

# [EC2Instances.info - A Handy Interactive Guide to AWS EC2 Instance Sizing and Pricing](http://ec2instances.info)

One of the most challenging aspects of the AWS ecosystem is navigating the pricing and sizing options when looking at EC2 instances. Luckily, there is a rather nifty tool out there which has been created by a community member and hosted on GitHub which you can find at <http://ec2instances.info>

The [ec2Instances.info](http://ec2instances.info) site lets you dig around all of the different configuration options including (at the time of this blog):

- EC2 Instance types by region
- Reserved Instance options
- RDS Instance types (also at <http://rdsinstances.info>)
- Pricing for On-Demand licenses such as Windows and SQL
- Hourly/Daily/Weekly/Monthly/Yearly pricing detail

You can also see and contribute to the code [directly on GitHub by visiting the source repository](#).

This is a very helpful resource that you should bookmark for reference. The project is being updated by 53 contributors (at the time of this blog) and has well over a 1000 stars on the GitHub project.

You can see from the column selector that there is a lot of potential data to show:

EC2 RDS

Region: US East (N. Virginia) Cost: Hourly Reserved: 1 yr - No Upfront Columns Compare Selected Clear Filters

Filter: Min Memory (GB): 0 Compute Units: 0 Storage (GB): 0

Name	API Name	Memory	Compute Units (ECU)	Arch	Net
Cluster Compute Eight Extra Large	cc2.8xlarge	60.5 GB	88 units	64-bit	10 Gbps
Cluster GPU Quadruple Extra Large	cg1.4xlarge	22.5 GB	33.5 units	64-bit	10 Gbps
T2 Nano	t2.nano	0.5 GB	Burstable	64-bit	Low
T2 Micro	t2.micro	1.0 GB	Burstable	32/64-bit	Low
T2 Small	t2.small	2.0 GB	Burstable	32/64-bit	Low
T2 Medium	t2.medium	4.0 GB	Burstable	64-bit	Low
T2 Large	t2.large	8.0 GB	Burstable	64-bit	Low
T2 Extra Large	t2.xlarge	16.0 GB	Burstable	64-bit	Medium
T2 Double Extra Large	t2.2xlarge	32.0 GB	Burstable	64-bit	Medium
M4 Large	m4.large	8.0 GB	6.5 units	64-bit	Medium
M4 Extra Large	m4.xlarge	16.0 GB	13 units	64-bit	High
M4 Double Extra Large	m4.2xlarge	32.0 GB	26 units	64-bit	High
M4 Quadruple Extra Large	m4.4xlarge	64.0 GB	53.5 units	64-bit	High
M4 Deca Extra Large	m4.10xlarge	160.0 GB	124.5 units	64-bit	10 Gbps
M4 16xlarge	m4.16xlarge	256.0 GB	188 units	64-bit	20 Gbps
C4 High-CPU Large	c4.large	3.75 GB	8 units	64-bit	Medium
C4 High-CPU Extra Large	c4.xlarge	7.5 GB	16 units	64-bit	High
C4 High-CPU Double Extra Large	c4.2xlarge	15.0 GB	31 units	64-bit	High
C4 High-CPU Quadruple Extra Large	c4.4xlarge	30.0 GB	62 units	64-bit	High
C4 High-CPU Eight Extra Large	c4.8xlarge	60.0 GB	124 units	64-bit	10 Gbps

Big thanks go out to [Garret Heaton](#) for putting this together and sharing it out with the community. Nicely done!

## [Understanding the Costs of Public and Private Cloud](#)

There have been a lot of articles over the last few years around public cloud versus private cloud. These have spoken to both sides of the argument that one may be better than another, but we often get lost in the classic case of “everyone I talk to” which presents a rather unscientific sample set to derive real data from.

There was an [interesting post put out recently by Matt Asay](#) which touched on the difficulties in dealing with private clouds. The conversation that triggered it, [or a tweet in this case](#), was based on some licensing hurdles that someone was being faced with. What was interesting is that it led Matt to theme the article around the advantage of public cloud over private cloud using the Microsoft licensing issue as the trigger.

Licensing in the cloud is difficult, period. Trust me, I’ve had to evaluate licensing options for software covering everything from Microsoft tools, to Oracle, to Red Hat Enterprise Linux, and dozens of other products and platforms. Licensing is challenging at the best of times to figure out for some products, but it becomes particularly adventurous on elastic workloads. It is especially brutal

when you get into mixing CPU, socket, vCPU, server, and client licenses.

Let's put licensing aside for a minute though, and get back to the basics of why we see wins and challenges (no, I won't say losses) in private and public clouds. Here are a few of my thoughts on the wins and challenges with either platform. There are many things that may not be covered here, so comments are always welcome if you have more to add. This is just meant to cover the top of mind things that I have discussions around with customers and peers.

**DISCLAIMER/WARNING:** These are just a few reasons on either side of the discussion. I know there are many more, and there is more detail on each, but I just wanted to lay the foundation a bit here.

## **What are some Wins with Private Cloud?**

**Control of cost:** You know what it costs to run your infrastructure every month. Right or wrong, and all of the utilization percentage questions aside, the predictability of the costs is a real win for many organizations.

**Data Locality:** Keeping the data close to the compute workload and to the consumer of the data. This is a hard reality that many discover. We have TB of data for a reason, so it's not always appropriate to fire it up into the cloud when it's used by desktop applications or report cubes which reside in a data center just to name a couple of examples.

**Data Compliance:** It's 10 AM, do you know where your sensitive data is? Yes, it's in my data center.

**Lifecycle Control:** One thing that we do enjoy is having a sense of control over the lifecycle of the applications and infrastructure. We may not quite be ready for rapid release cycles and updates that can be thrust upon us as public cloud consumers. We have to admit that it's called legacy for a reason, and whether we like it or not, the organization may not quite be ready for the forced agility that can come with a provider controlled environment.

## **What are some Challenges with Private Cloud?**

**Security:** If you haven't included security in your workflow and planning when deploying a private cloud, you will find out quickly that you may have a problem. There are lots of excellent advantages, but added complexity at the same time.

**Compliance:** This goes hand in hand with security, because understanding the security and ACL model of a private cloud is critical for ensuring compliance. Making sure that data protection and segregation in a multi-tenant cloud requires more thought and care than a traditional virtualization infrastructure. It's all there to be managed, but it is not ready to go out of the box.

**Cost:** Private clouds require infrastructure maintenance. Both physical and software resources are going to be a part of both capital and operational expenses. If your environment is truly elastic and varied, you can often find yourself having to heavily over provision in order to keep up with demand.

**Complexity:** Both setup and operation of private cloud infrastructure can be expensive and difficult. It isn't to say that it is always this way, but I've seen a lot of teams face both technical and organizational problems adopting private clouds as a tool.

**Knowledge:** Who knows your environment the best? As some companies hire the supposed 10x engineers to build a cloud platform, they have to also be sure that 10x engineer is doing 10x documentation and training. Many private cloud builds go well, but the engineers who design them

may go elsewhere and leave a challenging platform to manage going forward without the knowledge.

## **What are some Wins with Public Cloud**

**Agility to deploy and manage:** Base setup is already completed. Simply leverage existing service offerings which can be done easily through self-service portals and via externally facing APIs. It really seems like it couldn't be simpler than that.

**Try and Buy:** Kick the tires on a service, and if you like it, evaluate the viability for cost, ROI, performance and more. The advantage is that you can try before you buy and if you don't get the results, you are

**Burst capability:** Need 100 instances for a day or a week or a month? No problem. Need 1000 instances for a day or a week or a month? No problem. It's really just that easy. Putting load balancing gear in front of the environment is often baked into the public cloud platform too, so you really don't have to go elsewhere to design your resilient, elastic application platform.

**Innovate at a rapid pace:** Amazon has thousands (that's a total ballpark based on some estimates it's about 3000) of software engineers. I doubt that many organizations could have that pool of resources to design, build, and innovate their platform. Some do, like some of the large application environments and financial institutions, but it's not as common to have thousands of staffers dedicated to software innovation.

**Continuous improvement of service (usually):** It's assumed that the service will become better over time as the greater use of the services. More customers will drive more new development and features. That's a major value proposition of the cloud services.

## **What are some Challenges with Public Cloud?**

**Data gravity:** Unless your data is close to where it is being consumed by applications, there is a little thing called latency that will become very obvious to you.

**Cost:** I both agree and disagree that public cloud is cheaper. For elastic workloads there are distinct cost advantages. The ability to get on-demand expansion and to retract as needed to reduce costs is undeniably there in public, but not private cloud. That said, persistent workloads can be, and often are, more expensive on public cloud platforms.

**Security:** Where is my data? Who may have access to it? Has there been a breach? What auditing is available? The list goes on with regards to security

**Regulatory issues:** PIPEDA (Personal Information Privacy and Electronic Documents Act) is one I've been very close to over the years. This is just one example, and many more exist. European nations have strict regulations around data storage, encryption, and privacy. Other countries have specific limitations on the use of encryption.

**Hey, what happened to Service X?:** If you bank on a cloud service, it may also disappear. In other words, just like any other software or hardware platform you choose, it could have a fixed lifecycle that is outside of your control.

## **Forget Big Data, How about Bad Data**

I've worked in organizations that use file serving as a major part of their business. Yes, are always told that all data should be in the cloud. When you have dozens of terabytes of data that is being used for Excel, Word, and other day-to-day productivity applications, the reality is that data is expensive to keep in the cloud. It is expensive because it is big, with odd read/write patterns, and it requires quite low-latency access to the applications that run it. And yes, those applications are on lots of desktops.

We all agree that Office365 can solve this, but the very same organizations that I've seen running terabytes of Excel, Word, and Access files, are in Financial Services, or Health Services. Cloud is not an option due to regulations as highlighted above. That will change at some point probably, but not for quite some time.

## **Both Options Win**

Let's be honest about this and admit that both options have legitimate wins, challenges, and in some cases there are absolute limitations that prevent choosing one of the options.

What I do want to emphasize here is that public cloud did something wonderful for both business and technology. Public cloud taught us that it could be done differently, and that agility in infrastructure gives real tangible and intangible benefits. I'm a huge proponent for public cloud, but for the right reasons. Here are some that i think stand out when we look at what the public cloud is able to deliver today, and as it evolves.

And one thing I have to add is that you should definitely [read a lot of what Matt has to say elsewhere](#) as he's a great writer and covers some excellent topics and has been a contributor in many ways to the tech community.