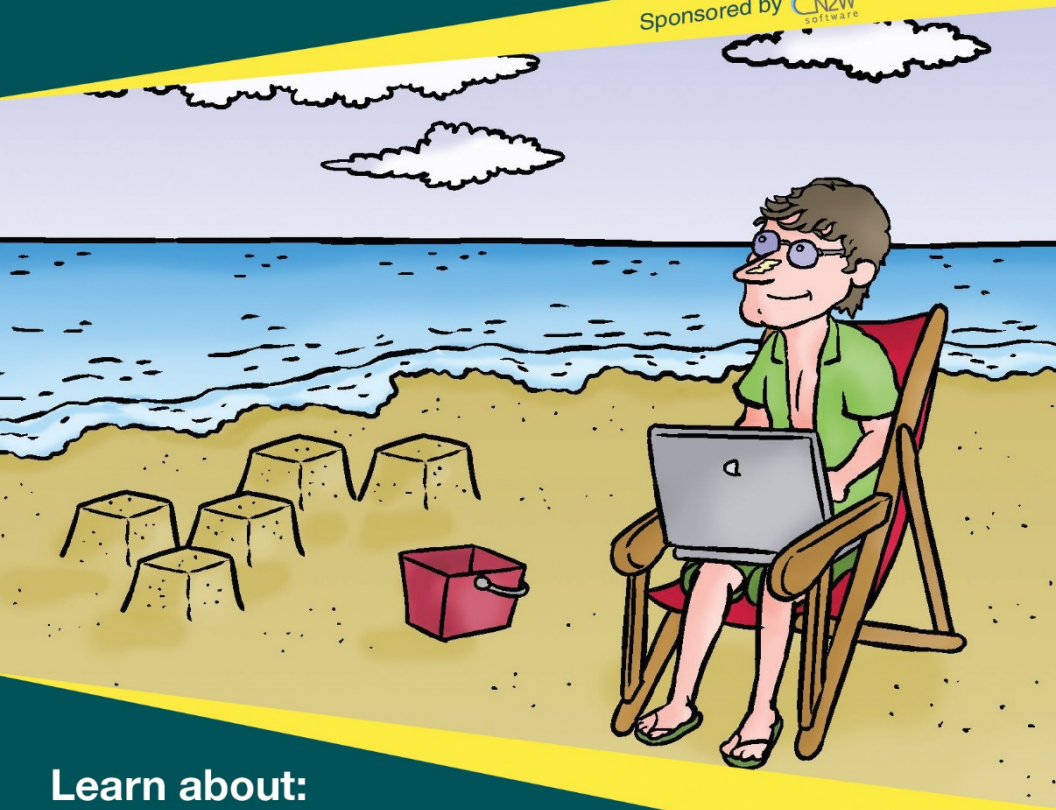
Conversational Amazon Web Services (AWS)

A ConversationalGeek® Book

Sponsored by N2W software



Learn about:

- The key components that make up AWS
- How to better predict the cost of operating in the Amazon cloud
- The purpose and value of the AWS Marketplace

2nd Edition

By Brien M. Posey (Microsoft MVP, Commercial Scientist Astronaut Candidate)

Sponsored by N2W Software

N2W Software was founded in 2012 with the mission of providing enterprise-class data protection for production environments deployed in the public cloud.

Cloud Protection Manager (CPM) from N2W Software was designed and built from the ground up to meet the backup and DR requirements of the most demanding customers migrating to the public cloud. CPM is now the leading enterprise-class backup, recovery, and disaster recovery solution specifically optimized for Amazon's AWS EC2 infrastructure. For more information visit:

www.n2ws.com

Conversational
Amazon Web Services (AWS)

By Brien Posey

ConversationalGeek®

Conversational Amazon Web Services (AWS)

Published by Conversational Geek® Inc.

www.conversationalgeek.com

Trademarks

Warning and Disclaimer

Additional Information

For general information on our other products and services, or how to create a custom Conversational Geek book for your business or organization, please visit our website at ConversationalGeek.com

Publisher Acknowledgments

All of the folks responsible for the creation of this guide:

Author(s):	Brien Posey
Project Editor:	J. Peter Bruzzese
Copy Editor:	John Rugh
Content Reviewer:	Karla Reina

Note from the Author

Welcome to Conversational Amazon Web Services (AWS)! I'm Brien Posey, a long time tech author, and I am going to be giving you a quick crash course on the subject of Amazon Web Services (AWS).

Over the last several years, there has been a relentless push from vendors and from the IT industry as a whole to move to the cloud. This push has been so pronounced that I have actually had family members who have absolutely no connection to the IT world call and ask me why it's so important for them to move to the cloud, and how they can go about doing it.

As you can imagine, it's always fun to try to explain that the term cloud is basically a meaningless buzz word. There are many different types of clouds and there are also many different cloud providers. Each of these providers has its own way of doing things. It's up to each individual provider to decide what services they want to offer, and how those services should be exposed to customers.

As you have probably guessed from the thickness of this book (or the lack thereof), this text isn't meant to be an exhaustive, super deep, hard core technical reference. Rather, it's me taking the time to explain some of the more important concepts in an effort to familiarize you with the subject. That way when you go shopping for an AWS solution you'll know which features and capabilities you should be looking for, and you'll understand why those features are important.

Brien M. Posey

The “Conversational” Method

We have two objectives when we create a “Conversational” book: First, to make sure it’s written in a conversational tone so it’s fun and easy to read. Second, to make sure you, the reader, can immediately take what you read and include it into your own conversations (personal or business-focused) with confidence.

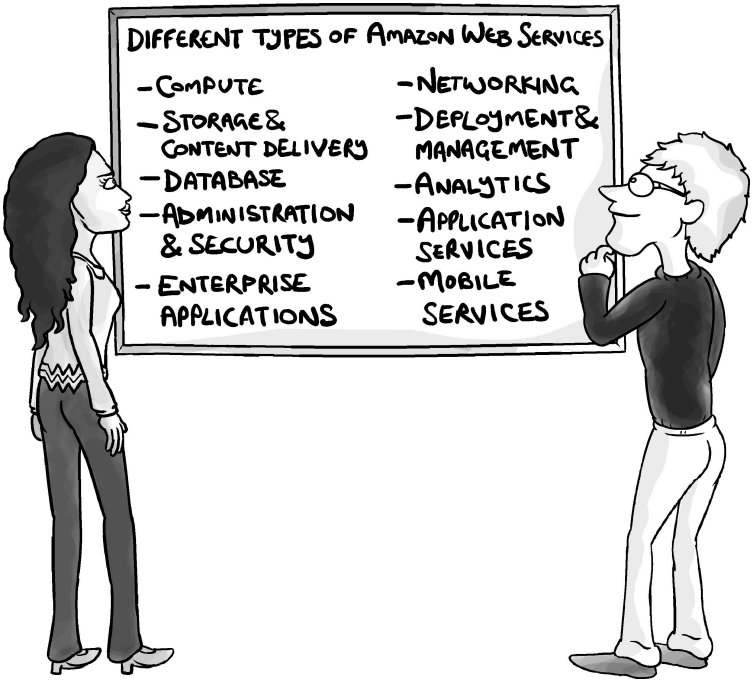
These books are meant to increase your understanding of the subject. Terminology, conceptual ideas, trends in the market, and even fringe subject matter are brought together to ensure you can engage your customer, team, co-worker, friend and even the know-it-all Best Buy geek on a level playing field.

“Geek in the Mirror” Boxes

We infuse humor into our books through both cartoons and light banter from the author. When you see one of these boxes, it’s the author stepping outside the dialog to speak directly to you. It might be an anecdote, it might be a personal experience or gut reaction and analysis, it might just be a sarcastic quip, but you don’t want to skip these “geek in the mirror” boxes.

Within these boxes, I can share just about anything on the subject at hand. Read ’em!

Amazon Web Services (AWS)



Amazon is one of the major players in the enterprise cloud space. Amazon's cloud is often referred to as Amazon Web Service or AWS. As previously mentioned however, it's up to each individual cloud provider to choose which services it wants to make available to its customers. One of the reasons why the Amazon cloud has become so popular is because, as the name implies, the Amazon Web Services are just that – a collection of services. Actually, that's putting it mildly. There are a lot of different services (think components) that fall under the AWS umbrella. In fact, it's relatively common for those who are new to the Amazon Web Services to feel somewhat overwhelmed because there are so many different services offered, and many of the services have cryptic sounding names that might not immediately convey the service's intended use. I'm talking about services with names

like EC2 or Glacier. As such, the key to getting to know the Amazon cloud is to get a handle on some of the Amazon lingo.

In my opening remarks, I mentioned that there are many different types of clouds. I've often said that reading about the public cloud is kind of like staring at a bowl of alphabet soup. There are all of these crazy acronyms that describe various types of clouds. For instance, we have:

- IaaS – Infrastructure as a Service
- PaaS – Platform as a Service
- DRaaS – Disaster recovery as a Service
- BaaS – Backup as a Service

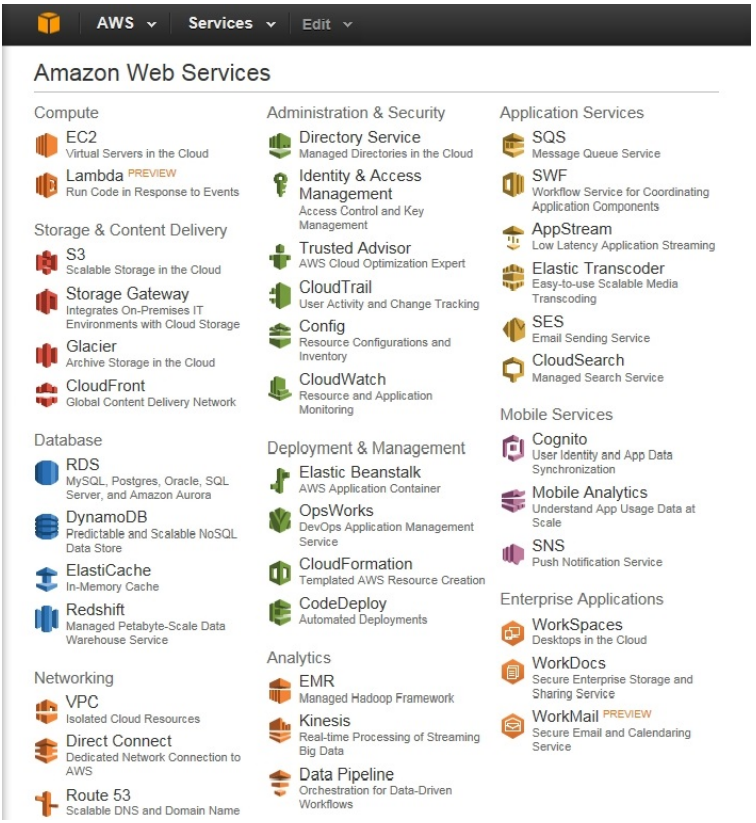
And the list goes on and on. My personal favorite acronym is SaaS because it can mean so many different things. SaaS originally referred to Software as a Service, but I have seen it used to refer to just about any IT-related word that starts with the letter S and can be loosely tied to the cloud.

For example, I once had an editor ask me to write about SaaS. Given the context of our conversation, I assumed the editor wanted an article on Storage as a Service. Soon after submitting the article however, I received an e-mail message from a bewildered editor who wanted to know why I hadn't written about Security as a Service. But I digress.

So why am I telling you all of this? Well, earlier I said that I was going to talk about some of the major components that make up the Amazon Web Services. Most of the cloud acronyms (something aaS) map directly to one or more of the various Amazon components. About the only public cloud type that isn't represented is Software as a Service (at least not in the

traditional sense). Typically, that is something that would be offered by an application vendor on a subscription basis.

So what are some of the individual services that make up AWS? Well, the list is constantly evolving, but note a screen capture of some of the services that are being offered today.



These are some of the services that make up the Amazon Web Services.

As you look at this figure, you will notice that there are nearly 40 different services that make up the Amazon Web Services. That's a lot of services! Thankfully, you don't have to memorize all of these service names and what each service does. Most AWS subscribers I've talked to only use a hand full of services.

The impression that I get when I look at the list of services that are included in the Amazon Web Services is that Amazon is trying hard to be all things to all people. The reason why I say this is because some of these services are general purpose, while others do really specialized things that probably won't benefit the average organization. This is especially true for smaller organizations. Think about it this way… Small startups often try to operate almost entirely in the cloud because doing so is far less expensive than constructing a datacenter. So with that in mind, does it seem likely that the average startup with 10 employees is going to need a managed Hadoop framework or scalable media transcoding? Some might, but I am guessing that most startups probably stick to the basics.

The point is that almost nobody uses all of the available services, so you don't have to worry if you don't completely understand what all of the services do, or if there are certain services that just don't seem like a good fit for your organization. It's OK to pick and choose. There's no penalty for not using a particular service, and you will save money by not using every conceivable service.

Of course, even if you don't plan to use all of the services, you are still going to have to work through the list of services in order to figure out what services Amazon offers, and which might be beneficial to your organization. Even though some of these services have unusual names, Amazon does provide a lot of help. If you look back at the previous figure for example, you will notice that the services are grouped into categories such as Compute, Networking, and Database. Sure, there are multiple options in each category, but as you can see in the previous screen capture, Amazon provides a short description of each service.

So which services do you really need to know about? The answer to that question really just depends on your organization's unique needs. Based on my own experience,

these are the services I personally have found to be the most useful:

- EC2 – EC2 is hands down the most useful service in the Amazon Web Services. EC2 is the service that lets you create and run virtual machines in the cloud. I will be talking a lot more about EC2 later on.

- S3 – Think of S3 as general purpose cloud storage. It's scalable, flexible, and easy to use.

- Glacier – S3 storage works well for most things, but it does come at a price. I plan to talk about costs later on, but generally speaking, storage costs are based on the storage capacity consumed and on the IOPS that you generate. If you have seldom accessed data, you may be able to reduce costs by placing that data into archive storage (Glacier). Glacier doesn't offer the performance or flexibility of S3, but it costs less.

- RDS – RDS is a database service. If you need to link to a database, RDS is a good, general purpose choice. The nice thing about RDS is that Amazon gives you options. If you like Microsoft software, you can use RDS to deploy a SQL Server database. If you prefer open source, you might be happier using MySQL. Not a SQL fan? No problem. RDS also supports Oracle databases and a variety of other database types.

- Direct Connect – Direct Connect is a service that is useful for tying your local network into the Amazon cloud.

- Directory Service – The Directory Service component allows you to create managed directories in the cloud.

Now obviously, you don't have to use the Directory Service service in order to establish a cloud-based directory. You always have the option of creating a virtualized domain controller instead. However, Directory Service lets you easily build a cloud-level directory in a manner similar to that of Microsoft's Azure AD.

How Much Does All of this Cost?

As previously noted, I'm going to spend quite a bit of time talking about running virtual machines in the cloud. And as I do, I am going to try to answer the most commonly asked questions. First, however, I want to get the big question out of the way – what does all of this cost?

The marketing hype coming from some of the cloud vendors makes it seem as though running services in a public cloud is ridiculously cheap. In reality however, the costs can be significant, depending on how you are using the cloud.

The main financial benefit cloud service providers such as Amazon deliver over running services locally is that public clouds allow an organization to run workloads without making a large upfront investment. Think of it as renting hardware and software resources rather than buying them. Suppose for instance that you need to deploy a Windows Server 2012 R2 virtual machine. You could purchase a host server, storage, a hypervisor, and a guest operating system, or you could opt instead to deploy the VM in the cloud. The first approach would require many thousands of dollars of up-front investment, and a significant amount of time to deploy. The second approach would require paying a relatively small subscription fee and the virtual machine could be up and running in a matter of minutes.

Avoiding the up-front investment in server hardware and software isn't the only financial benefit to running workloads in the cloud. Another potential financial benefit comes in the form of reduced maintenance costs. The cloud provider takes care of making sure that service level agreements are met, and that resources are kept up to date. Therefore, the IT staff is freed from at least some of the mundane maintenance and troubleshooting tasks that have long been associated with working in IT.

It is worth pointing out that although cloud services such as Amazon EC2 have been traditionally marketed as the cheap alternative to running workloads in your own datacenter, the long term cost of running a virtual machine in the cloud can dwarf the long term cost of running the same virtual machine in your own datacenter. The reason for this is simple. When you run virtual machines on premises, the hardware and software costs are one-time expenses (at least for the purpose of this example).

In contrast, running a virtual machine in the public cloud results in an ongoing cost. You will never actually own the hardware or the software licenses. Over time, the cumulative subscription fees will eventually exceed the price at which you could have outright purchased the resources for use in your own datacenter. That's how cloud service providers like Amazon can afford to stay in business.

It's also worth noting that the cost of running a workload in the cloud can sometimes be unpredictable. I'm going to discuss workload costs in more detail later on, but generally speaking, you have to pay for the hardware resources you consume. A high demand virtual machine, for instance, is usually going to cost more to operate than a comparably configured low demand virtual machine, because the high demand virtual machine is consuming more hardware resources.

The problem that some organizations have run into is that it's sometimes difficult to predict workload demand, which in turn can make it difficult to predict costs. A couple of years ago, for example, a friend decided to run a virtualized application server in the cloud (it wasn't the Amazon cloud).

It cost my friend about $400 to run this virtual machine for the first month. However, the bill for the second month was nearly $10,000. The reason for the huge difference in price was that during the first month, the application server was mostly idle, because it was being used primarily for testing. At some point during the second month, the virtual machine was put into production and the cost sky rocketed as demand for the VM increased.

So does that mean that you should avoid running workloads in the public cloud? Of course not. You just need to be smart about doing so. There are some situations in which running a workload in the cloud makes a lot of sense, and other situations in which it makes more sense to run the workload locally.

If you do opt to run a workload in the cloud, then you need to understand how the workload is being billed, and you need to know whether or not there are any safety mechanisms available to you that can keep you from falling victim to runaway costs.

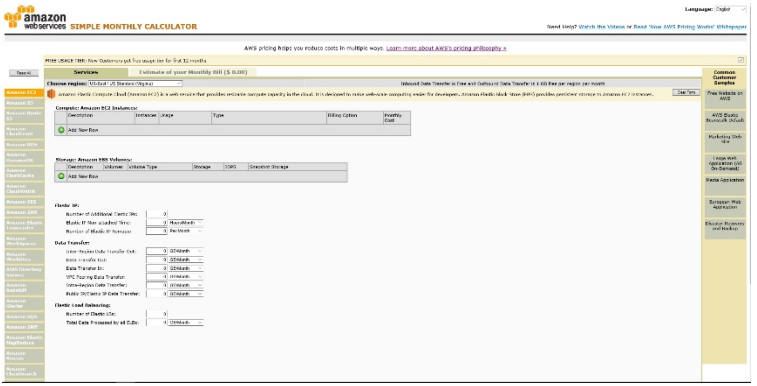
The Amazon Cost Calculator

As I said earlier, Amazon's pricing model is based on resource consumption. In the case of storage, for example, Amazon bases the costs on the number of gigabytes of storage in use, the number of PUT / COPY / POST / LIST requests, the number of GET requests, data transfers, inter-region data transfers, and the list goes on and on. As if that weren't complicated enough, Amazon breaks S3 storage costs down by Standard Storage, Standard – Infrequent Access Storage, and Reduced

Redundancy Storage. Needless to say, with so many different variables being used to tally the monthly cost, the pricing formulas tend to be really complicated. You can see some pricing information at: http://aws.amazon.com/ec2/pricing/

Thankfully, Amazon has created a pricing calculator you can use to estimate the cost of running your workloads in the cloud. You can access this calculator at: http://calculator.s3.amazonaws.com/index.htm l

You can see what the calculator looks like in the following figure. As you look at this figure, you will notice that there are a series of tabs on the left side of the screen. These tabs correspond to the various Amazon services. For instance, the Amazon EC2 tab is selected in the figure, but there are also tabs for Amazon S3, Amazon Glacier, and all of the other Amazon services.



The Amazon Web Services Simple Monthly Calculator is the best tool available for estimating the cost of using AWS.

Each tab is designed so you can enter information about how you plan to use the corresponding service in an effort to determine the estimated monthly cost. The Amazon EC2 tab, for instance, is set up so that you can add a new row for each virtual machine. The monthly cost of each virtual machine is displayed, as is the total estimated monthly bill. You can even

use the cost calculator as a tool for experimenting with various ways to potentially decrease your monthly costs.

As you saw in the previous figure, the AWS pricing calculator works really well for helping you to estimate your monthly costs. However, there is one big caveat. The calculator's accuracy is based solely on your ability to enter information that accurately reflects your intended resource usage.

Entering accurate information can be tough. How can you estimate, for example, how many GET requests your workloads will perform in a given month if the workloads are brand new and you have no history to reference for answers?

Unfortunately, there are no easy answers to this question. At best, your usage estimates may be based on an educated guess. At worst (and yes, I have actually seen this), the estimates might be based on random numbers when the administrator has no idea what to enter. As such, it's conceivable that you could suffer from a bit of sticker shock when the bill arrives. Fortunately, Amazon does give you some tools you can use to keep the costs in check.

Amazon provides billing information to customers from within the Billing and Cost Management Console and Dashboard. You can access these resources by logging into the AWS Management Console and then opening the Billing and Cost Management Console (https://console.aws.amazon.com/billing/home).

When you open the link listed above, you are taken to a dashboard containing your billing information. The dashboard contains graphs and other cost visualizations, but the most helpful feature is the ability to create alerts on cost limits. In other words, if you need to stick to a budget, you can use Amazon CloudWatch to automatically generate billing alerts at preset threshold values. For instance, you might generate an alert when the costs you have incurred for the month reach

80% of your budget. That way, you will know when you are coming close to reaching your budget, and you will have the opportunity to shut down some services (if necessary) before your budget is exceeded.

How Do I Get Started?

If you have decided to take the plunge and try out the Amazon cloud, then you may be wondering how you can get started. You can create an account by going to http://aws.amazon.com and clicking on the Sign Up button.

Believe it or not, Amazon makes it really easy to try out AWS without spending any money. When you subscribe, Amazon gives you access to the free tier. The free tier allows you to run workloads in the Amazon cloud for free for the first year.

It's worth noting that the free tier does not give you the ability to use unlimited resources and services for free. There are some usage caps. Amazon could potentially change these limits at any time, but as of the writing of this book, the free tier specifically allows AWS customers to try out Amazon EC2. Each month, AWS customers operating in the free tier get the following EC2 resources for free:

- 750 hours of EC2 running Linux, RHEL, or SLES t2.micro instance usage
- 750 hours of EC2 running Microsoft Windows Server t2.micro instance usage
- 750 hours of Elastic Load Balancing plus 15 GB data processing
- 30 GB of Amazon Elastic Block Storage in any combination of General Purpose (SSD) or Magnetic, plus 2 million I/Os (with Magnetic) and 1 GB of snapshot storage

- 15 GB of bandwidth out aggregated across all AWS services

- 1 GB of Regional Data Transfer

Keep in mind that you only have access to the free tier for a year. After that, you have to pay for any resources you use (you only pay for the resources you consume, and there are no minimums). Similarly, if you are operating within the free tier, and you consume resources beyond those that have been allotted, then you are going to be billed for the excess resource consumption.

What About Virtual Machines?

Even though Amazon offers a wide variety of services, it's Amazon EC2 that tends to get the most attention. EC2, as you will recall, is Amazon's cloud-based virtual machine service. EC2 is such a major part of Amazon's overall offerings in fact, that it will be my primary area of focus for the rest of this book.

Incidentally, as I talk about EC2, I'm going to focus mostly on Windows virtual machines, but Linux virtual machines are also supported. As you would probably expect, there are sometimes differences in the way that administrative tasks have to be performed, depending on whether the virtual machine is running Linux or Windows.

Amazon EC2 Terminology

One of the keys to being able to use Amazon EC2 in an effective manner is to understand the terminology Amazon uses. If you've worked with virtual machines in your own datacenter, then you are probably familiar with terms such as virtual machine, VM template, and static IP address. Well, Amazon EC2 makes use these same basic concepts, but Amazon has a tendency to call things by names that are different from what you might be used to.

To give you a more concrete example, a logical representation of a physical machine is commonly referred to as a virtual machine, a VM, a virtual server, or a virtual computer. For whatever reason, Amazon has decided to refer to virtual machines as instances.

In virtualized environments, it's common for administrators to use virtual machine templates (sometimes called VM templates) to deploy pre-configured virtual machines. Like pretty much every other virtualization platform, Amazon EC2 allows for the use of virtual machine templates. However, Amazon refers to these templates as Amazon Machine Images. You may sometimes see this term abbreviated as AMIs.

When you create a new virtual machine within Amazon EC2, you do so by using an Amazon Machine Image to create an instance. Of course, instances are not always identical to one another. They can vary based on their guest operating system, hardware configuration, or any number of other factors. That being the case, Amazon offers a large collection of Amazon Machine Images.

In Amazon speak, the Amazon Machine Image refers to the way the instance's operating system and other software is configured. You will also occasionally encounter the term Instance Type. An instance type refers to the hardware allocated to the instance (CPU, memory, storage, and networking capacity).

There is one more EC2-related term you need to be aware of (most of the other terms Amazon uses are pretty standard). This term is Elastic IP Address. An Elastic IP Address is really just a static IP address for dynamic cloud computing.

Creating an Instance

Now that I have talked about some of the terminology Amazon uses, let's take a look at how to create an instance. The process for creating an instance is actually pretty straight forward. You

will begin by logging into the Amazon EC2 console (https://console.aws.amazon.com/ec2/) and then choosing the Launch Instance command from the console's dashboard.

At this point, the console will display the Choose an Amazon Machine Image (AMI) page. As you'll recall, an Amazon Machine Image is really just a template used in the creation of virtual machines (or Images as Amazon likes to call them). Amazon makes lots of different Amazon Machine Images available. Some of these images are used to create Windows instances, while others are used to create Linux instances. In some cases, images are preconfigured to host certain infrastructure roles or to run applications. I highly recommend you take the time to review the available Amazon Machine Images. Keep in mind however, that not every Amazon Machine Image is eligible for use within the free tier.

Once you have chosen an Amazon Machine Image, the next step is to choose the instance type. Once again, the instance type refers to the hardware resources that will be allocated to the instance. As was the case for Amazon Machine Images, not every instance type is available for use within the free tier.

Once you have made your selection, click on Review and Launch. When you do, the console will display the Review and Launch page. This page contains a number of different settings you can implement for the instance you are creating. These settings tend to be security related, and there aren't any overly difficult configuration options.

For instance, Amazon gives you the opportunity to link the instance with a security group (you can create a security group if you don't already have one). You can also configure a DNS name for the instance, and you can associate a key pair with the instance you are creating.

Key pairs (public and private keys) are used for encryption, and the option to choose a key pair is the one gotcha that exists in

the instance creation process. Amazon gives you three options for associating a key pair with an instance:

1) You can use an existing key pair. This is the option you will probably use most often.

2) You also have the option of creating a new key pair. This option is good if you don't already have a key pair or if for some reason you don't want to use an existing key pair.

3) Finally, you have the option of not using a key pair at all. This is the option I warned you about. If you create an instance without a key pair, you will be unable to connect to that instance. Therefore, you should always use either a new or existing key pair. In either case, make sure you keep track of which key is used for which instance. Otherwise, you may have trouble logging into the instance.

What About Licensing?

Before I move on, I want to take just a moment and talk about virtual machine licensing. One of the big questions people often have when they first get started with building virtual machines in the public cloud is how those virtual machines are licensed.

I recommend taking the time to read Amazon's license terms so you will be familiar with the various nuances as well as any changes Amazon might make after this book is released. However, the basic concept used within the licensing process is quite simple.

As a general rule, any software provided by Amazon is licensed by Amazon. Suppose for instance you create a brand new virtual machine running Windows Server 2012 R2. You don't have to worry about purchasing a license for this virtual machine (instance), because Amazon licenses the operating system for you.

So what happens if you want to run an application on the Windows Server instance you just created? If you are the one who provides the software, then you are the one who is responsible for licensing the application. If on the other hand, Amazon provides the application, the cost of the application license is built into the price of the instance.

Things can get a little bit trickier (but not by much) when you begin migrating virtual machines to or from the Amazon cloud. If you have a virtual machine currently running in your own datacenter then presumably you have licensed all of the software running on that virtual machine. If you decide to migrate that virtual machine to the Amazon cloud (not all VMs can be migrated, but some can) then Amazon applies their own operating system license to the newly migrated instance. You are of course still responsible for licensing any applications that may be running on the instance.

If on the other hand, you decide you want to move the instance back to your own datacenter, then you can't take the Amazon licenses with you. It will be up to you to license the virtual machine for use on premises. You can read more about Amazon's licensing guidelines at:
https://aws.amazon.com/ec2/vm-import/

Connecting to an Instance

By default, Amazon allows you to establish up to two simultaneous remote sessions to Windows virtual machines so long as those sessions are used for administrative purposes. If you need more simultaneous sessions, then you will need to acquire an RDS license.

So with that said, connecting to a running instance is easy. Open up the EC2 console, select your instance, and then click on Connect. This will cause the Connect to Your Instance page to be displayed. Now you would probably expect the next step

to be entering your username and password, but first you have to figure out what the instance's default password is.

If the instance is a domain member then you can log in using domain credentials. Otherwise, you are going to have to fall back on that key pair you specified when you created the instance.

Click on the Get Password option, and then click Browse, and select the private key file that was used when you created the instance. Use the Open option to copy the file's contents into the dialog box. Next, choose the Decrypt Password option. The default password for the instance will now be displayed.

The next step is to download the Remote Desktop File. You can then use this remote desktop file to connect to the instance. Log in using the built-in Administrator account and the password you retrieved using the private key file.

Deleting an Instance

As previously discussed, Amazon charges its customers for the resources they use. That being the case, it's clearly in your best interest to prevent virtual machine sprawl. After all, you don't want to be charged for virtual machines you are no longer using. As a best practice, you should delete virtual machines once they reach the point where they are no longer needed.

Amazon refers to the process of deleting a virtual machine as terminating an instance. It's pretty easy to terminate an instance, but you have to be careful to only terminate instances when you are absolutely sure they are no longer needed, because once an instance is terminated, it's gone for good.

To terminate an instance, open the EC2 console and choose the Instances option to display a list of your instances. Now, go ahead and locate and then select the instance that you want to get rid of. Once you have selected the instance, click on

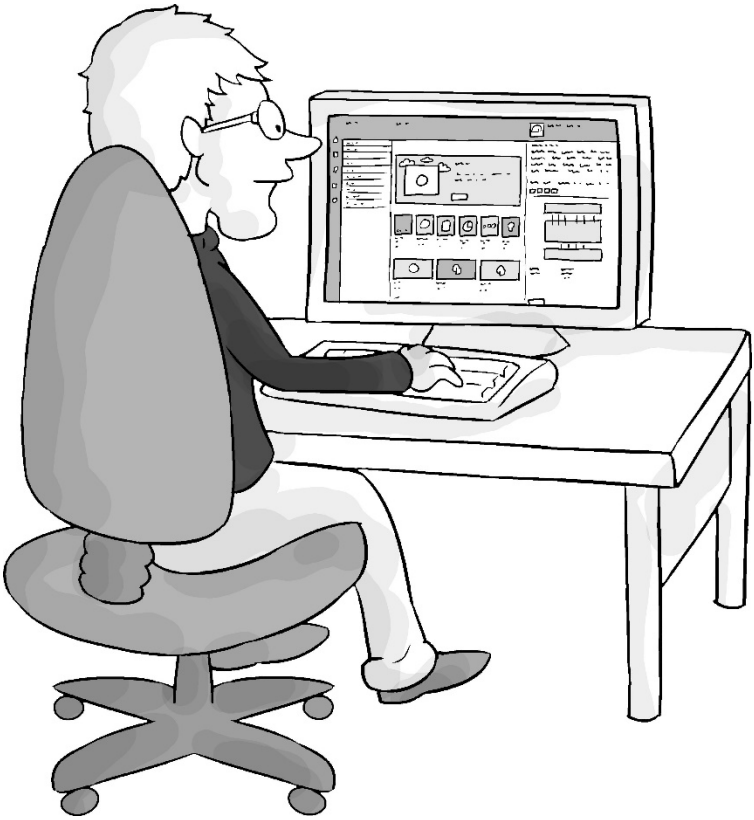
Actions, followed by Instance State, and Terminate. Amazon will give you one last chance to change your mind just in case you picked the wrong instance, or you've changed your mind, or whatever.

When you are ready to remove the instance, click on the Yes, Terminate option and the instance will be deleted. Incidentally, it's normal for the deleted instance to remain visible for a period of time, but it will eventually be removed.

The Big Takeaways

AWS is made up of dozens of individual components and learning about all of them can be a daunting challenge. My advice is to get started by focusing on those components that are most likely to benefit your organization. Once you get a good feel for how those components work, it will be easier to go back and learn about other components. Keep in mind that Amazon is constantly adding new features to AWS, so it is worth occasionally revisiting the list of components to see if there is anything new that might be of use to you.

The AWS Marketplace



It is tempting to think of AWS as being primarily a provider of infrastructure services. Amazon gives you the basic infrastructure building blocks, and you can use those building blocks to create something awesome. Consider Amazon EC2. EC2 is a platform for building virtual machines, which Amazon calls instances. Amazon has taken a cookie cutter approach to the instance creation process. All you have to do is answer a few basic questions, and an automated process creates the virtual machine instance for you. Once the instance is up and running, you can use it for anything that you want.

Infrastructure clouds such as AWS are inherently different from Software as a Service (SaaS) clouds. Infrastructure clouds let you create an eco-system for running your own applications, while SaaS clouds simply rent you a license to run a commercial application.

Although infrastructure clouds and SaaS clouds each serve a distinct purpose, the lines are somewhat blurred between the two. AWS includes a service called the AWS Marketplace that functions somewhat similarly to a SaaS cloud.

The reason why I described the AWS Marketplace as being somewhat similar to a SaaS cloud, is because there is a key difference between an SaaS cloud and the AWS Marketplace. Typically, an SaaS cloud is run by a specific vendor. The Microsoft Office 365 cloud for example, is a SaaS cloud consisting of various Microsoft products. In contrast, the AWS Marketplace contains software from a variety of vendors.

Right now the AWS Marketplace is divided into eight basic categories, and there are 3782 marketplace items listed.

Another way of thinking of the AWS Marketplace is that it is a lot like the app store that is found on your tablet or smart phone. Of course the “apps” in the AWS Marketplace are designed to run in the AWS cloud, not on a smart phone, but like the app store on your smart phone, the AWS Marketplace contains a wide variety of applications that you can use.

Although the AWS Marketplace looks, feels, and even functions a lot like the app store on a smart phone, comparing the AWS Marketplace to a run of the mill app store isn’t really a fair comparison, because the AWS Marketplace is more tightly controlled than the typical app store.

The app stores that exist on our tablets and smart phones are something of a free for all, in that anyone can publish an app so long as they adhere to a few basic rules. Consequently, the app store content is widely varied. Spend a little bit of time browsing the app store on your favorite device, and you are sure to find that some apps are professional grade, while others are super amateurish. Some of the apps work really well, while others are full of bugs. Some apps are clean and easy to use, while other apps cram so many ads onto the screen that there is barely room for a user interface.

Given the widely varying quality of the apps in the average app store, I think that it's safe to say that we've all probably downloaded an app at one time or another that just wasn't all that it was cracked up to be. Perhaps when that happened, you wondered why the device manufacturer would allow such an app in their app store.

I don't really know the official reason why device vendors allow inferior apps into their app stores. For some vendors, it may have something to do with free expression. For other vendors, it may be that they simply lack the resources to thoroughly evaluate every app that gets submitted. Regardless of the reason however, I think that the quality of the apps in the app store is a direct reflection of the vendor's business objectives.

Think about that one for a moment. What is the primary business objective for a smart phone vendor? They want to sell smart phones, right? That being the case, it is clearly in a smart phone vendor's best interest to have as many apps in their app store as humanly possible. The smart phone vendor needs to convince potential customers that the device will be able to do whatever it is that the customer wants to use it for.

One of the most commonly cited reasons for the demise of Windows Phone was that the Windows Phone app store contained far fewer apps than were offered on competing platforms.

So with that in mind, consider Amazon's business objectives. AWS is first and foremost an infrastructure cloud. As such, Amazon's main business objective for AWS is to sell infrastructure. The AWS Marketplace is a relatively small part of Amazon's business.

So why is this important? Well, Amazon knows that it is the quality of its infrastructure offerings that will attract and retain AWS subscribers. They also know that nobody is going to choose AWS over a competing cloud service such as Azure based solely on the number of applications that can be purchased in an app store.

Unlike the smart phone market, which is largely geared toward consumers, AWS caters to businesses who want to run mission critical workloads in the cloud. As such, Amazon knows that anything that it offers through AWS has to work well enough to satisfy Amazon's high end corporate customers. That being the case, Amazon simply cannot afford to allow junk apps to be published to the AWS Marketplace. Can you imagine if a Fortune 500 company tried using an AWS Marketplace application, only to discover that it wasn't stable? Amazon would probably suffer some serious consequences.

The point is, that because of the nature of its business, Amazon cannot allow just any application to be made available through the AWS Marketplace. Applications that are offered through the Marketplace are vetted to ensure that they are stable, reliable, and are of good quality. The application vendor has to demonstrate a certain degree of competency before their app will be added to the AWS Marketplace.

You can access the AWS Marketplace at
http://aws.amazon.com/marketplace

Notes

Easily “converse” about Amazon Web Services (AWS) in any setting.

Cloud computing has changed IT like few other things, and has been one of the defining technologies of the last decade. Even so, cloud computing can be somewhat ambiguous – it's become an alphabet soup of acronyms and over-hyped buzz words, making those who want to leverage unsure of where the cloud even begins. In this book, you will learn about some of the more important AWS components, as well as about important concepts such as licensing, cost control, virtualization and the AWS Marketplace.

About Brien M. Posey

Brien Posey is a 15 time Microsoft MVP and an internationally published author and conference speaker, with over two decades of IT experience. He is currently in this third year of training as a commercial Scientist-Astronaut Candidate. (http://www.brienposey.com/space). Follow him on Twitter @BrienPosey