start up a cluster fairly quickly without spending too much time remembering what's different about Solaris.

Docker meets Solaris Zones

Joyent's real focus these days is on Docker containers, not full instances, in part because Joyent engineers have managed to hack the SmartOS kernel and strip away a number of the layers that usually exist in the cloud. Instead of putting the container inside an OS running in a VM, Triton offers "native containers." They've arranged for the SmartOS kernel to emulate the Linux kernel running Docker, a simpler process with fewer levels of indirection than hardware virtualization. (As the Fundamental Law of

Deep Dive

Create Instance

Data Center:

Quick | Saved | Docker Container | Infrastructure Container | Hardware Virtual Machine

Choose Image | Select Package | Review

OS: Search image

alpine-3 Container-native Alpine 3.2 64-bit image. Built to run on containers with bare metal speed, while offering all the services of a typical unix host. <input type="button" value="Select"/>	apache A SmartOS 64-bit image pre-configured and optimized as an Apache 2.4.12 Server with Event MPM and PHP-FPM. Comes packaged with a -- <input type="button" value="Select"/>	base A 32-bit SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>	base-32 A 32-bit SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>	base-32-lts A 32-bit SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>
base-64 A 64-bit SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>	base-64-lts A 64-bit SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>	base-multiarch A multibarch SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>	base-multiarch-lts A multibarch SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>	base64 A 64-bit SmartOS image with just essential packages installed. Ideal for users who are comfortable with setting up their own environment and tools. <input type="button" value="Select"/>
cassandra A SmartOS 64-bit image that comes with Apache Cassandra (2.1.2) and OpenJDK7 (1.7.76) pre-installed and configured. <input type="button" value="Select"/>	centos-6 Container-native CentOS 6.7 64-bit image. Built to run on containers with bare metal speed, while offering all the services of a typical unix host <input type="button" value="Select"/>	centos-7 Container-native CentOS 7.1 64-bit image. Built to run on containers with bare metal speed, while offering all the services of a typical unix host. <input type="button" value="Select"/>	couchdb A SmartOS 64-bit image pre-configured and optimized as a CouchDB 1.6.1 server. Comes packaged listening on localhost only, SSL support, and a -- <input type="button" value="Select"/>	debian-7 Container-native Debian 7.8 (wheezy) 64-bit image. Built to run on containers with bare metal speed, while offering all the services of a typical -- <input type="button" value="Select"/>

Joyent offers many of the popular options for creating an instance from a standard image.

Computer Science states, programmers solve problems by inserting another layer that translates IP addresses, API calls, or whatever. Triton strips away some of these layers.)

It's always dangerous to speak about speed with too much certainty. One app can perform amazingly quickly while another will sink into a swamp. Still our tests over the years have shown both [Java performance](#) and [Docker performance](#) to be very good on Joyent Cloud. When I was rerunning some of the tests for this article, I found that the performance was substantially better. Your mileage will vary, of course, but there's reason to believe that Joyent is doing something right.

The most interesting aspect to Joyent's container service may be the speed it offers at startup. The other clouds will offer to host your Docker container, but this usually means they'll sell you an instance that's preconfigured with some of the basics of Docker. It's not a bad option, but you have to wait for the instance to start (and eventually stop). This can easily take several minutes — maybe close to

15, if it gets busy. Google, for instance, has a 10-minute minimum for starting up an instance to discourage wanton starting and stopping by users taking the overhead into account.

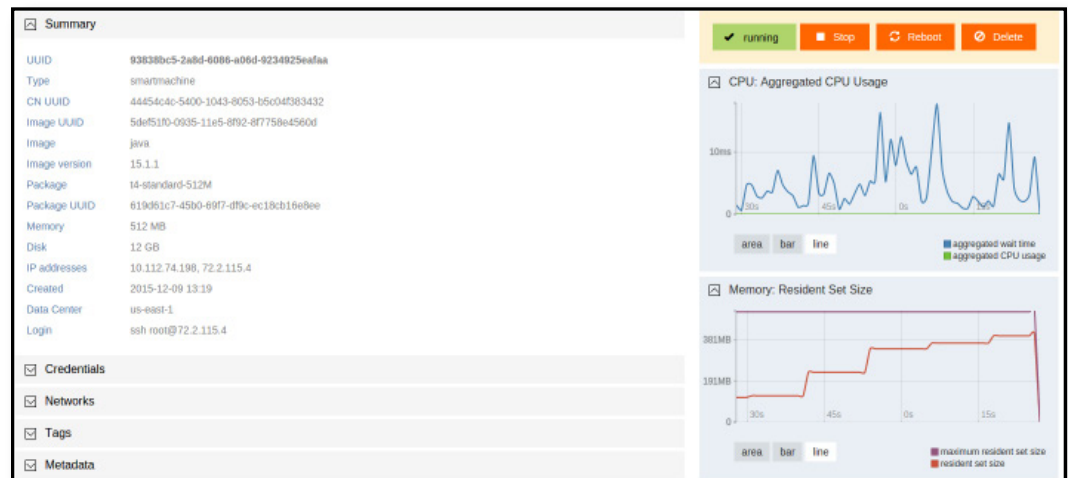
Joyent's cloud can start up Docker containers in seconds. The OS is already running underneath, and it only needs to do a bit of housekeeping to initiate the container. This means you can start to think in smaller slices of computing time. In my conversations, I found the Joyent engineers suggesting that they were getting close to being fast enough to allow their cloud to respond to an HTTP request by starting up the container.

The upshot is that Joyent lets you rethink your architecture to include containers that start up occasionally when they happen to be needed. If you're rebuilding your apps to include many microservice calls, the opportunities for isolating these microservices in their own containers are enticing.

Storage and compute: Better together

Joyent is also starting to dig into the world of big data by adding compute capabilities to its

Deep Dive



Joyent's performance page for an instance reports CPU and memory usage.

object storage layer. As with the other clouds, you can park your data in Joyent's object storage service, called Manta, with a simple REST API. Manta starts off making two copies, and you can add more replication to store more copies if you need them.

That's all pretty standard. The difference is that Joyent has installed many common software tools in the nodes storing the data. The common architectural strategy is to split off the storage from the computation in a completely different service that may or may not be hosted nearby. (It's not exactly clear why society has chosen that path — it only increases latency and adds traffic to the network.) Joyent is ignoring that memo and offering what it calls "integrated compute." You might think of it as "Hadoop but with Unix" because you specify the job as a JSON block full of Unix shell scripts.

Once you store your data, you can then start computing on it. Many of the standard tools like R, Python, and Node.js are already installed on the

SmartOS instance that's also storing your data. You simply need to trigger them. It's much faster than waiting for the data to be fetched from another machine, which may be why Joyent can offer to charge only for each second you do some computation with no "provisioning, data movement, or scheduling latency costs."

Joyent doesn't offer all of the extras you can find in the other clouds. There's not much talk about machine learning, managed service APIs, or public data sets. You can't turn to Joyent for help with displaying maps on your website or predicting travel times. The organism is smaller than some of the big reptiles that dominate the space. But you can run your code on what Joyent calls "bare metal."

Bare metal is not the best metaphor because there are still some software layers in the way, but substantially fewer than you'll find in the standard cloud configuration. That means faster performance and lower prices. It's a simpler model, but it worked for the mammals. ■

Peter Wayner is contributing editor at InfoWorld and the author of more than 16 books on diverse topics, including open source software ("Free for All"), autonomous cars ("Future Ride"), privacy-enhanced computation ("Translucent Databases"), digital transactions ("Digital Cash"), and steganography ("Disappearing Cryptography"). His work regularly appears in InfoWorld, and he consults on software projects big and small. The second edition of his book on robot cars, "[Future Ride](#)," is out now.