N2WS

7 Steps to Save Big on Your AWS Bill

A Practical Guide to Cloud Cost Optimization

www.n2ws.com

Overview

The public cloud offers many advantages over the traditional model of IT – through low-cost infrastructure, fast provisioning and access to cutting-edge cloud-based technologies.

But only if you adapt to the new on-demand pay-as-you-go model of the cloud and learn how to manage your resources efficiently.

Without an effective cost-optimization strategy, you can end up wasting money on unused and underutilized cloud resources.

You can miss out on opportunities for significant discounts on your monthly cloud bills. And you may not choose the most cost-efficient solution for your applications.

This paper provides you with a strategy for reducing your costs on leading cloud platform AWS. But it also serves as a practical guide to cloud cost optimization – with diagrams, screenshots and links to additional resources, so you can start taking cost-saving measures straight away.

Let's start by looking at the optimization tools you'll need for the job.

Table of Contents

Cost-Optimization Toolkit

Monthly AWS Bill

Your monthly billing report is key to understanding what's driving spending.

Use it to identify the services that are costing you the most, as these are the low-hanging fruit where you can make the biggest savings.
You can also use it to check whether you're making good use of any Reserved Instances (RIs) – by comparing the total usage available with the total number of hours you actually consume.

What's more, you can track spending and monitor the progress of your cost-optimization regime.
In the case of consolidated bills, line items are broken down by account, product and region. Bills for individual accounts are simply organized by product and region.

AWS CloudWatch

An infrastructure monitoring service that provides insights for managing your AWS resources.

You can use it to visualize log data from your EC2 deployments. You can monitor estimated charges for individual AWS services and set up email alerts that notify you when costs exceed a certain threshold. You can also create alarms and take automated actions whenever resource utilization reaches predefined thresholds.

CloudWatch provides CPU utilization data for EC2 instances by default. But you can also set up custom metrics to monitor memory utilization.

AWS Cost Explorer

A cost analysis and forecasting tool, which helps you make sense of your costs across a complex array of resources, providing high-level visualizations of spending patterns over periods of up to 12 months.

Cost Explorer includes a suite of default reports that help you quickly compare costs between different products, accounts or regions.

In addition, a Reserved Instance (RI) reporting feature shows you how well you're utilizing your RIs and how much of your overall instance usage is covered by them.

You can also dig deeper by filtering and grouping your data as well as creating customized reports.

AWS Trusted Advisor

An AWS online management tool that makes a series of **best-practice checks** aimed at helping you reduce cost and improve system performance, reliability and security.

The cost-optimization category includes checks and alerts for:

- Underutilized EC2 and RDS instances
- EBS volumes with little or no write activity
- Idle load balancers
- Unused Elastic IP addresses

It also makes RI recommendations based on your last month's instance usage patterns and alerts you to RIs that have recently expired or due to expire in the next 30 days.

EC2 Right Sizing

An implementation recommended by AWS that provides detailed recommendations for right-sizing your Amazon EC2 instances base on utilization over a two-week period.

The solution is based on a CloudFormation template, which launches a series of components that form the application stack. Setup requires technical knowledge of working on the AWS platform.

Amazon S3 Analytics – Storage Class Analysis

An S3 storage optimization tool, which helps you see how much of your data you access as it ages over time.

It filters your data into a series of predefined age groups, starting from objects less than 15 days old through to objects 730 days and older, and presents a visualization for each group showing how much data you stored during that time period and how much of it you retrieved.

Storage Class Analysis is intended as a visual aid for making decisions on when to transition objects from S3 Standard to the lower-cost S3 Standard – IA tier.

Amazon QuickSight

QuickSight lets you dig deeper into your S3 access patterns by providing more detailed graphical insights that you can use as a basis for setting your S3 lifecycle policies.

It includes a number of default visualizations, which provide side-by-side comparisons of storage consumption and retrieval across each of the different predefined age groups.

You can switch between different visualization formats, such as multi-line graphs, pie charts and bar charts, and also create your own visualizations.

AWS Instance Scheduler

An AWS implementation, based on a CloudFormation template, for setting custom stop-and-start schedules on your EC2 and RDS instances.

The solution helps you reduce your monthly bills by only running instances during the periods you actually need them. For example, Dev/Test resources are rarely used outside regular 9–5 hours.

So you can define a schedule that closes them down in the evenings, at weekends and during public holidays. Setup is relatively simple but requires some technical knowledge of working on the AWS platform.

AWS Lambda

A fully managed event-driven serverless service, which you can use to deploy code that runs in response to CloudWatch events.

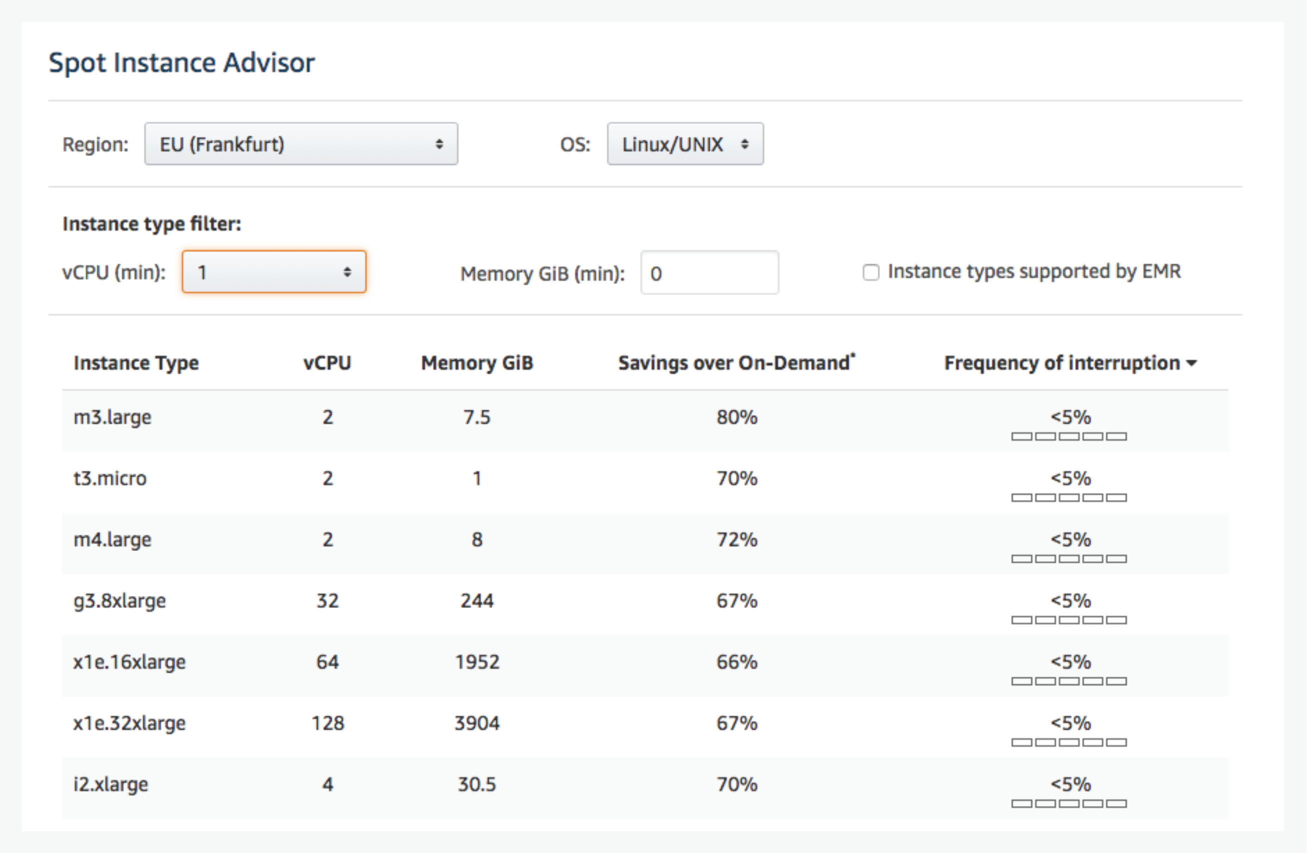
This gives you the scope to automate some of your cost-optimization processes. For example, you could set up a Lambda function to automatically find and delete unused EBS volumes or delete snapshots when they reach the end of a specified retention period.

Lambda is also a component of the automated **AWS Instance Scheduler** deployment.

Spot Instance Advisor

A simple online reference tool for weighing up the likelihood of a Spot Instance interruption against the savings you can make over the equivalent on-demand instance.

It is particularly useful if you're relatively new to Spot Instances, as it can help you identify low-risk instance types for applications that don't yet have sufficient flexibility or fault tolerance to cope with higher levels of interruption.



Spot Instance Advisor

Region: EU (Frankfurt) OS: Linux/UNIX

Instance type filter:

vCPU (min): 1 Memory GiB (min): 0 □ Instance types supported by EMR

Instance Type	vCPU	Memory GiB	Savings over On-Demand*	Frequency of interruption
m3.large	2	7.5	80%	<5%
t3.micro	2	1	70%	<5%
m4.large	2	8	72%	<5%
g3.8xlarge	32	244	67%	<5%
x1e.16xlarge	64	1952	66%	<5%
x1e.32xlarge	128	3904	67%	<5%
i2.xlarge	4	30.5	70%	<5%

Tag Editor

Tags facilitate visibility over your infrastructure by helping you quickly identify who is responsible for each of your resources what they use them for. They also play an important role in lifecycle management.

You can tag resources as and when you create them. You can also manage tags for an individual resource within the associated AWS console for that resource.

But if you need to centrally manage a number of tags at any one time, it's easier to use the AWS Tag Editor. With the Tag Editor, you can search through your inventory of taggable resources and then change, remove or apply new tags.

AWS Command Line Interface (CLI)

The AWS CLI gives you the power to directly manage all your AWS infrastructure from a single unified interface.

You can perform many of your cost-optimization tasks quickly and efficiently by running a single command or script. For example, you can **view Spot Instance pricing** history, **scale elastic EBS volumes** and **delete old EBS snapshots** without leaving the same terminal session. Although the AWS CLI is a very powerful tool, it takes a lot of time and learning to reach a useful level of proficiency.

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdefg
```

Example of a **delete-snapshot** command in the AWS CLI

Third-Party Tools

In addition to Amazon's own set of cost-management solutions, you should also consider third-party tools, which typically offer additional functionality, such as enhanced best-practice checks, multi-cloud capabilities and higher levels of automation.

These include conventional cost-optimization tools, which provide features such as right-sizing, RI recommendations and budget reporting.

But other types of third-party software can also help you reduce your monthly AWS bills. For example, data management tools can help you take advantage of the most cost-efficient storage methods by giving you the flexibility and control to move data around your AWS environment. Moreover, a solid tool will allow you to stop and start Amazon EC2 instances at predefined times to reduce the unnecessary costs of idle servers.

STEP 01

Nurture Cost Awareness

"The ability to scale up and down on demand has allowed resource procurement to transition from being solely owned by the finance team to being the responsibility of stakeholders across IT, engineering, finance, and beyond. This democratization of resource procurement has initiated an ever-growing group of cost-conscious stakeholders."

- Amazon Web Services, March 2018

Nurture Cost Awareness

Whatever public cloud provider you use, nurturing cost awareness should be one of you top cloud priorities.

With the on-demand utility model of the cloud, it's easy to provision computing resources. But, at the same time, it's just as easy to leave them running when you no longer need them.

Not only that, but you no longer need large upfront investments to fund new infrastructure. As a result, where IT spending was previously the sole responsibility of central finance, it's now devolved to a much broader spectrum of departments.

This calls for a change in culture towards a far more cost-aware mindset. It requires governance over and accountability for cloud usage. But it also requires a cost-aware approach to application architecture.

Governance and Accountability

You can meet your governance and accountability objectives by:

- Enforcing tagging policies so everyone knows what each resource is used for.
- Accurately allocating costs to individual business units.
- Regularly reviewing your AWS costs and the success of your cloud transformation.

Cost-Aware Architecture

Your application design should take advantage of the most cost-efficient AWS services wherever possible.

You should also make full use of the cloud's elastic features by breaking your applications up into smaller, individual, loosely coupled components.

This will help you scale your applications more efficiently, as you can increase and decrease the capacity allocated to each component independently, providing finer grained matching of discrete application functions to resource requirements.

Similarly, you can deploy different components to AWS instances in different families. That way, you can achieve more granular control by matching each component to an instance type with just the right mix of CPU and memory.

What's more, you're better able to scale your applications horizontally. This is the method whereby you deploy each component to a cluster of virtual machines (VMs) and scale in or out by removing or adding machines. It is a cost-efficient way to scale your infrastructure because you can use a network of smaller instances and adjust it by smaller, more precise increments.

STEP 02

Ditch Unused Resources

Now we're onto the cost-saving measures that can make an immediate impact on your AWS bill. The best place to start is to clean up all unused resources, as these eat away at your budgets without serving any purpose.

The following core AWS services should be the first on your agenda:

EC2 Instances

Idle EC2 instances are amongst the biggest causes of inflated AWS bills.

But you can prevent unnecessary waste by using CloudWatch alarms to automatically stop or terminate instances that have been idle for longer than a specified period. You can only terminate machines backed by Instance Store. But you can either stop or terminate instances backed by EBS – depending on whether you're likely to need them again.

For example, you may want to stop instances where you only need them during normal working hours. However, you may want to terminate others altogether if they remain unused for a longer period of time.

You can set up CloudWatch alarms in the:

- EC2 dashboard in the AWS Management console
- CloudWatch console
- AWS CLI by running the put-metric-alarm command

This gives you the scope to automate some of your cost-optimization processes. For example, you could set up a Lambda function to automatically find and delete unused EBS volumes or delete snapshots when they reach the end of a specified retention period. Lambda is also a component of the automated AWS Instance Scheduler deployment.

EBS Volumes

By default, when you terminate an EBS-backed instance, only the EBS volumes that were attached at launch are deleted. Any additional volumes you attached after you launched the instance continue to run.

This can cost you big if you leave unattached EBS volumes running for longer than necessary – especially where you use SSD-backed volumes, which cost more than twice as much as their HDD counterparts.

You can manually find and remove unattached EBS volumes in the:

- EC2 dashboard in the AWS Management console
- AWS CLI by running the describe-volumes and delete-volume commands

Don't forget to take a snapshot of your volume before you terminate it. This gives you the option to recover it should you need it again in the future. You can take snapshots in the EC2 dashboard from the same menu you use to delete volumes. ⚠

Orphan Snapshots

Just as any additional EBS volumes remain after you terminate an instance, so do your snapshots.

Although snapshots are incremental, your first backup captures the entire volume. What's more, if your retention period is long, you may need to store as much incremental data as the initial snapshot.

In other words, your snapshots may be costing you far more than you think. So, unless you want to use your orphaned snapshots to create new EBS volumes, you should delete them as part of your cost-optimization routine. You can manually find and remove orphan snapshots in the:

- AWS CLI by running the describe-snapshots and delete-snapshot commands

Idle Elastic Load Balancers (ELBs)

You should periodically check the status of each load balancer to ensure it's actively serving registered instances. If it isn't then either register instances to it or delete it. You can manually find and remove idle load balancers in the:

- EC2 dashboard in the AWS Management console
- AWS CLI by running the describe-load-balancers and delete-load-balancer commands

Elastic IPs

Public Internet addresses are a limited resource. So, to discourage waste, AWS restricts each account to five Elastic IP addresses per region.

It also levies a small hourly charge on addresses that aren't associated with a running instance. The same goes for Elastic IPs associated with a stopped instance or unattached network interface. To keep costs down, you should release Elastic IPs if you no longer need them. You can manually find and release unattached Elastic IPs in the:

- VPC dashboard of your AWS Management console
- AWS CLI by running the describe-addresses and release-address commands

Tools to Consider:

Trusted Advisor
CloudWatch
AWS Lambda

Where to Learn More:

Create Alarms to Stop, Terminate, Reboot or Recover an Instance
Terminating Your EC2 Instance
Trusted Advisor Cost-Optimization Best Practice Checks
AWS Elastic IP Addresses
Deleting an Amazon EBS Snapshot

STEP 03

Fine-Tune Your Compute Resources

Size matters when it comes to cloud cost efficiency. If your instances provide far more CPU and memory than your workloads need then you're simply pouring money down the drain.

But if they don't provide enough then your application performance will suffer.

Fine-Tune Your Compute Resources

Size matters when it comes to cloud cost efficiency.

What capacity do you need to meet average consumption? And how much additional power do you need to cope with peaks in demand?

The answers to these questions will help you determine the optimum utilization level of your instances. But, as a rule of thumb, you should generally aim for 70–80% utilization if you want to minimize costs without affecting latency.

But right-sizing your instances shouldn't end there. You can fine-tune your resources by switching to instances with different performance characteristics.

The following table highlights common examples where a different instance family may be a better fit for your application:

Use Case	Switch to
Compute-intensive workloads requiring a higher proportion of CPU	**Compute-optimized** family of instances (C5 and C4)
Memory-intensive workloads requiring a higher proportion of memory	**Memory-optimized** family of instances (R5, R5a, R4, X1e, X1, High Memory and z1d)
Disk-intensive workloads requiring locally attached storage	**Storage-optimized** family of instances (H1, I3 and D2)
Small and moderate workloads requiring occasional additional capacity to meet peaks in demand	**Burstable Performance** family of instances (T2, T3 and T3a)

You should also collect and analyze usage statistics on a regular basis and review your EC2 capacity over time. And, where possible, you should deploy your application to a fleet of smaller instances.

So, instead of scaling vertically by changing the size of your instance, you can scale your infrastructure horizontally a little at a time by adding or removing lower-cost machines.

T3 Burstable (General Purpose) Instance Pricing

US-East (N. Virginia)

Name	vCPUs	Baseline Performance / vCPU	Memory	Price / Hour (Linux)	Price / Hour (Windows)
t3.nano	2	5%	0.5 GiB	$0.0052	$0.0098
t3.micro	2	10%	1 GiB	$0.0104	$0.0196
t3.small	2	20%	2 GiB	$0.0209	$0.0393
t3.medium	2	20%	4 GiB	$0.0418	$0.0602
t3.large	2	30%	8 GiB	$0.0835	$0.1111
t3.xlarge	4	40%	16 GiB	$0.1670	$0.2406
t3.2xlarge	8	40%	32 GiB	$0.3341	$0.4813

You can access statistics for CPU and memory consumption of a specific instance in the:

- CloudWatch console
- AWS CLI by running the get-metric-statistics command

If you're an Amazon RDS user, you should be aware that DB instances are decoupled from storage. So, from time to time, you may need to switch to a different DB instance type to meet changing demand on your database system. ⚠

Tools to Consider:

Cost Explorer,

Trusted Advisor,

EC2 Right Sizing,

CloudWatch

Where to Learn More:

Amazon EC2 Instance Types

EC2 Pricing

Cost Optimization: EC2 Right Sizing

EC2 Right Sizing: AWS Implementation Guide

Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

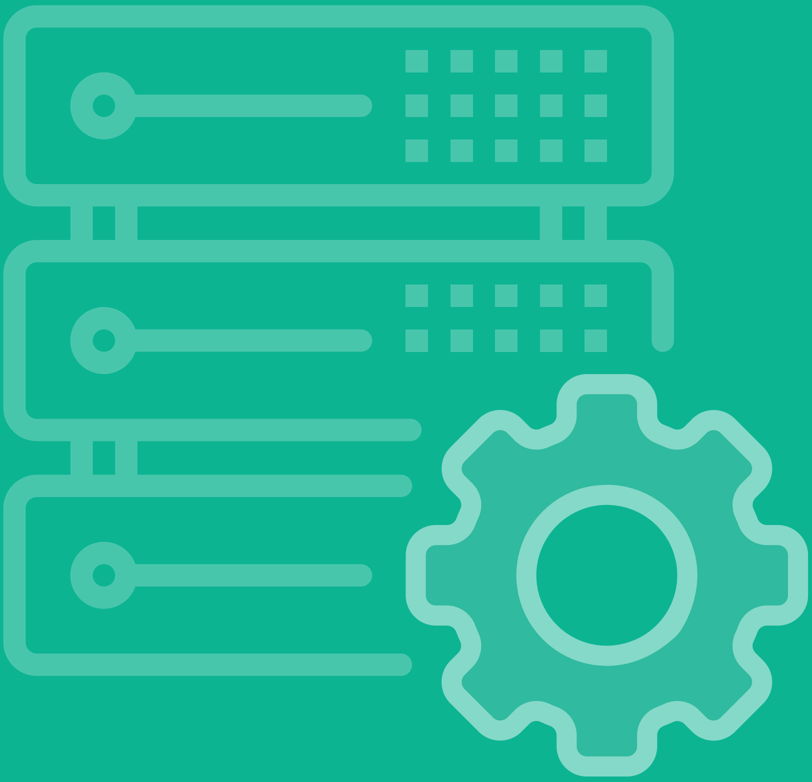
Tagging Your Amazon EC2 Resources

STEP 04

Optimize Your Storage

“Organizations tend to think of data storage as an ancillary service and do not optimize storage after data is moved to the cloud. Many also fail to clean up unused storage and let these services run for days, weeks, and even months at significant cost.”

- Amazon Web Services, March 2018



EBS Volumes

In February 2017, AWS changed the way EBS works, making it possible to adjust the size of current-generation volumes without detaching them first.

This now makes it much easier to optimize your block storage capacity and manage costs. To monitor the disk utilization of your EBS volumes, you'll need to publish custom usage metrics to CloudWatch. You can modify the size of an EBS volume in the:

- EC2 dashboard in the AWS Management console
- AWS CLI by running the modify-volume command

EBS Snapshots

In addition to removing orphan snapshots, you should maintain a snapshot retention policy that properly balances your data backup requirements with cost.

Deleting an old snapshot won't always necessarily reduce your storage overhead. This is because it may contain referenced data that's required to restore the volume. If this is the case then the data is preserved and continues to rack up charges.

However, if a good proportion of the data on your volumes changes frequently then deleting old snapshots will remove duplicate files and help to keep your storage costs down.

AWS compresses your snapshots and stores them in S3. So your snapshot costs may seem relatively inexpensive. But the vendor uses a different pricing system from its regular S3 tiers. This is more than twice the Standard S3 rate.

Not only that, but the buckets AWS uses to store snapshots are not publicly visible. So you cannot retrieve them using the standard S3 methods.

Currently, the only way to retain snapshots at a lower cost is to use a third-party cloud data management tool that's able to copy them to a standard S3 bucket.

Amazon S3

Amazon's object storage offering S3 is the cloud provider's most cost-efficient way to store data. So, to keep your bills down, you should make use of it wherever possible.

This means limiting your use of more expensive services, such as EBS and EFS, to only those use cases where they're strictly necessary. S3 is not only the least expensive storage solution, but it also automatically scales with you. So you don't have to worry about wasting unused provisioned storage.

The service is available in three main storage classes aimed at different use cases:

- **Standard**

  Low-latency storage option with 99.999999999% durability and 99.99% availability. Designed for frequently accessed data.

- **Standard – IA**

  The same low latency and durability as S3 Standard but availability slightly lower at 99.9%. Standard – IA is designed for infrequently accessed data, as storage charges are lower but data retrieval charges higher.

  Objects can exist in the same bucket as those set as S3 Standard. This means you can transition data between classes without making application changes.

- **Glacier**

  The most economical storage option. However, data retrieval is slow and expensive, making it suitable only for data archiving and long-term backups.

Example Storage Pricing by Class

US-East (N. Virginia)

	S3 Standard	S3 Standard - IA	Glacier
First 50 TB/month	$0.023 per GB	$0.0125 per GB	$0.004 per GB
First 450 TB/month	$0.022 per GB	$0.0125 per GB	$0.004 per GB
First 500 TB/month	$0.021 per GB	$0.0125 per GB	$0.004 per GB
Charges for data/requests/retrieval also apply			

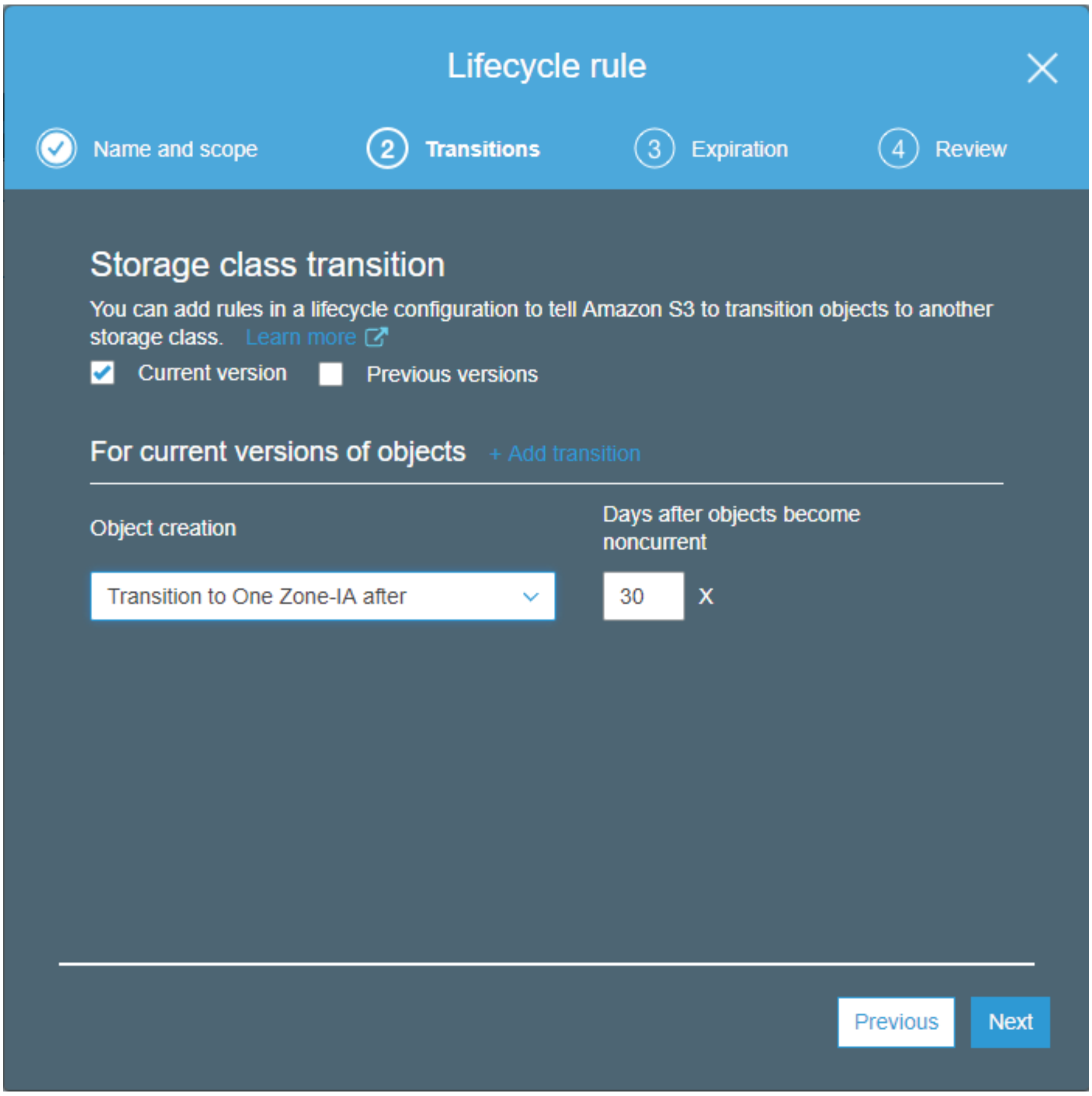
Efficient lifecycle management of your S3 objects is key to lowering your storage costs. So AWS provides a number of tools to help you manage your buckets and take advantage of its different storage classes.

Object Tagging

Tags help you better understand your S3 bucket usage. They can be used to filter objects both as part of your lifecycle analysis and for configuring your lifecycle policies.

You can tag an S3 object in the:

- S3 dashboard in the AWS Management console
- Tag Editor
- AWS CLI by running the put-object-tagging command



Lifecycle rule

✓ Name and scope 2 Transitions 3 Expiration 4 Review

Storage class transition

You can add rules in a lifecycle configuration to tell Amazon S3 to transition objects to another storage class. Learn more

☑ Current version ☐ Previous versions

For current versions of objects + Add transition

Object creation	Days after objects become noncurrent
Transition to One Zone-IA after	30

Setting up a lifecycle rule in the S3 console

Amazon S3 Analytics – Storage Class Analysis

Before you configure your lifecycle policies, you'll need to understand the patterns of access to your data over time.

Storage Class Analysis is a simple data visualization tool that can play a useful role in your S3 optimization workflow. Although it provides a basic overview of retrieval patterns, it may be all you need to decide at what age your data is ready for transition between tiers.

You can configure **Storage Class Analysis** in the:

- S3 dashboard in the AWS Management console

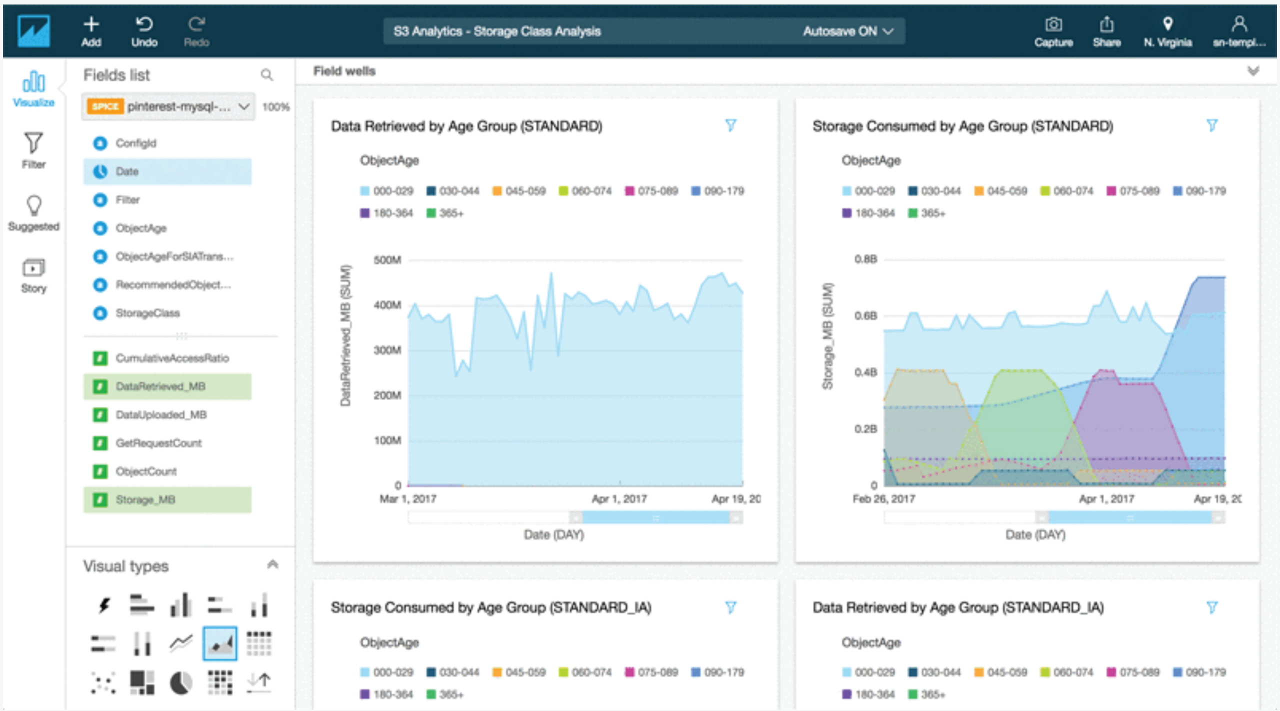
Typical dashboard output from Storage Class Analysis

Amazon QuickSight

Amazon QuickSight is a business intelligence (BI) and data visualization service with a pre-built integration for S3 Storage Analytics.

It provides more detailed views of your storage retrieval patterns for fine-tuning your S3 lifecycle policies. If you have Storage Class Analysis enabled, you can access Amazon QuickSight via the:

- QuickSight link in the S3 dashboard



The QuickSight dashboard

Tools to Consider:

Tag Editor

Amazon S3 Analytics – Storage Class Analysis

Amazon QuickSight

Where to Learn More:

Amazon EBS Update – New Elastic Volumes Change Everything

Amazon EBS Pricing – Snapshots to Amazon S3

Amazon S3 Pricing

Amazon S3 Storage Classes

S3 Lifecycle: Supported Transitions and Related Constraints

How to Set Up and Use AWS Storage Class Analysis

Amazon QuickSight

Setting Up Amazon QuickSight

STEP 05

Use Auto Scaling

EC2 Auto Scaling is a horizontal scaling feature, which automatically spins up new instances as the load on your application increases and terminates them when demand goes down.

Use Auto Scaling

You can adjust the size of your cluster manually, based on a schedule or dynamically based on a target value for a specific metric, such as CPU utilization.

Dynamic scaling is a particularly effective cost-optimization technique, as it can take almost immediate action to changes in demand. By contrast, it can take time to manually respond to scaling events – while, in the meantime, your application can suffer performance issues or rack up unnecessary costs.

Auto Scaling Groups

A cluster of EC2 instances is known as an Auto Scaling group, where each group is defined by the following three key components:

- **Minimum size**

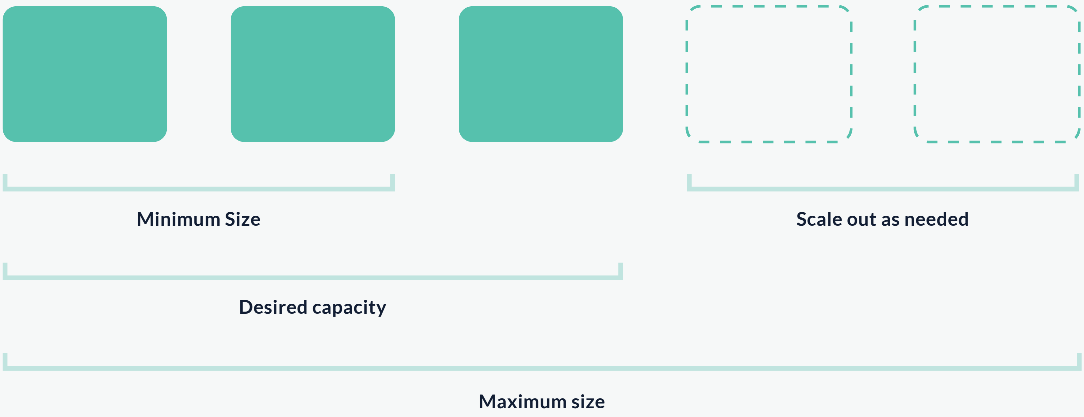
  The baseline number of instances below which the size of your group should never fall.

- **Desired capacity**

  The number of instances you want to run based on your scaling policy. Your desired capacity moves up and down dynamically as the load on your application changes. Desired capacity adjustments are triggered by CloudWatch alarms.

- **Maximum size**

  The ceiling figure above which the number of instances in your group should never exceed.



A simple Auto Scaling group

Before you create an EC2 Auto Scaling group, you'll first need a launch configuration, which is a template for the instances your group will use.

You can set up an **EC2 Auto Scaling** group in the:

- EC2 dashboard in the AWS Management console
- AWS CLI by running the create-auto-scaling-group command

Top Tip

Auto Scaling is also available for other AWS services, including ECS, EC2 Spot Fleets, Aurora, DynamoDB and EMR.

You can configure all of these resources using AWS Auto Scaling – a single unified interface that simplifies Auto Scaling management across your services without having to navigate to other consoles.

Tools to Consider:

CloudWatch

ELB

Where to Learn More:

What Is AWS Auto Scaling?

Benefits of Auto Scaling

Creating an Auto Scaling Group Using a Launch Configuration

New AWS Auto Scaling – Unified Scaling for Your Cloud Applications

Using Elastic Load Balancing Health Checks with Auto Scaling

STEP 06

Adopt an RI Strategy

Reserved Instances are an alternative billing mechanism to the on-demand pricing model, where you purchase a discounted usage credit for a specific type of instance over a fixed term.

Adopt an RI Strategy

They can deliver savings of up to 75% compared with standard on-demand charges, where the maximum discount you can expect depends on the following:

- ### RI specifications

  The instance type, operating system (OS), tenancy and Availability Zone (AZ).

- ### RI term

  Either one or three years

- ### Payment option

  All upfront, partial upfront or no upfront payment at all.

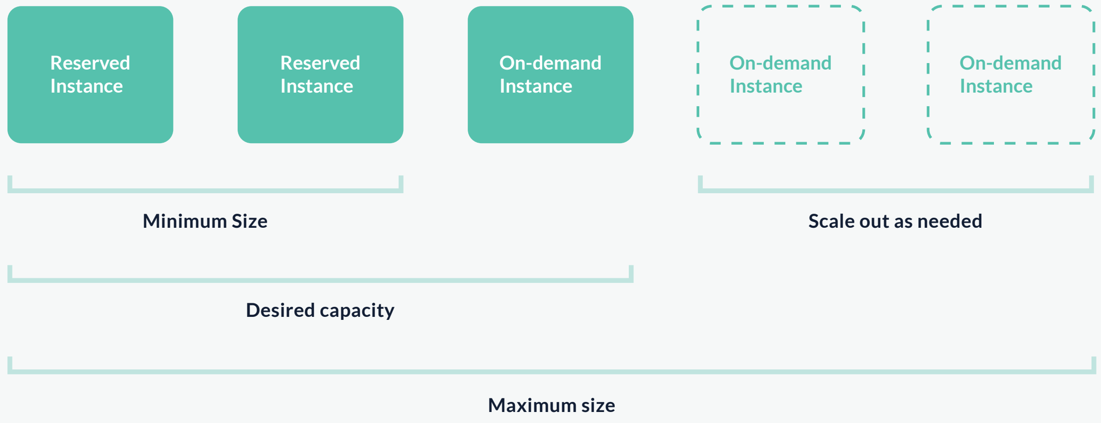
- ### Whether you purchase a Standard RI or Convertible RI

  Standard RIs offer higher discounts but limited flexibility to modify your instance specifications. Convertible RIs promise lower discounts but give you the option to exchange your RI for another in a different instance family or with a different OS, tenancy or payment option.

For maximum return on your RI purchases, your running instances should consume your usage credits as much of the time as possible. This is because, during any period where you have no active matching instances, your credit will remain unused and go to waste. This is why RIs tend to suit steady predictable workloads.

RIs are a more effective cost-optimization strategy when you use consolidated billing. This is because an RI credit can apply to a matching instance in any of a company's linked accounts. However, the account that buys the reservation is always first in line to the discount.

Another strategy for maximizing RI savings is to use them to cover the baseline capacity of an Auto Scaling group.



Use RIs to cover the capacity required for your minimum group size

As your Auto Scaling group will always run at or above your minimum number of instances, you can effectively guarantee full utilization of your matching reservations.

Other AWS services with reservation models include **RDS, Redshift, ElastiCache and DynamoDB.**

You can purchase EC2 Reserved Instances in the:

- EC2 dashboard in the AWS Management console
- AWS CLI by running the purchase-reserved-instances-offering command

Top Tip

Don't forget to monitor your RIs so you can allocate other workloads to them if they're unused or underutilized. And keep track of their expiry date, as AWS doesn't renew them automatically.

Tools to Consider:

Tag Editor
Cost Explorer
Trusted Advisor

Where to Learn More:

Amazon EC2 Reserved Instance Pricing
AWS Reserved Instance (RI) Reporting
Modifying Reserved Instances
Understanding Consolidated Bills
AWS Consolidated Billing: How to Allocate EC2 Reserved Instance Costs

STEP 07

Purchase Spot Instances

Spot Instances are a disposable class of instances, which are priced based on supply of and demand for EC2 spare capacity.

Purchase Spot instances

Using Spot Instances can potentially bring the biggest savings to your AWS monthly bill, with discounts of up to 90% compared with standard on-demand machines.

When you purchase a Spot Instance, you're simply charged the current market price for the machine type you need – subject to an optional maximum price you're willing to pay.

However, AWS can interrupt your Spot Instance if it doesn't have enough capacity to meet demand. It can also do so if the current Spot price is greater than your maximum price or your request includes a constraint, such as a launch group or an AZ group, it cannot meet.

Although interruptions are relatively infrequent, they can still happen at any time. So Spot Instances aren't suitable for mission-critical applications that rely on continuous availability.

Interruption Behavior

You can specify how EC2 interrupts your Spot Instances.

The default interruption behavior is to terminate them. But you also have the option to either stop or hibernate them.

When the Spot service stops or hibernates your instances, it preserves your EBS volumes. However, hibernation also preserves instance memory on the root volume. This means your workloads can resume automatically when your Spot capacity becomes available again.

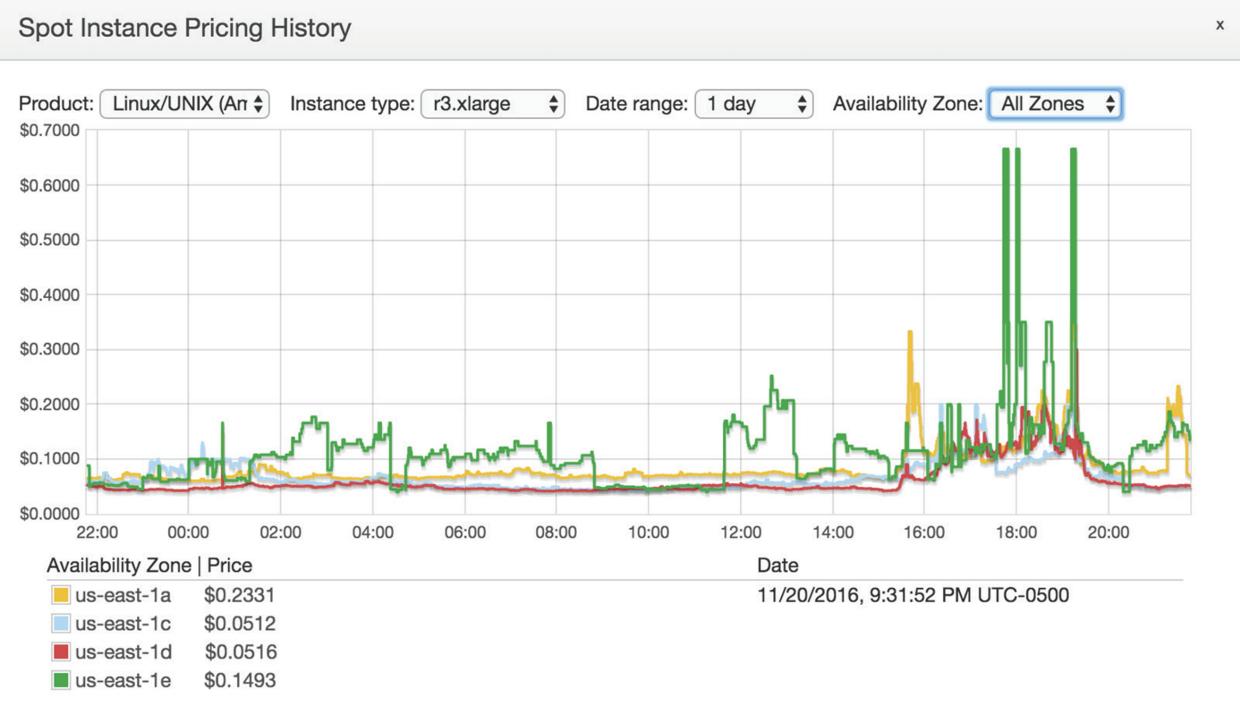
Currently, hibernation is supported by a limited number of instance types. It also requires installation of the EC2 Hibernation Agent on your guest OS.

You can make a Spot Instance request in the:

- EC2 dashboard in the AWS Management console
- AWS CLI by running the request-spot-instances command

More Predictable Spot Instance Pricing

In 2017, AWS launched a new Spot Instance pricing model that replaced the previous bidding mechanism with the current simpler system. This reduced both fluctuations in prices and the number of interruptions.



Product: Linux/UNIX (An ⬍ Instance type: r3.xlarge ⬍ Date range: 1 day ⬍ Availability Zone: All Zones ⬍

Availability Zone | Price		Date
🟨 us-east-1a	$0.2331	11/20/2016, 9:31:52 PM UTC-0500
🟦 us-east-1c	$0.0512	
🟥 us-east-1d	$0.0516	
🟩 us-east-1e	$0.1493	

Spot Instance prices are less volatile under the new pricing model

Usage Pattern	Cost-Saving Measure	Maximum Potential Savings
Steady state	Reserved Instances	75%
Variable but predictable workload	Auto Scaling	Subject to use case
Dev/Test environment	Instance Scheduler	70%
Temporary deployment	Spot Instances	90%

At-a-glance optimization measures based on usage patterns

Tools to Consider:

Spot Instance Advisor

Where to Learn More:

Amazon EC2 Spot Instances

Amazon's New EC2 Spot Pricing Model

Spot Instance Advisor

How to View Spot Instance Pricing History

Spot Instance Requests

Managing Spot Interruptions

How Spot Fleet Works

> *"Over a period of six years, the average total cost per minute of an unplanned outage has risen by 58% – from $5,617 per minute in 2010 to $8,851 per minute in 2016."*
>
> Ponemon Institute Cost of Data Center Outages report, January 2016

Avoid the Rising Cost of Downtime

A carefully coordinated cost-optimization strategy is essential to the efficient running of your AWS resources. But it's important you never lose sight of the bigger picture.

Downtime can end up being a far greater cost to your business – through reputational damage, loss of productivity and loss of revenue. So it's equally essential to have backup and recovery measures in place to help you maintain business continuity and keep downtime costs down.

Third-party data management tools are designed to simplify the backup and recovery process, so you're not scrambling for the AWS documentation whenever you need to restore your cloud infrastructure.

But data management tools don't just help in potential disaster situations. They also help you reduce your AWS bills by providing continuous cloud storage optimization.

N2WS Backup & Recovery

N2WS

Built for scale. Built for simplicity. Built for AWS.

As more companies move to the public cloud, they need a way to ensure their critical data is always available. The first step is finding a solution that provides easy, automated backups and a giant instant recovery button to protect your data from any outage any time.

The next step, after automation, is optimization —making sure your environment is set up for cost-effectiveness (and compliance). We help you on both counts.

Top Rated Backup & Recovery solution in AWS Marketplace

Automated Backup + Instant Recovery = No Downtime + No Worries

- Automate backups and get 1-click recovery for Amazon EC2 instances and EBS volumes
- Recover only what you need in 30 seconds (from individual files to full volumes or instances)

Your Database Protection Plan: Always Available, Application-Consistent

- Capture the most complete database picture with application-consistent backups for Amazon RDS databases engines: Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, MSQL
- Enjoy complete support for NoSQL databases: Cassandra, DynamoDB, Mongo DB

Savvy Data Lifecycle Management to Save Your AWS (and Your Budget)

- Save up to 60% on long-term retention costs by decoupling and transferring your EBS snapshots to the N2WS S3 repository, with the ability to recover to any region or account
- Start, stop and hibernate groups of Amazon EC2 or Amazon RDS instances and save on computing costs with N2WS LightSwitch

About N2WS

N2WS Backup & Recovery is a cloud-native data protection solution built specifically for the AWS platform. Our solution gives you the flexibility and control to move data around your AWS environment, providing backup and disaster recovery functionality across regions and accounts.

Get Free AWS Backup ›

+1 888 426 4329 • info@n2ws.com • www.n2ws.com