Choosing the Right EC2 Instances to Optimize Your Cloud

How to Find the Best Instances for Cloud Cost Optimization

Contents

Introduction

With many enterprises, AWS EC2 instances account for the largest piece of cloud costs. As these organizations grow, so do the complexities of choosing the EC2 instances that best fit their workloads. The costs of operating in the Amazon cloud might fly under the radar for a while, but once finance and executive members start to see usage costs spike, perhaps due to successful projects and growth, there will be some difficult questions to answer if you don't have the visibility you need to understand how your business operates in the cloud.

Use this guide to understand the strengths and opportunities that each EC2 instance family delivers and to learn how Cloudability helps you to build a cost-effective, efficient cloud infrastructure.

PART 1:

The Basics

What Is Amazon EC2?

EC2 is an AWS service that provides compute capacity in the cloud. Like a server, an EC2 instance has resources like a CPU, an operating system, local storage, RAM, etc. With EC2, you can easily build an Amazon Machine Image (AMI) that's secure, is exactly what you need and is available in minutes.

Those server images are called instances. When you spin one up, you choose the instance type, then launch the number of instances you need. You can do this manually, from the Management Console or programmatically. You only pay for what you use and, when you're done with your instances, you spin them down and stop paying for them.

If It's Not the Perfect Fit, Then There's Probably Waste

There are many EC2 families, each optimized for various types of workloads like general purpose, compute and memory. For enterprises with multiple teams, choosing the best-fitting families can affect efficiency across thousands of instances.

Fit is important when choosing AWS EC2 instances as it's critical toward maintaining cost efficiency. Teams that take the time to identify EC2 instances that fit their workloads best will attain better savings and performance, as those instances will likely be optimized for the team's specific workloads.

Here are a few general tips for getting a handle on your EC2 costs:

Tip 1 - Many customers assume that an older generation of an EC2 family will be cheaper than the newest generation, but usually the opposite is true. Newer generations run on newer processors that usually require less power and cooling. Simply put, the newer generation is cheaper to operate, so Amazon charges you less.

To save money, take a look at the most recent addition to a family. They generally offer the best price-performance ratio. For example, M5 instances deliver 14% better price-performance than M4 instances on a per-core basis.

Tip 2 - EC2 prices vary from region to region. If you can be flexible about where your EC2 instances live, then taking the time to do some price comparisons can really pay off.

Tip 3 - Automate turning on and turning off your instances. You'll save money if you don't rely on doing these actions manually. Even something as simple as turning text environment instances off over the weekend can make a big difference on your bottom line.

Tip 4 - Get some hard data on what your application actually needs. Profiling an application is so important, and so often neglected, that we'll be saying this again.

PART 2:

General Purpose

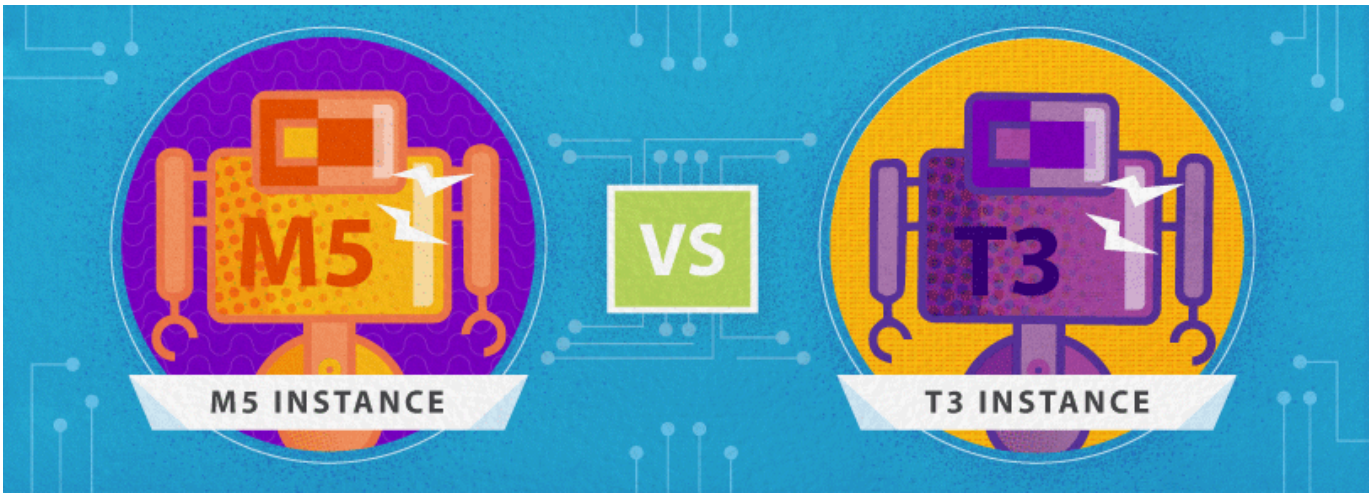
M & T: The General Purpose Families

Designed for general purpose workloads, the M and T families are the workhorses of EC2, but there are some differences between the two.

M5: All About the Balance

The M5 family, the latest addition to the M family, has a good balance of CPU, RAM and disk size/performance, making it the best choice for applications with consistent performance needs. If you're not sure what the performance demands of your application are, an M5 instance might be a good place to start. You can monitor your application's performance for a while to see if its performance is limited by one of the hardware characteristics. If it is, you can switch over to another family that's more specialized.

In general, the M5 family is a good fit for web and application servers, back-end servers for enterprise applications, gaming servers, caching fleets and app development environments. Many companies run their production applications on M5 instances.



Their specs might be similar, but understanding the performance capabilities of both families can help AWS EC2 users choose the most cost-effective instance family and size to start with.

In addition to the standard M5 instances, there's also the M5a. M5a instances are powered by a custom AMD EPYC processor running at 2.5 GHz and are priced 10% lower than comparable M5 instances, which use Intel processors. At the same time, M5a instances have a lower EBS bandwidth than M5.

A FEW THINGS TO NOTE ABOUT THE M5 FAMILY:

- Family sizes begin with the m5.large (no medium, small, etc).
- M5a and M5 instances require EBS volumes for storage. (M5d has attached SSD storage.) Remember that EBS volume costs are in addition to the EC2 instance costs,
- The M5 family is a great place to start, but make sure you measure the performance of your full application, using realistic loads, so you'll know if it's the best family for you.

T3: Burstable Compute

The T3 family is a lower-cost option than the M family. It's also aimed at general purpose workloads, but is meant for applications that are "bursty," meaning it's only CPU-intensive in bursts. It turns out that many applications fit that profile.

One example is a continuous integration server or a build server. There are lots of short bursts while a build is going on, but the server's idle the rest of the time. Average CPU

usage is low, but when the build is running you're getting great performance, can do the build quickly and make your developers happy. Other common use cases for T3 are low throughput applications such as administrative applications, low-traffic websites, development and testing.

T3 instances accumulate CPU credits when a workload operates below the baseline threshold (amount of CPU utilization). Each CPU credit means the T3 instance can burst with the performance of a full CPU core for one minute.

Amazon thinks of the model for CPU credits as a bucket. When the instance is operating below the baseline, credits go into the bucket. When it operates above the baseline, credits come out of the bucket. You can only accumulate a certain number of credits over a 24-hour period because that's how much the bucket holds. The baseline and number of CPU credits earned per hour is determined by the instance size.

By default, T3 instances are created in Unlimited mode, which means they can burst for as long as they need to. For example, a t3.nano instance earns 144 CPU credits over a rolling 24-hour period, which it can redeem for 144 minutes of 100% vCPU use. When it depletes its CPU credit balance (which is represented by the CloudWatch metric CPUCreditBalance), it can spend surplus CPU credits — credits it has not yet earned — to burst for as long as it needs. If the balance is depleted and more than 144 excess credits are used, the instance can still run, but the extra use leads to additional billing.

A COUPLE OF THINGS TO NOTE ABOUT THE T3 FAMILY:

- Maximum accumulation of credits is based on instance size
- For running instances, credits never expire
- For stopped instances, credits are stored for up to 7 days and then they expire
- Credits expire when an instance is terminated
- The T3 family offers smaller sizes than the M5 family

A1: For Arm-Based Applications

A1 instances are the only EC2 instances designed specifically with Arm-based applications in mind. A1 instances are powered by AWS Graviton Processors that feature 64-bit Arm

Neoverse cores — the first custom silicon designed by AWS. This architecture makes them a good choice for web servers, containerized microservices, caching fleets and distributed data stores using the Arm ecosystem.

The Arm architecture also means that A1 instances have a lower price point than M5 or T3. Arm is designed to run with lower power and lower power consumption. The hardware behind A1 instances is cheaper to run, which means AWS charges less to use them. Compare the prices and specs of these comparable M5, T3 and A1 instances for US West (Oregon):

Instance	vCPU	ECU	Memory (GiB)	Storage (GB)	Usage
a1.2xlarge	8	N/A	16	EBS Only	$0.204/hr
t3.2xlarge	8	Variable	32	EBS Only	$0.3328/hr
m5.2xlarge	8	31	32	EBS Only	$0.384/hr

The key is to make sure you have the right workload for the right instance. As you can see, A1 is substantially cheaper to run per hour, but that won't help you if the lower compute power means that you have to run it twice as long as the same process would run on an M5 or T3 instance.

As the number implies, A1 is the first version of this instance family. While primarily Arm-focused, the A1 also holds the potential to run architecture-agnostic workloads that could run on Arm.

What's the Best Start for New Projects?

Deciding between the two general-purpose EC2 instance families is a common debate. Both instance families can serve new projects well. The T3 family is a low-cost means to kick off a project before a team understands where performance bottlenecks might occur. The M5 family has larger family sizes and a balance of CPU, storage and memory that makes it a good choice for understanding performance.

PART 3:

Optimize

Optimized Instance Families

As opposed to the M5 and T3 families, which are general-purpose, optimized families are specifically designed to be the most cost-efficient choice for particular workloads.

C5 Optimizes for Compute

For workloads that require high amounts of compute, the EC2 C5 family is the best instance to spin up. The CPUs in this family run at very high clock speeds. Amazon says that the C5 family offers a 25% price/performance improvement over the C4 instances, with over 50% for some workloads. The C5 family also has additional memory per vCPU, and twice the performance of the C4 family for vector and floating point workloads.

The C5 family is denser than all the previous C generations. There can be up to 72 vCPUs on a single instance. In comparison, the c4.8xlarge has up to 36 vCPUs. The practical effect is that you get much more compute power from a single instance.

The C5 family uses 3.0 GHz Intel Xeon Platinum processors with new Intel Advanced Vector Extension 512 (AVX-512) instruction set. AWS advertises these processors as being optimized specifically for EC2 use. With Intel Turbo Boost Technology, you can run a single core at 3.5 GHz.

The C5 family has sizes that range from large to 18xlarge. If you compare a c5.large to an m5.large, you'll see that the c5.large offers more compute units and runs at a lower price (these prices are for the US West (Oregon) region.

Instance	vCPU	ECU	Memory (GiB)	Storage (GB)	Usage
m5.large	2	8	8	EBS Only	$0.096/hr
c5.large	2	9	4	EBS Only	$0.085/hr

For Windows, the cost is $0.188 per hour for an m5.large and $0.177 per hour for a c5.large. (Note that ECU or EC2 Compute Unit is a measurement Amazon uses to make it easy to compare CPU capacity between different instance types.)

Compute optimization comes with a bit of a trade off. The C5 family has higher available compute units, but a lower amount of memory available across sizes. So if the compute-intensive workload can make the most of the available vCPU units and can go without the extra memory or storage optimization, the C5 instances are a great choice.

If you need a lot of network bandwidth, then reach for C5n. C5n instances offer significantly higher network performance across all instance sizes, ranging from 25 Gbps of peak bandwidth on smaller instance sizes to 100 Gbps of network bandwidth on the largest instance size. In addition, C5n instances also feature a 33% higher memory footprint compared to C5 instances.

Amazon suggests that some good fits for the C5 family are high-performance web servers, scientific modeling, batch processing, distributed analytics, high-performance computing (HPC), machine/deep learning inference, ad serving, highly scalable multiplayer gaming and video encoding. These workloads need lots of compute power, but not a lot of RAM.

Avoid Over-Resourcing for Compute Power

It's not uncommon for IT and operations teams to spin up C5 instances as a starting point because they assume that compute power will be the bottleneck for their projects. This may be the case but, as always, profiling and stress testing your application is the way to really understand what's the best fit.

Even if the application does require a lot of compute power, only profiling will tell you if you're using the right sizes and numbers of instances so you don't waste money on overprovisioning. For example, do you really need multiple c5.8xlarges or can you get the same performance with a fewer number of c5.4xlarges? Only the data will give you the right answer.

What Are Your Storage Options?

All C5 instances are EBS-optimized by default, so getting dedicated bandwidth doesn't incur any extra charges. Spinning up additional EBS volumes, however, will add to your costs.

But what if you have a compute-heavy application that really needs high-speed, ultra-low latency local storage? For example, video encoding, image manipulation and other forms of media processing often require large amounts of temporary storage. This is because only the input and output files are valuable and these are typically stored in S3, which is much cheaper than EBS. The intermediate files, however, are temporary. For these situations, take a look at the C5d instances. These have local NVMe-based SSDs that are physically connected to the host server and provide block-level storage that's coupled to the lifetime of the C5 instance.

R5, X1 and X1e Optimize for Memory

For AWS users with memory-intensive workloads, the R5 family provides high amounts of memory. This EC2 family gives users one of the the best costs of memory per Gbps, per hour compared to other families, with higher memory to CPU ratios and approximately 8 GiB per vCPU.

Even the smaller R5 instances offer great network performance of up to 10 Gbps. The largest size, the r5.24xlarge, offers up to 25 Gbps. Although most R5 instances require EBS, the R5d family offers local NVMe-based SSDs.

R5a instances use the AMD EPYC processor. They're available in six sizes, with lower per-GiB memory pricing in comparison to the R5 instances. Like M5a, they have lower EBS bandwidths than their normal counterparts.

Amazon suggests you use R5 instances for memory-intensive workloads such as data mining, in-memory analytics, caching, high-performance databases, distributed web scale in-memory caches, midsize in-memory databases and real-time big data analytics.

X1 and X1e

The X1 and X1e families deliver very large amounts of storage and are designed for applications with a huge memory footprint that must be on a single instance.

Initially, the X1 family was targeted at customers who wanted an instance that could support large SAP HANA workloads. The X1e was introduced so that customers could store and process far larger datasets, making them a great fit for larger production deployments. As it turns out, both these families are ideal for any big, in-memory database. Along with SAP HANA, customers use these families for very large data processing systems, such as Apache Spark, Hadoop and even some HPC workloads.

On the network side, the instances offer up to 25 Gbps of network bandwidth when launched within an EC2 placement group, powered by the Elastic Network Adapter (ENA), with support for up to eight Elastic Network Interfaces (ENIs) per instance. The instances are EBS-optimized by default, with an additional 14 Gbps of dedicated bandwidth to your EBS volumes, and support up to 80,000 IOPS per instance. Each instance also includes SSD instance storage for temporary block-level storage.

Note that the regional size flexibility for Reserved Instances does not apply across X1 and X1e.

Considering the Engineering Costs of Migrating

Whether it's better to have a fleet of R5 instances or fewer X1 or X1e instances can be a difficult choice. The X1 and X1e families are more expensive, but managing multiple R5 instances and making sure they are always cost effective can be time consuming, which also means more money will be spent. Another consideration is that with the larger X1 or X1e instances, you may be able to decrease compute time by running your workloads in parallel across all the virtual cores.

Monitoring instance utilization will give AWS users the right data to work with to determine which is the better fit for a given memory-intensive workload. While the X1e is a nice solution with the horsepower to contain multi-instance workloads in one high-powered instance, it might not be worth the migration if engineering and operations teams find more administrative and cost efficiencies from running multiple R5 instances.

G3, P3 and F1 Optimize for Accelerated Computing and Graphics

Accelerated instances have additional hardware that can perform processing functions in addition to the onboard CPU.

G3

The G3 family is a high-performance platform for graphics applications. It's good for 3D visualizations, streaming graphics, server-side graphics workloads and graphics

applications based on DirectX and OpenGL.

In addition to the high-frequency Intel Xeon E5-2686 v4 (Broadwell) processors, G3 instances provide access to NVIDIA Tesla M60 GPUs, each with up to 2,048 parallel processing cores, 8 GiB of GPU memory and a hardware encoder supporting up to 10 H.265 (HEVC) 1080p30 streams and up to 18 H.264 1080p30 streams. With the latest driver releases, these GPUs support for OpenGL, DirectX, CUDA, OpenCL and Capture SDK (formerly known as GRID SDK).

The G3 family is available in four sizes, ranging from the g3s.xlarge to the g3.16xlarge. Here are the specs for those two instances.

Instance	GPU	vCPU	Memory (GiB)	GPU Memory (GiB)	Network Performance
g3s.xlarge	1	4	30.5	8	Up to 10 Gb
g3.16xlarge	4	64	488	32	25 Gb

P3

Traditionally, extremely demanding computational problems could only be tackled by large, well-funded organizations that could afford to set up their own very expensive, very specialized infrastructure. With the advent of the cloud and offerings such as the P3 family, these types of applications become approachable for a much broader range of enterprises. They can spin up as many P3 instances as they need, run their computations, then tear down the system.

The P3 family is designed for applications such as deep learning, HPC simulations and batch rendering. In fact, any workload that can take advantage of GPUs for computational capabilities is a good fit for P3 instances.

Depending on the size, a P3 instance can have up to 8 NVIDIA Tesla V100 GPUs, each pairing 5,120 CUDA Cores and 640 Tensor Cores, as well as high-frequency Intel Xeon E5-2686 v4 (Broadwell) processors. The family also supports GPU peer-to-peer (P2P) communication across the PCI switch. The P3 is available in three sizes. Here are the specs.

Instance	GPU	vCPU	Memory (GiB)	GPU Memory (GiB)	Storage (GiB)	Dedicated EBS Band-width	Network Perfor-mance
p3.2xlarge	1	8	61	16	EBS Only	1.5 Gbps	Up to 10 Gb
p3.8xlarge	4	32	244	64	EBS Only	7 Gbps	10 Gb
p3.16xlarge	8	64	488	128	EBS Only	14 Gbps	25 Gb
p3dn.24x large	8	96	768	256	2 x 900 NVMe SSD	14 Gbps	100 Gb

You'll notice that there's a single P3dn instance at the bottom of the chart. These massive instances are the fastest, largest and most powerful P3 instances. Designed for machine learning, their size means workloads have to be distributed across less nodes while providing greater bandwidth for the data needed behind machine learning. The goal is to significantly lower the time it takes for developers to train their machine learning models.

F1

The F1 family (F stands for FPGA or field programmable gate arrays) is useful for all sorts of workloads that can benefit from specialized hardware designed to meet the demands of the compute pipeline. FPGAs have traditionally been out of reach for most developers because they required a big investment in dedicated, on-premises hardware. The F1 family makes FPGAs accessible to many more organizations.

F1 instances are high performance, easy to access and fully customizable. You can directly access custom FPGA hardware with a few clicks and accelerate application performance over 30 times what's possible with general purpose CPUs. Amazon also provides a full set of design tools, simulators and a hardware development kit. With these tools, you can produce an Amazon FPGA image, or AFI, which contains your custom FPGA design. Once you've created your AFI, you can apply it directly to an F1 instance or offer it to other customers by listing it on the AWS Marketplace.

You can associate multiple AFIs to the same F1 instance and instances can switch between multiple AFIs during runtime without a reboot. Just like the P3 family, the F1 family has peer-to-peer communication over a PCI fabric.

The F1 family offers three sizes. Here are the specs:

Instance	FPGA	vCPU	Memory (GiB)	SSD Storage (GB)	Network Performance
f1.2xlarge	1	8	122	470	Up to 10 Gb
f1.4xlarge	2	16	244	940	Up to 10 Gb
f1.16xlarge	8	64	976	4 x 940	25 Gb

Good fits for the F1 family include image and video processing, genomic sequencing, compression, security algorithms and search/analytics. F1 instances are particularly well-suited for applications that are time sensitive, such as clinical genomics, real-time video processing and financial risk analysis.

Optimized for Storage

Amazon offers three EC2 families that are optimized for storage. Two of them, the H1 and D2, use hard drives. The third, I3, uses SSDs.

H1

The H1 family was specifically designed for big data and data-intensive workloads, including MapReduce, distributed file systems like HDFS and MapR-FS, network file systems, log or data processing applications like Apache Kafka and big data clusters.

Powered by 2.3 GHz Intel Xeon E5 2686 v4 (Broadwell) processors, H1 instances provide up to 64 vCPUs and 256 GiB of DRAM. With up to 16 TB of inexpensive, magnetic storage and Enhanced Networking that provides up to 25 Gbps of network bandwidth per instance, H1 instances are ideal for processing very large data sets.

The H1 family offers six sizes, from the h1.2xlarge to the h1.16xlarge. Here are the specs for these two sizes:

Instance	vCPU	Memory (GiB)	Storage (GB)	Network Performance
h1.2xlarge	8	32	1 x 2,000 HDD	Up to 10 Gb
h1.16xlarge	64	256	8 x 2,000 HDD	25 Gb

Local storage is optimized to deliver high throughput for sequential I/O; you can expect to transfer up to 1.15 gigabytes per second if you use a 2 megabyte block size. The storage is encrypted at rest using 256-bit XTS-AES and one-time keys. Moving large amounts of data on and off of these instances is facilitated by the use of Enhanced Networking, giving you up to 25 Gbps of network bandwidth within Placement Groups.

Amazon says that, compared to the older D2 (dense storage) instances, H1 instances provide more compute and memory per terabyte of magnetic disk, along with increased network bandwidth.

H1 instances use disks that are physically attached to the instance, so make sure your data is replicated so you won't lose anything in case of a hardware failure.

D2

The D2 family uses high-capacity magnetic disks, just as the H1 does, but it has an even higher ratio of disk to CPU and memory, which makes it a good fit for applications such as Massively Parallel Processing (MPP), MapReduce and Hadoop distributed computing, or a distributed storage system that needs a large amount of local storage or streaming throughput.

The D2 family has four sizes, ranging from the d2.xlarge to the d2.8xlarge. Here are the specs for these two sizes.

Instance	vCPU	Memory (GiB)	Storage (GB)	Network Performance
d2.xlarge	4	30.5	3 x 2000 HDD	Moderate
d2.8xlarge	36	244	24 x 2000 HDD	10 Gb

Amazon says that the D2 family offers the lowest price per disk throughput performance on Amazon EC2. The cost efficiency of the D2 family is in the available amount of storage per hourly price. As far as compute and memory go, the larger sizes can provide a decent amount of each, but at a higher premium, as the most benefit per dollar is in the available HDD storage.

Keep in mind that, because D2 instances use disks that are physically attached to the instance, you should make sure your data is replicated so you won't lose anything in case of a hardware failure.

I3: High I/O Optimization

If you're looking for a high number of IOPS for random reads and writes, you'll probably want to start looking at the I3 instances. They're really great for transactional workloads like NoSQL databases, clustered databases and online transaction processing (OLTP) systems. All these workloads require a significant amount of random I/O. In fact, if you need very high IOPS and very low latency, an I3 instance is probably preferable to using another instance family in conjunction with EBS.

The I3 instance features up to 64 vCPUs, high-frequency (2.3GHz) Intel Xeon E5-2686 v4 (Broadwell) processors, NVMe SSD storage capable of 3.3 million IOPS in 4 KB blocks, up to 16 GB/second of sequential disk throughput and enhanced networking capabilities.

There are six sizes available, from the i3.large to the i3.16xlarge. Here are the specs for these two sizes:

Instance	vCPU	Memory (GiB)	Storage (TB)	Network Performance
i3.large	2	15.25	1 x 0.475 NVMe SSD	Up to 10 Gb
i3.16xlarge	64	488	8 x 1.9 NVMe SSD	25 Gb

For workloads that need access to physical resources or workloads that may have license restrictions, there's the i3.metal. Here are the specs:

Instance	vCPU	Memory (GiB)	Storage (TB)	Network Performance
i3.metal	72 logical processors on 36 cores	512	8 x 1.9 NVMe SSD	25 Gb

Remember that for the I3 family, disks are physically attached to the instance, so you'll need to make sure you have data backups and replication.

PART 4:

Visibility & Control

Full Visibility & Full Control Over EC2 Costs

Managing your EC2 costs can seem more than a little daunting. Spreadsheets can help with a small number of instances, but it doesn't take much for an EC2 architecture to scale too large for a spreadsheet to handle.

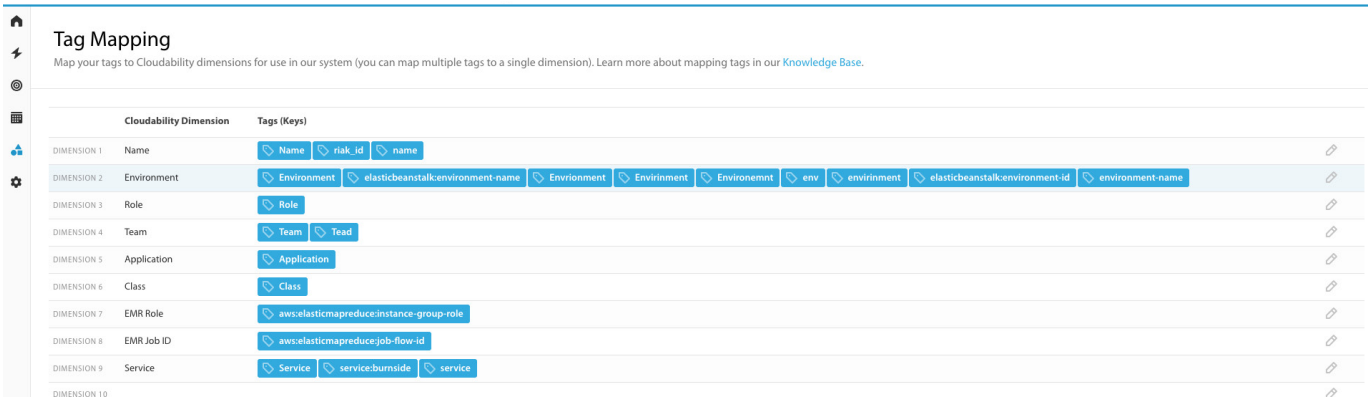
It starts with having enough visibility into your costs and usage to provide actionable insights. Once you have a strong reporting and analysis foundation, you can move on to prediction and cost optimization. A cloud cost management platform like Cloudability is a crucial tool in helping you manage both your current and future EC2 costs.

Keeping Track of Tags

Tags are metadata labels (each with a customer-defined key and value) that you assign to resources so you can keep track of them. Tagging is essential to making sense of the enormous amount of EC2 cost and usage data AWS produces. Cloudability has a few features that let you take full advantage of your tags and share that valuable data with all your stakeholders for allocation, tracking, rightsizing and more.

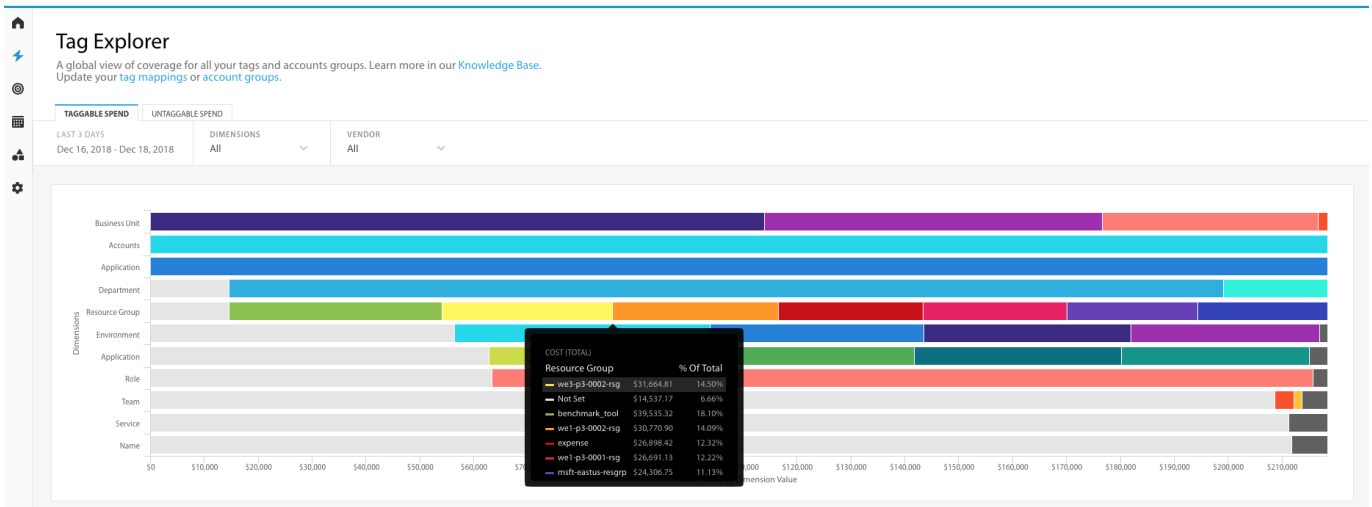
To learn more about building your tagging strategy, check out our "AWS Tagging Strategy Best Practices: Using Tags and Consolidated Billing to Lower Your AWS Spend" e-book.

Tag Mapping



AWS views tags as a collection of characters, and tags need to match exactly if they're going to be lumped together. That means "Environment," "environment" and typos like "envronment" will all be sorted into different tags. Tag Mapping lets you take multiple versions of what should be a single tag and map them to one dimension so you can make sure your tags are grouped accurately.

Tag Explorer

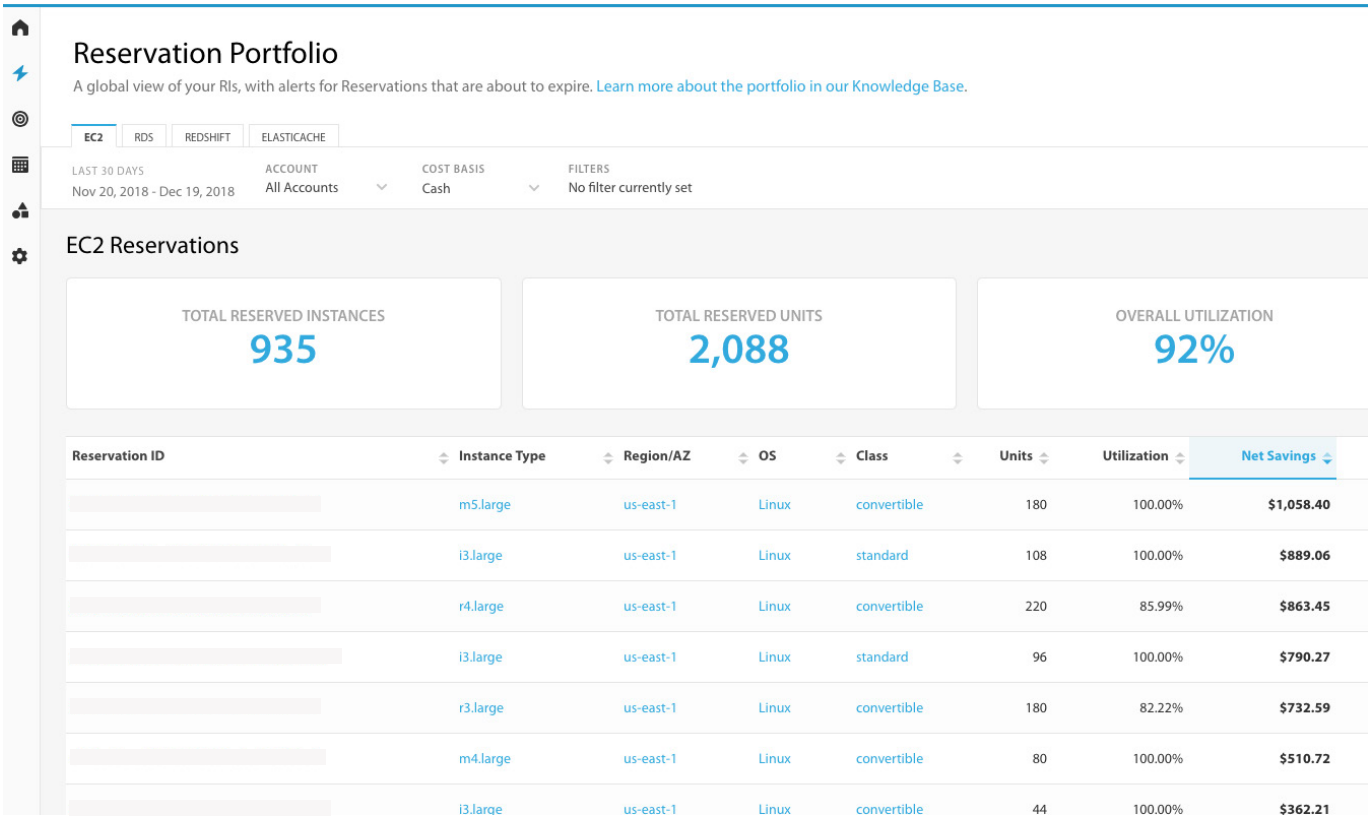


Tag Explorer gives you a global view by sorting all of your resources into tag keys and breaking down those resources by tag values. At a glance, you'll be able to see how your spend is distributed — and which spend isn't tagged.

Managing Your RIs

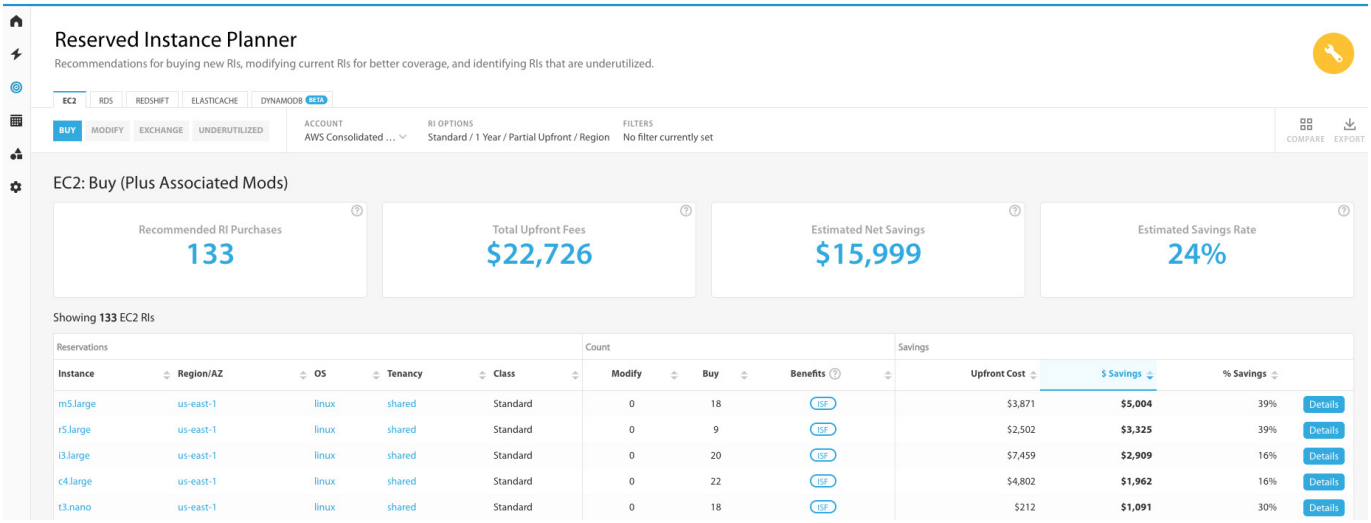
RIs are vital for optimizing your EC2 use and getting the most from your cloud. The right tools will give you the ability to manage your current portfolio and predict your future use so you can make informed purchases. Cloudability has several features devoted to helping you build a solid RI strategy and a dependable RI portfolio.

Reservation Portfolio

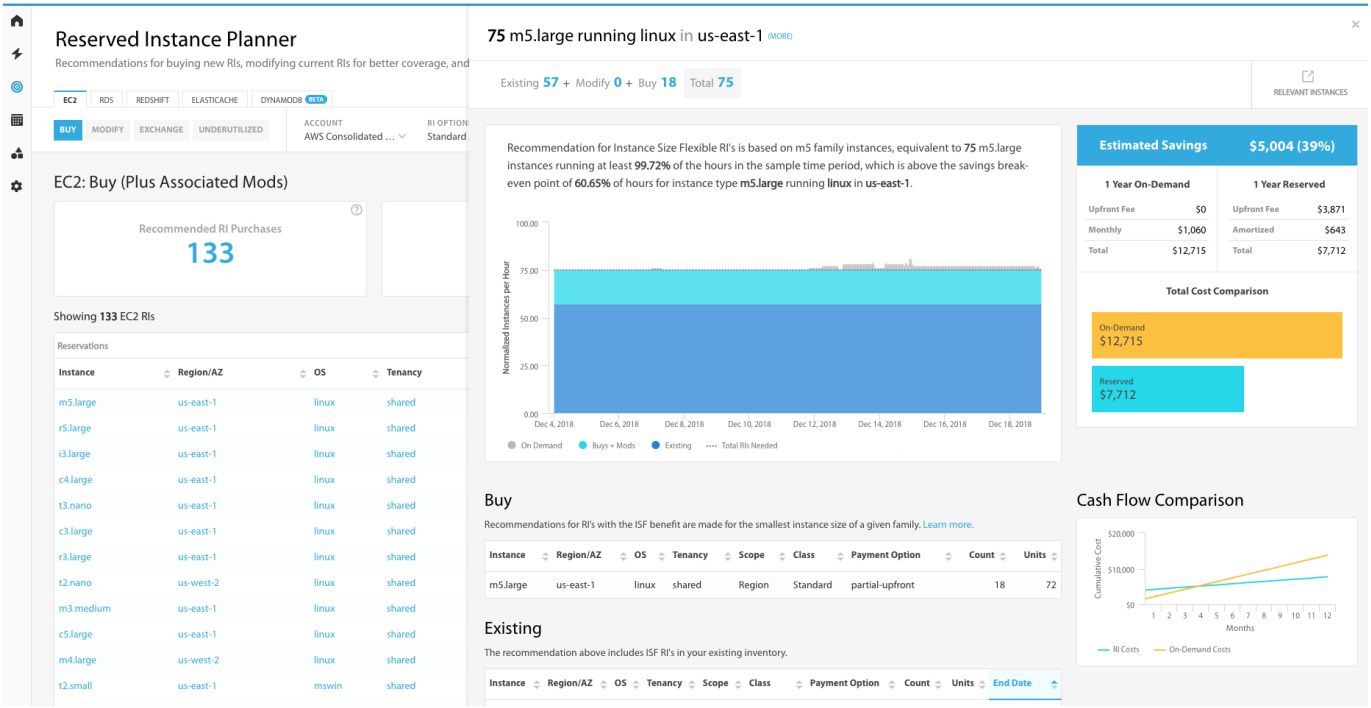


The Reservation Portfolio provides a global view of RIs from all member and payer accounts in a single place. The list is sortable and filterable, and perhaps most importantly, shows the expiration date of each reservation so that you don't get any surprise drops in coverage. It also shows you both your current savings and potential unrealized savings. To make it easy to keep tabs on expiring reservations, the Reservation Portfolio lets you subscribe to a scheduled email alert to warn you about expiring RIs.

Reserved Instance Planner



The RI Planner pulls from your existing inventory and usage data to make recommendations for RI purchases. Using the RI Planner, you can uncover potential savings from RI purchases and modifications. It can also highlight underutilized RIs so you can make sure you're getting your money's worth. By setting your RI parameters, you can focus the recommendations by things like utilization rate, savings rate, term, payment option, etc.



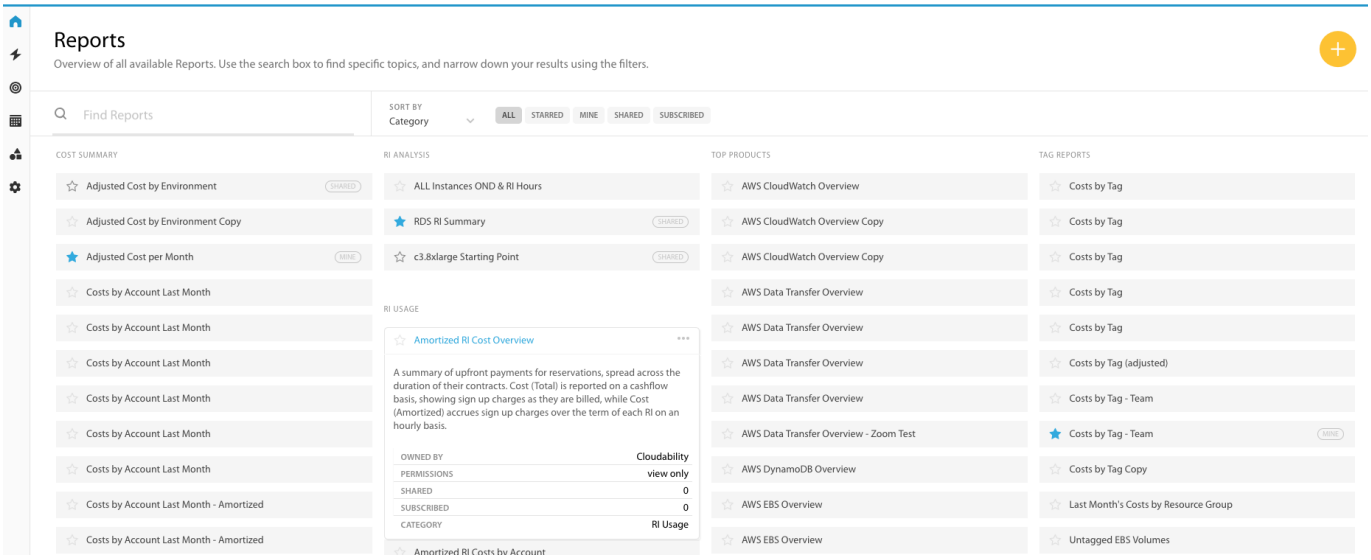
The best way to get the most out of your RIs is to revisit your portfolio and purchases on a regular basis. The RI Planner is vital for these efforts.

Full Visibility & Full Control Over EC2 Costs

Full visibility means getting the EC2 cost data you need when you need it, and any platform you're using should have the capability to give you the exact data you need. Full control means that you never have to worry about EC2 costs exploding from out of nowhere. Cloudability is built around giving you the visibility you need along with the tools to proactively prevent cost spikes and to let you know quickly if a spike does occur.

Reports

Cloudability offers a wide variety of reports. If you want to see something about your cloud costs or usage, then we have a report to show it to you — and that includes EC2. Get reports on key factors like the effectiveness and usage rate of Spot Instances, your RI waste, your RI coverage rate, a list of untagged EC2 Instances, data transfer costs and more. And that's only the prepackaged reports. Reports can be customized to fit the exact metrics that mean the most to your business.



This information is critical for understanding the financial health of your EC2 architecture.

Automation

The Automation feature lets you automatically scale down or stop development and test resources during periods of underutilization. For example, you might want to turn your development resources off during nights and weekends. The Automation feature includes an audit log where you can see each task run, when it was executed and the number of resources affected during the run.

Anomaly Detection

Usage and cost anomalies can add up quickly, and you need to know about them as soon as possible. If a program error spins up a bunch of EC2 instances on Friday at 7 p.m., you want to find out about it so you can shut it down much earlier than when you log in Monday morning. Anomaly Detection prevents issues like that by monitoring your costs and notifying you when there are unusual spending patterns.

Typically, AWS billing files are updated 4-6 times per day. Every time they come in, the costs are automatically compared to past usage and anomalous activity is flagged. The comparison includes taking normal usage spikes into account to avoid false positives. When you get an alert, you can have confidence that there really is an anomaly.

What's Next?

Building a Culture Around EC2 Cost Optimization

In the old data center days, cost was controlled by IT operations and finance, with any new hardware purchase going through a lengthy procurement process. The cloud (and EC2) changed all that, allowing IT development and IT operations to merge into one group that could spin up and down the instances they need. Now developers could add the resources they needed to get the job done and control what they need to maintain the code — and DevOps was born.

But with the ability to control cloud resources comes the ability to control cloud costs. Developers now have the ability to optimize their cloud costs and directly control how their budget is used. It's a democratization of IT cost control that requires a change in approach just as fundamental as DevOps.

Over the last few years of cloud innovation, companies across industries have worked to figure out their own solutions to the challenge. The result is a combination of best practices, systems and culture starting to be known as FinOps.

You can find out more about FinOps on the Cloudability site. These practices are essential to optimizing your EC2 infrastructure. And when your EC2 systems are optimized, you free up valuable resources that can be used to fuel innovation and give your company the competitive differentiators it needs to succeed.

About Cloudability

Cloudability helps IT, Finance and Business teams manage the variable spend model of cloud with a FinOps platform that uses data science, machine learning and automation. With over $9 billion in cloud spend under management, we enable customers to create financial accountability and lower the unit economics of cloud.

Get the resources you need at cloudability.com/resources

About FinOps

FinOps is a combination of best practices, culture and systems that enable distributed IT, Finance and Business teams to tune cloud deployments for speed, cost or quality. The FinOps journey consists of three iterative phases — Inform, Optimize, Operate.

Learn about FinOps by reading FinOps: A New Approach to Cloud Financial Management.

Get Your Cloud Under Control

Whether you're a cloud-native company moving quickly or an enterprise looking to migrate to the cloud, there's a complex journey ahead. Get the resources to learn more about building and managing a cost-efficient cloud.

cloudability.com/resources