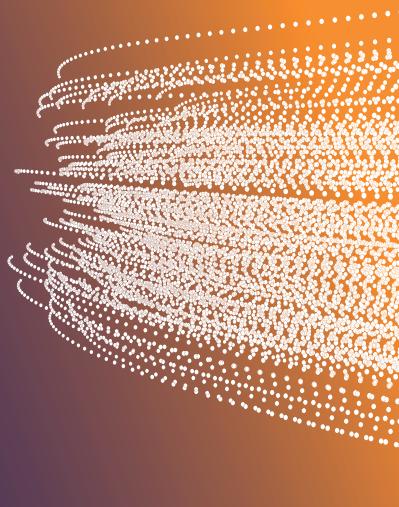


10 best practices for reducing spend in AWS



INTRODUCTION



Amazon Web Services (AWS) forever changed the world of IT when it entered the market in 2006 offering services for pennies on the dollar. While its prices have come down significantly over the years, many companies learned the hard way that moving to the public cloud didn't always achieve the cost savings they expected.

"In fact, organizations have frequently noticed public cloud bills two to three times higher than expectations"

In fact, organizations have frequently noticed public cloud bills two to three times higher than expectations⁽¹⁾. This doesn't mean that moving to the public cloud is a mistake. The public cloud provides huge benefits in agility, responsiveness, simplified operation, and improved innovation.

The mistake is assuming that migrating to the public cloud without implementing management, governance and automation will lead to cost savings. To combat rising cloud infrastructure costs, use these proven best practices for cost reduction and optimization to make sure you are getting the most out of your environment.

⁽¹⁾ Gartner, *Innovation Insight for Dynamic Optimization Technology for Infrastructure Resources and Cloud Services*, Donna Scott and Milind Govekar, 29 February 2016.

1

DELETE UNATTACHED EBS VOLUMES

It's common to see thousands of dollars in unattached Elastic Block Storage (EBS) volumes within AWS accounts. These are volumes that are costing money but aren't being used for anything. When an instance is launched, an EBS volume is usually attached to act as the local block storage for the application. When an instance is launched via the AWS Console, there is a setting that ensures the associated EBS volume is deleted upon the termination of the instance.

"By continuously checking for unattached EBS volumes in your infrastructure, you can cut thousands of dollars from your monthly AWS bill."

However, if that setting is not checked, the volume remains when an instance is terminated. Amazon will continue to charge for the full list price of the volume, despite the fact that the volume is not in use.

By continuously checking for unattached EBS volumes in your infrastructure, you can cut thousands of dollars from your monthly AWS bill. One large online gaming company reduced its EBS usage by one third by eliminating unused EBS volumes and proactively monitoring for unattached volumes.



CLOUDOLOGIST TIP

Best practices are to delete a volume when it has been unattached for two weeks, as it is unlikely the same volume will be utilized again.

2

DELETE AGED SNAPSHOTS

Many organizations use EBS snapshots to create point-in-time recovery points to use in case of data loss or disaster. However, EBS snapshot costs can quickly get out of control if not closely monitored. Individual snapshots are not costly, but the cost can grow quickly when several are provisioned.

"One B2B SaaS company found that among its millions of EBS snapshots, a large percentage of them were more than two years old, making them good candidates for deletion."

A compounding factor on this issue is that users can configure settings to automatically create subsequent snapshots on a daily basis, without scheduling older snapshots for deletion.

Organizations can help get EBS snapshots back under control by monitoring snapshot cost and usage per instance to make sure they do not spike out of control. Set a standard in your organization for how many snapshots should be retained per instance. Remember that the majority of the time, a recovery will occur from the most recent snapshot.

One B2B SaaS company found that among its millions of EBS snapshots, a large percentage of them were more than two years old, making them good candidates for deletion.

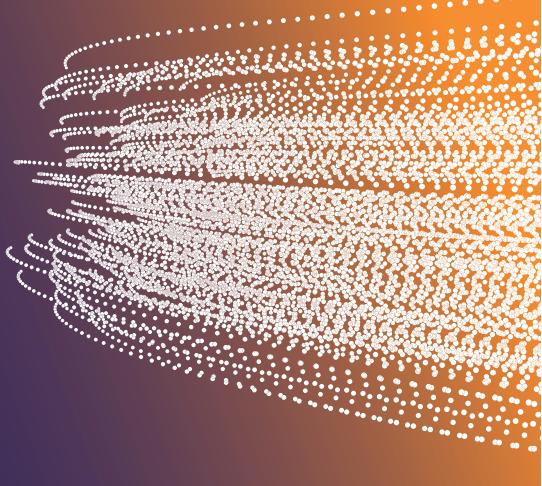


CLOUDOLOGIST TIP

One way of finding snapshots that are good candidates for deletion is to identify the snapshots that have no associated volumes. When a volume is deleted, it's common for the snapshot to remain in your environment. Be careful not to delete snapshots that are being utilized as a volume for an instance

3

DELETE DISASSOCIATED ELASTIC IP ADDRESSES



An Elastic IP address is a public IP address that can be associated with an instance and allows the instance to be reachable via the Internet. The pricing structure for an Elastic IP address is unique in that when an instance is running, the Elastic IP is free of charge..

However, if an instance is stopped or terminated and the Elastic IP address is not associated to another instance, you will be charged for the disassociated Elastic IPs. Unfortunately, it is difficult to identify and manage disassociated Elastic IPs within the AWS console.

This may or may not amount to a significant cost drive in your AWS environment, but it's key to stay on top of wasted resources and be proactive versus reactive in managing costs before they spike out of control.

From a best practice standpoint, monthly Elastic IP charges should be as close to zero as possible. If disassociated Elastic IPs are within the AWS accounts, they should either be reassigned to an instance or outright deleted in order to avoid wasted cost.

One large telecommunications company learned the hard way that small changes in its environment can lead to significant charges in Elastic IPs. In an effort to reduce their overall monthly spend, the company terminated hundreds of idle instances in one of its accounts. Company leaders forgot, however, to release the attached Elastic IP addresses. The finance department did not learn about this exorbitantly costly mistake until the following month when the AWS invoices arrived with Elastic IP charges of almost \$40,000.

4

TERMINATE ZOMBIE ASSETS

Zombie assets are infrastructure components that are running in your cloud environment but not being used for any purpose. Zombie assets come in many forms. For example, they could be EC2 once used for a particular purpose but are no longer in use and have not been turned off. Zombie EC2 instances also can occur when instances fail during the launch process or because of errors in script that fail to deprovision instances.



CLOUDOLOGIST TIP

Start your zombie hunt by identifying instances that have a Max CPU% < 5% over the past 30 days. This doesn't automatically mean this instance is a zombie, but it's worth investigating further.

Zombie assets can also come in the form of idle Elastic Load Balancers (ELB) that aren't being used effectively or an idle Relational Database Service (RDS) instance. No matter the cause, AWS will charge for them as long as these assets are in a running state.

They must be isolated, evaluated and immediately terminated if deemed nonessential. Take a snap-shot, or point-in-time copy, of the asset before terminating or stopping it to ensure you can recover it if the asset is needed again.

One customer had a nightly process to help its engineering velocity -- loading an anonymized production database into RDS to use for testing and verification in a safe environment. The process worked well and saved lots of time for engineers.

However, while the automation was good at spinning up new environments, the customer never made a plan for cleanup.

Each night a new RDS instance was spun up, with the attached resources, and then was abandoned, eventually leaving hundreds of zombie resources.

5

UPGRADE INSTANCES TO THE LATEST GENERATION

“For example, upgrading a c1.xlarge to a c3.xlarge will cut costs by up to 60% while offering significantly faster processing.”

Every few years, AWS releases the next generation of instances with improved price-per-compute performance and additional functionality like clustering, enhanced networking, and the ability to attach new types of EBS volumes. For example, upgrading a c1.xlarge to a c3.xlarge will cut costs by up to 60% while offering significantly faster processing.

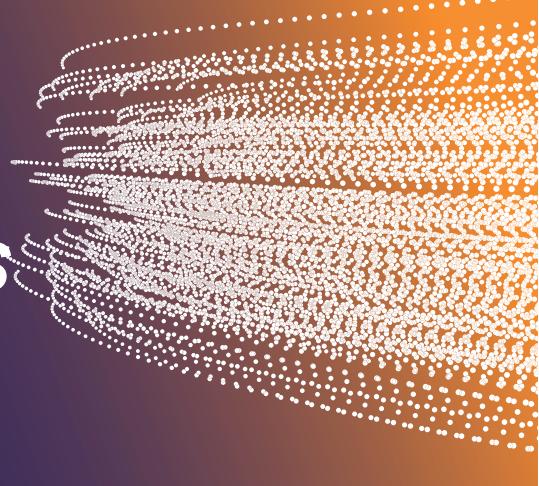
Every few years, AWS releases the next generation of instances with improved price-per-compute performance and additional functionality like clustering, enhanced networking, and the ability to attach new types of EBS volumes. For example, upgrading a c1.xlarge to a c3.xlarge will cut costs by up to 60% while offering significantly faster processing.

The migration of a cluster of instance types from first generation to second generation will likely be a gradual process for most companies. The first step is to decide which accounts have instances that are candidates for conversion. If you are heavily invested in Reserved Instances (RIs), only instances with expiring reservations or those running strictly on-demand should be converted.

One large B2B SaaS company found that almost 60% of the instance hours they ran in the past 12 months were using older-generation instance types. Analysis revealed that upgrading those instances to the latest generation would save them millions of dollars per year.

6

RIGHTSIZE EC2 INSTANCES & EBS VOLUMES



Rightsizing Elastic Compute Cloud (EC2) instances is the cost reduction initiative with the potential for the biggest impact. It's common for developers to spin up new instances that are substantially larger than necessary.

This may be intentional to give themselves extra headroom or accidental since they don't know the performance requirements of the new workload yet. Over-provisioning an EC2 instance can lead to exponentially higher costs.

Without performance monitoring or cloud management tools, it's hard to tell when assets are over- or under-provisioned.

Some information can be gathered from Cloud-Watch—it's important to consider CPU utilization, memory utilization, disk utilization, and network in/out utilization.

Reviewing these trended metrics over time, you can make decisions around reducing the size of the instance without hurting the performance of the applications on the instance. Because it's common for instances to be underutilized, you can reduce costs by assuring that all instances are the right size.

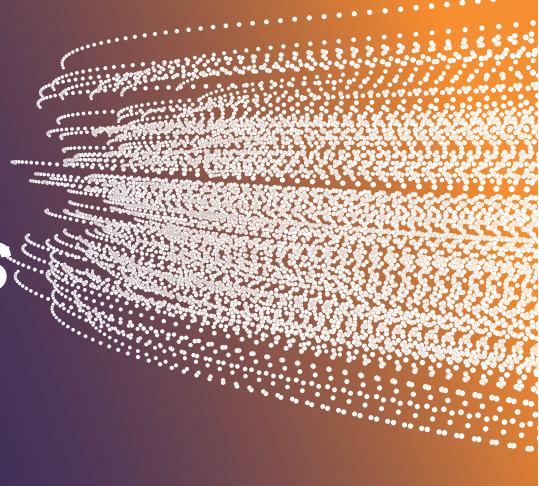


CLOUDOLOGIST TIP

A good starting place for rightsizing is to look for instances that have an Avg CPU < 5% and Max CPU < 20% for 30 days. Instances that fit this criteria are viable candidates for rightsizing or termination.

6

RIGHTSIZE EC2 INSTANCES & EBS VOLUMES



If a volume is attached to an instance and barely has any read/writes on that volume, the instance is either inactive or the volume is unnecessary. These are good candidates to flag for rightsizing evaluation.

Similar to EC2 instances, EBS volumes can also be rightsized. Instead of looking at dimension of CPU, memory, disk and network, the critical factors to consider with EBS are capacity, IOPS and throughput. As discussed earlier, removing unattached volumes is one way to reduce the cost associated with EBS volumes.

Another approach is to evaluate which volumes are over-provisioned and can be modified for potential cost savings. AWS offers several types of EBS volumes, from Cold HDDs to Provisioned IOPS SSDs, each with their own set of pricing and performance. By analyzing the read/writes on all volumes, you can find opportunities for cost savings.

If a volume is attached to an instance and barely has any read/writes on that volume, the instance is either inactive or the volume is unnecessary. These are good candidates to flag for rightsizing evaluation.

It's typical to see General Purpose SSD or Provisioned IOPS SSD volumes that barely have any read/writes for a long period of time. They can be downgraded to Throughput Optimized HDD or even Cold HDD volumes in order to reduce cost.

7

STOP AND START INSTANCES ON A SCHEDULE

As previously highlighted, AWS will bill for that instance as long as an instance is running. Inversely, if an instance is in a stopped state, there is no charge associated to that instance.

For instances that are running 24/7, Amazon will bill for 672 to 744 hours per instance, depending on the month. If an instance is turned off between 5pm and 9am on weekdays and stopped weekends and holidays, then total billable hours per month would range from 152 to 184 hours per instance, saving you 488 to 592 instance hours per month.

This is an extreme example, as flexible workweeks and global teams means that you can't just power down instances outside normal working hours. However, outside of production, you'll likely find many instances that do not need to truly run 24/7/365.

The most cost-efficient environments dynamically stop and start instances based on a set schedule. Each cluster of instances can be treated a different way. These types of lights-on/lights-off policies can often be even more cost effective than RI purchases, so it's crucial to analyze where this type of policy can be implemented.



CLOUDOLOGIST TIP

Set a target for weekly hours that non-production systems should run. One large publishing company set that target at less than 80 hours per week, which is saving them thousands of dollars a month.

8

BUY RESERVED INSTANCES ON EC2, RDS AND AUTOMATE OPTIMIZATION

"RIs can save you up to 75% compared to on-demand pricing, so they're a no-brainer for any company with sustained EC2 or RDS usage."

Purchasing Reserved Instances (RI) is an extremely effective cost saving technique, yet many organizations are overwhelmed by the number of options.

AWS Reserved Instances allow you to make a commitment to AWS to utilize specific instance types in return for a discount on your compute costs, and a capacity reservation that guarantees your ability to run an instance of this type in the future.

Reserved Instances are similar to coupons purchased either all upfront, partially upfront, or no upfront which customers can apply to running instances. RIs can save you up to 75% compared to on-demand pricing, so they're a no-brainer for any company with sustained EC2 or RDS usage.

One common misconception around RIs is that they cannot be modified.

This is not true!

Once purchased, RIs can be modified in several ways at no additional cost:

- Switching Availability Zones within the same region
- Switching between EC2 classic and Virtual Private Cloud
- Altering the instance type within the same family (this includes both splitting & merging instance types)
- Changing the account that benegits from the RI purchase

8

BUY RESERVED INSTANCES ON EC2, RDS AND AUTOMATE OPTIMIZATION

The most mature AWS customers are running more than 80% of their EC2 infrastructure covered by RI purchases. A best practice is to not let this number dip below 60% for maximum efficiency.

One consumer travel website is now running more than 90% of its EC2 instances covered by RIs, saving the company millions of dollars a year.

“It’s critical to not only purchase RIs but also continuously modify them to get the most value.”

It’s critical to not only purchase RIs but also continuously modify them to get the most value.

If a reservation is idle or underutilized, modification means the RI can cover on-demand usage to a greater degree. This ensures that the RIs are operating as efficiently as possible and that savings opportunities are being maximized.

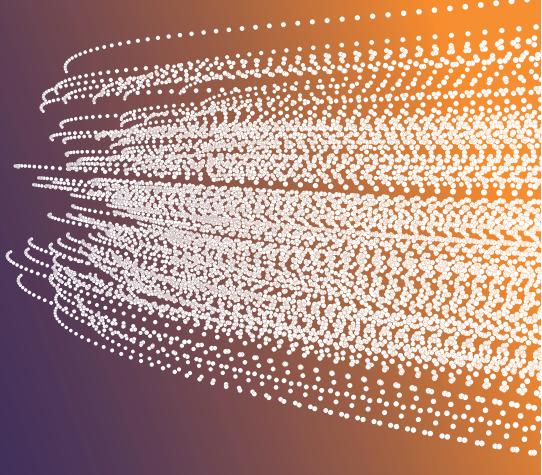


CLOUDOLOGIST TIP

A 1-year term reservation will almost always break-even after six months. This is when you can shut down an instance and still benefit from the reservation’s pricing discount. For a 3-year reservation, the break-even point usually occurs around nine months..

9

BUY RESERVED NODES ON REDSHIFT AND ELASTICACHE



EC2 and RDS aren't the only assets in AWS that use reservations. Redshift and ElastiCache are two additional services that you can buy reservations for to reduce cost.

Redshift Reserved Nodes function similarly to EC2 and RDS instances, in that they can be purchased all upfront, partially upfront, or no-upfront in 1- or 3-year terms.

ElastiCache Reserved Cache Nodes give you the option to make a low, one-time payment for each cache node you want to reserve and in turn receive a significant discount on the hourly charge for that Cache Node.

Amazon ElastiCache provides three ElastiCache Reserved Cache Node types (Light, Medium, and Heavy Utilization Reserved Cache Nodes) that enable you to balance the amount you pay upfront with your effective hourly price.

Taking advantage of Reserved Nodes can have a significant impact on your AWS bill.



CLOUDOLOGIST TIP

Reserved Nodes can save you up to 75% over on demand rates when used in steady state. One online gaming company reduced Redshift compute cost by nearly 75% by using Redshift Reserved Nodes.

10

MOVE OBJECT DATA TO LOWER COST TIERS



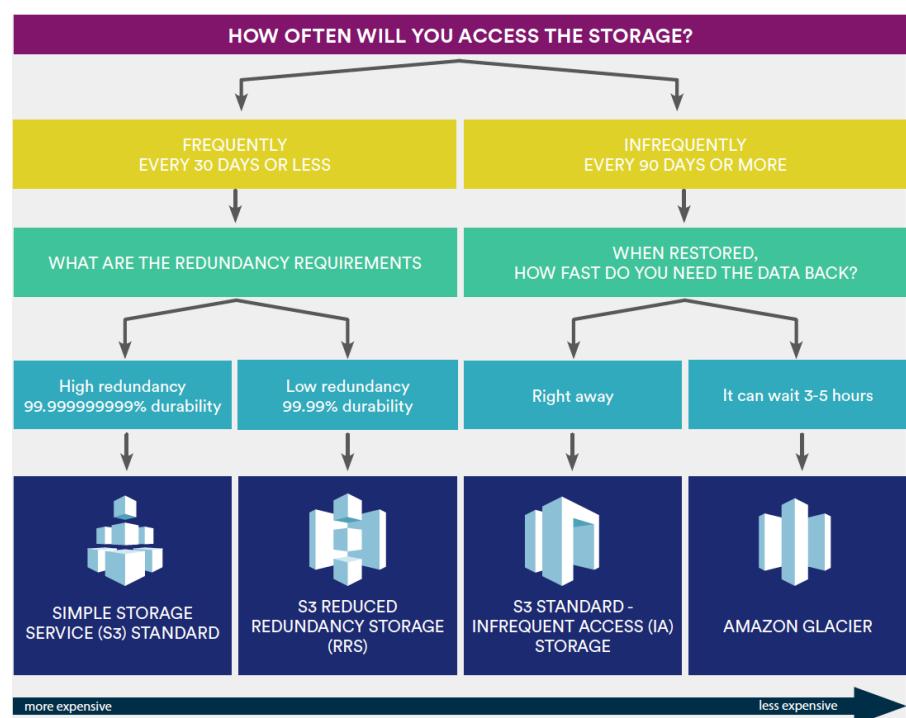
CLOUDOLOGIST TIP

Best practice is that any objects residing in S3 that are older than 30 days should be converted to S3 Infrequent Access. While standard storage class pricing is tiered based on the amount of content within the bucket, with a minimum price of \$0.0275 per GB per month, Infrequent Access storage remains consistent at \$0.0125 per GB per month.

AWS offers several tiers of object storage at different price points and performance levels. Many AWS users tend to favor S3 storage, but you can save more than 75% by migrating older data to lower tiers of storage.

The best practice is to move data between the tiers of storage depending on its usage. For example, Infrequent Access Storage is ideal for long-term storage, backups and disaster recovery content, while Glacier is best suited for archival.

In addition, the infrequent access storage class is set at the object level and can exist in the same bucket as standard. The conversion is as simple as editing the properties of the content within the bucket or creating a lifecycle conversion policy to automatically transition S3 objects between storage classes. Here is a quick overview of the current object storage offerings from AWS:



CONCLUSION

It's important to remember that these best practices are not meant to be one-time activities, but ongoing processes. Because of the dynamic and ever changing nature of the cloud, cost optimization activities should ideally take place continuously.

Learn more about how CloudHealth Technologies can help you automate the continuous optimization of your cloud environment by visiting www.cloudhealthtech.com.