Cloud Cost Optimization Guide

Compute, Storage & Database

Check out this checklist of technical recommendations that can be made to your **AWS cloud infrastructure** to launch or improve your organization's cost optimization processes.

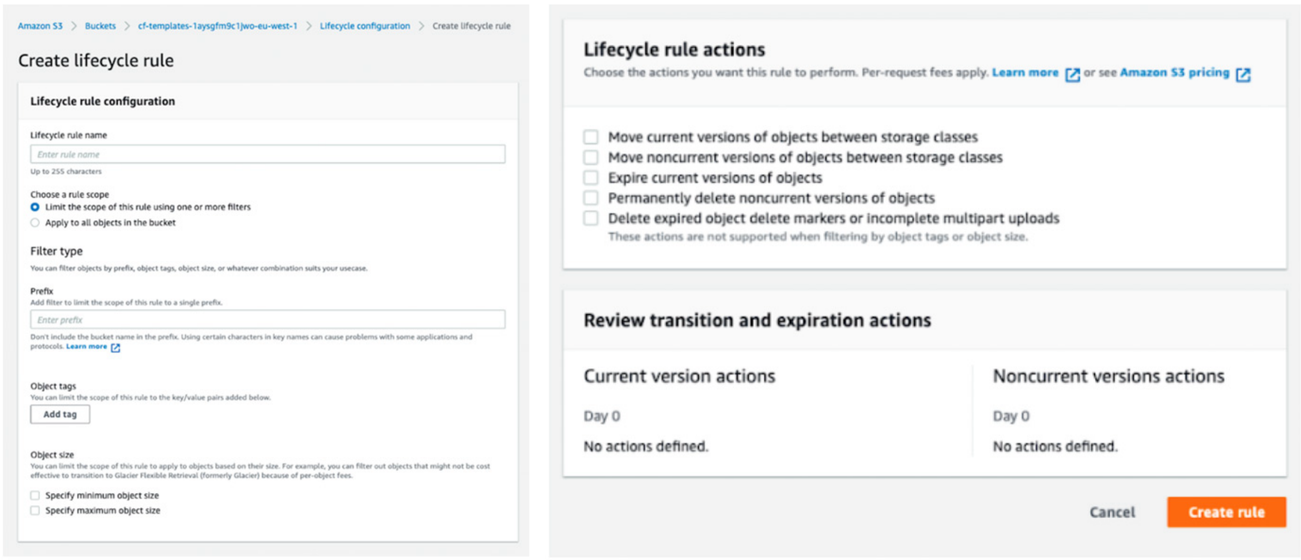
zesty

Storage

ROT is not doing you any favors. Redundant, Obsolete, and Trivial data clogs your cloud infrastructure, slows down compute times, and unnecessarily compounds costs. Here are a few strategies to eliminate **ROT.**

Delete unused object storage files

You can use an S3 lifecycle rule to set a policy that will automate the process of moving objects that aren't accessed frequently to an S3 bucket which takes slightly more time for data to be retrieved but provides a cost-effective storage location. You can set the policy of the time duration in which data can be moved across, and eventually deleted. Alternatively, this process can be automated by using ***S3 intelligent tiering***.



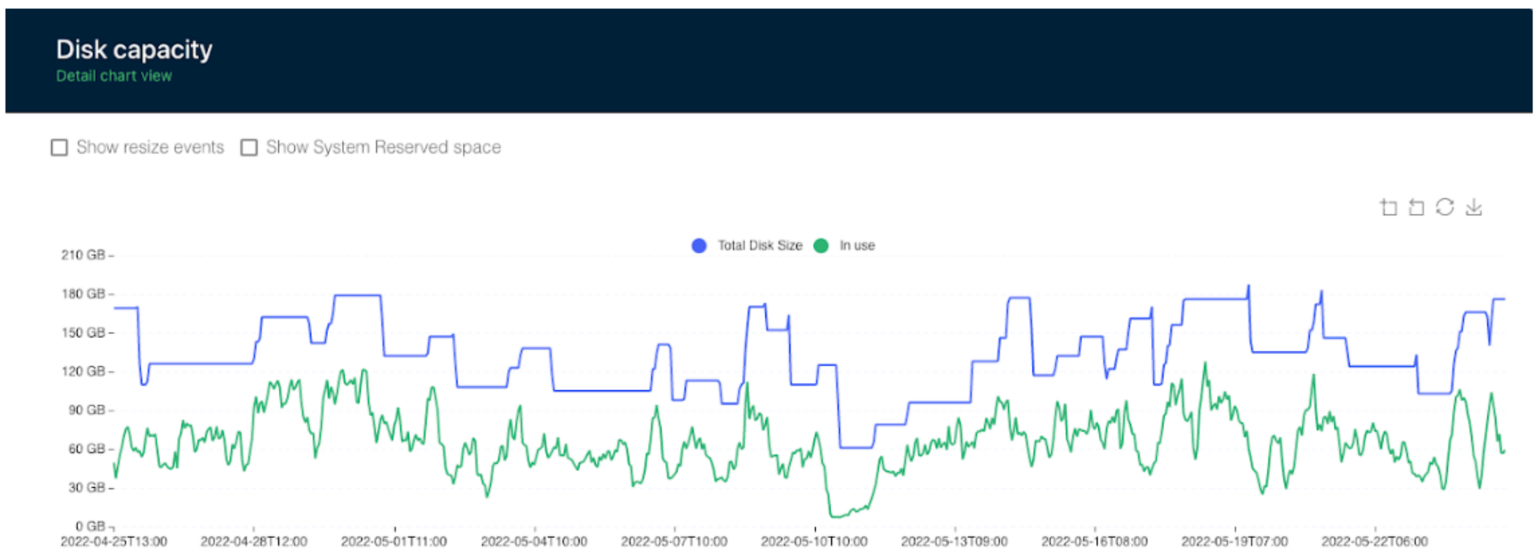
Create a lifecycle rule to delete unused S3 buckets

Identify volumes that have very low activity

This data is clogging up your infrastructure and may be good candidates for deletion. To do this you can use ***EBS Volumes Check*** in Trusted Advisor to identify these underutilized and possibly orphaned volumes. You can also do this using ***storage management and monitoring tools***. Take a snapshot to ensure that the data isn't lost, and then delete the volume.

Auto Scale your EBS Volumes

With the need to avoid application failure or slowdown, there is an industry-wide tendency to over-provision EBS storage volumes which results in paying between 2-5 times extra in cloud storage that doesn't end up getting used. However, it is possible to automatically scale volumes to application demand by using solutions like ***Zesty Disk***, which automatically adds filesystem storage when demand rises and removes them when demand drops off. This ensures that your application is always running optimally and cost-efficiently.



Zesty Disk auto-scales disk capacity to fluctuating demand

Automate the management of Snapshots

To make this process more efficient, you can automate the management of snapshots for volumes that are getting old and are rarely used by using ***Amazon Data Lifecycle Manager***.

Remove redundant snapshots

These may be orphaned snapshots or old snapshots that haven't been used for a while (the common parameter is 30 days). Snapshot retention policies can be set and modified so snapshots that aren't needed are no longer kept.

Event history Info

Download events on page

User name | DataLifecycleManager | 30m 1h 3h 12h Clear custom

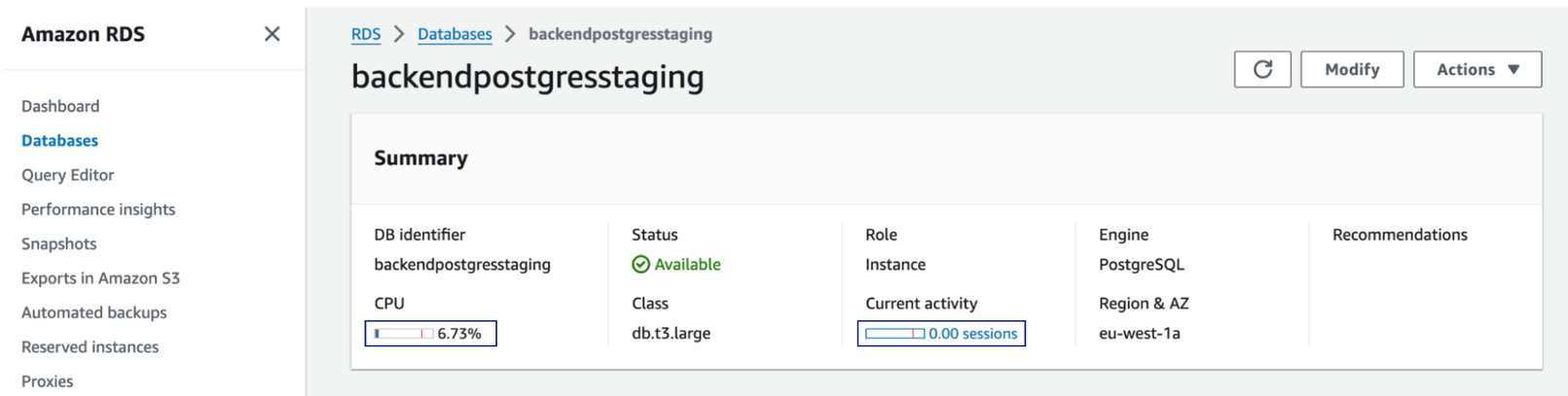
Event name	Event time	User name	Resource type	Resource name
CreateSnapshot	July 16, 2020, 06:23 (UTC-07:00)	DataLifecycleManager	AWS::EC2::Snapshot, AWS::EC2::Volume	snap-0f81c673b0b4dab67, vol-0794338b9a1ac2f66
DescribeVolumes	July 16, 2020, 06:23 (UTC-07:00)	DataLifecycleManager		
DescribeSnapshots	July 16, 2020, 06:23 (UTC-07:00)	DataLifecycleManager		
DescribeVolumes	July 16, 2020, 06:19 (UTC-07:00)	DataLifecycleManager		
DisableFastSnapshotRestores	July 16, 2020, 06:17 (UTC-07:00)	DataLifecycleManager		
DescribeFastSnapshotRestores	July 16, 2020, 06:15 (UTC-07:00)	DataLifecycleManager		
DescribeSnapshots	July 16, 2020, 06:15 (UTC-07:00)	DataLifecycleManager		
DeleteSnapshot	July 16, 2020, 06:15 (UTC-07:00)	DataLifecycleManager	AWS::EC2::Snapshot	snap-0dbe04c084c379974
DisableFastSnapshotRestores	July 16, 2020, 06:15 (UTC-07:00)	DataLifecycleManager		
DeleteSnapshot	July 16, 2020, 05:57 (UTC-07:00)	DataLifecycleManager	AWS::EC2::Snapshot	snap-099750f6fde776c7c
DescribeSnapshots	July 16, 2020, 05:57 (UTC-07:00)	DataLifecycleManager		

Automate EBS Snapshots using AWS data lifecycle manager

Database

Remove idle DB instances

If you have any DB instances that have not had any connection over the last seven days, it might be time to let go of them. Otherwise, you're paying for them to just sit around. You can identify them by using the ***RDS Idle DB instances check*** in Trusted Advisor. Additionally, you can have them automatically deleted by setting up ***the stop and start capability*** of Amazon RDS databases.



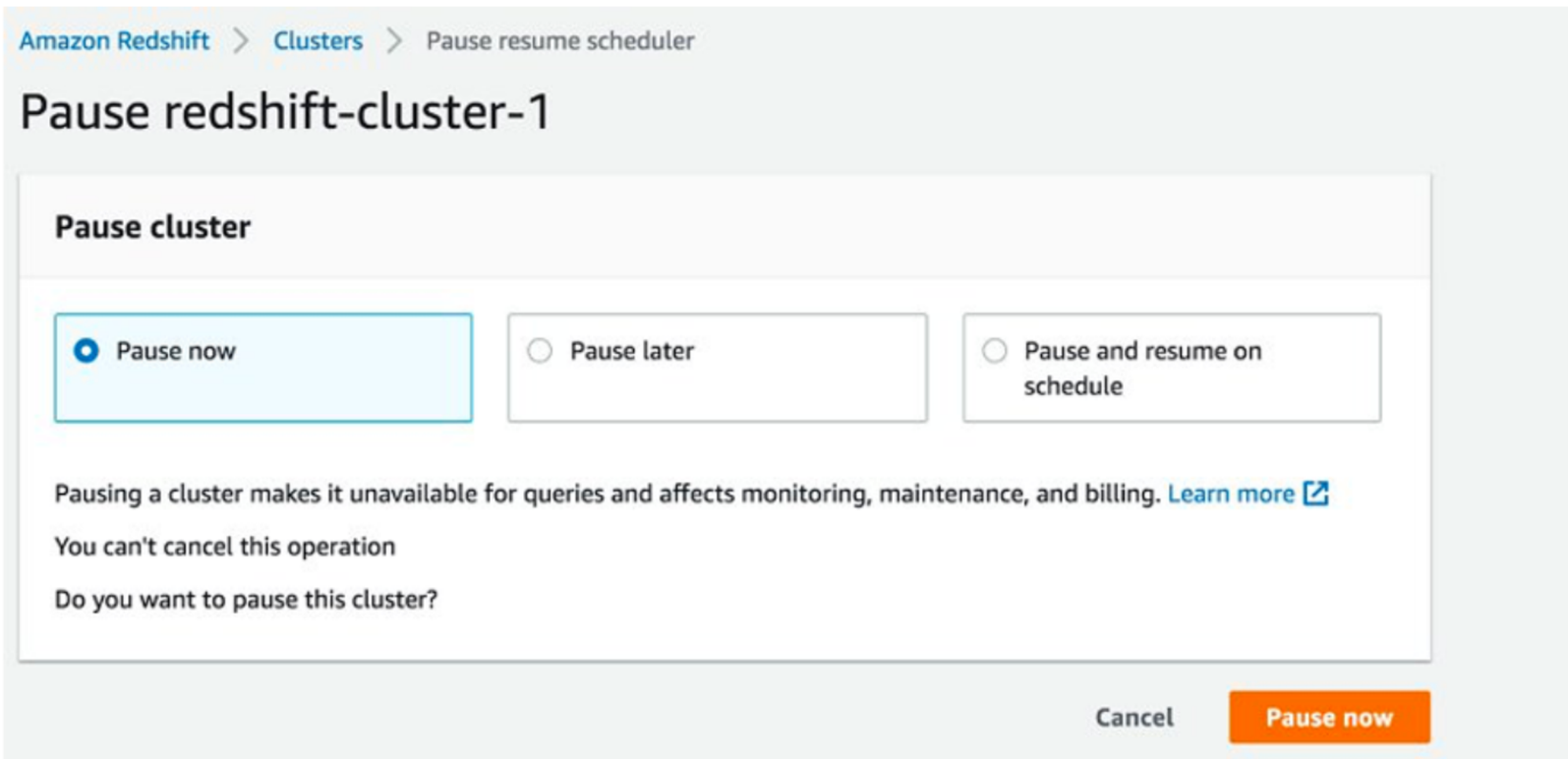
RDS idle DB instance check

Stop paying for idle Redshift clusters

It's easy to overprovision Redshift clusters or leave them running during the evenings, weekends, and holidays causing you to pay for Redshift nodes when they are idle. Instead, it's possible to resize, pause and later resume nodes in line with your fluctuating needs. This equally applies to ElastiCache and Elasticsearch.

Remove old Redshift clusters

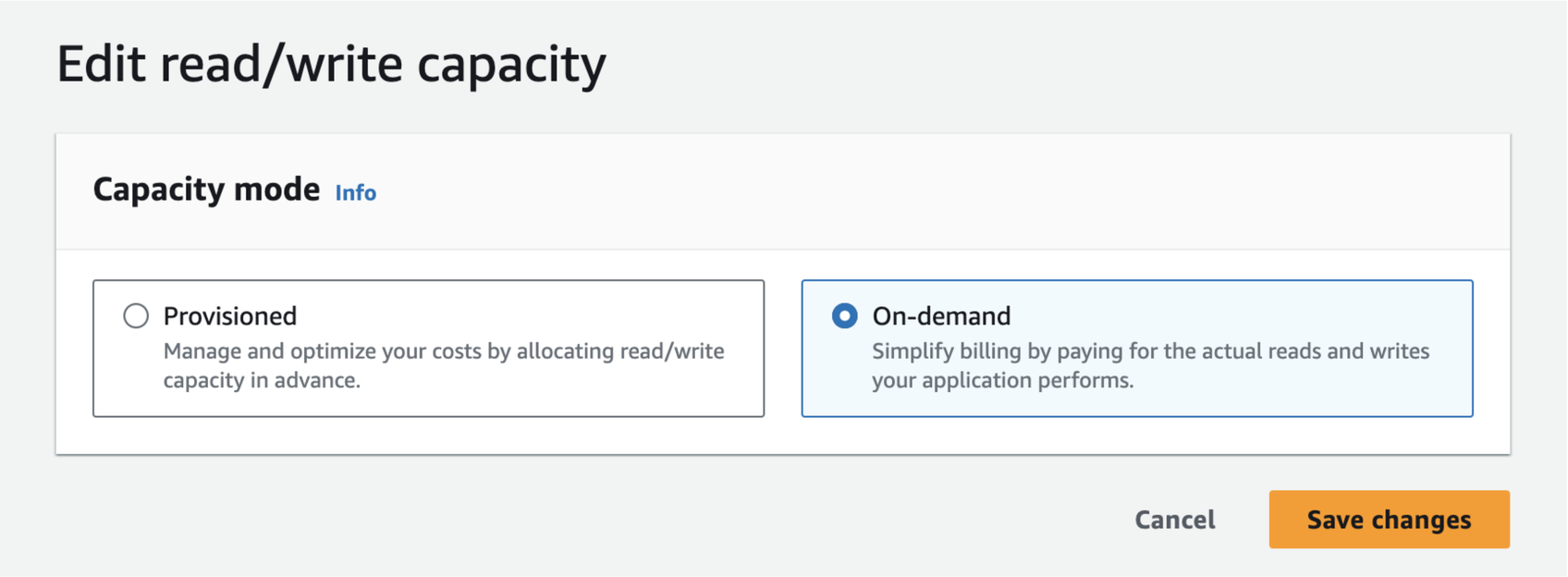
It is recommended to delete Redshift clusters that have had no connection for over seven days and less than 5% cluster-wide average CPU utilization for 99% of the last seven days. To put a stop to the cost generated by these underutilized clusters you can identify them by doing a ***Redshift clusters check*** and then pause them to suspend their compute and still retain the underlying data structures. This can be configured on the Amazon Redshift console or CLIs.



Remove old and underutilized redshift clusters

Monitor your DynamoDB usage

It's easy for DynamoDB usage to fluctuate and have costs spiral out of control. There are two metrics for analyzing usage; read capacity using ConsumedReadCapacityUnits and write capacity using ConsumedWriteCapacityUnits. Once you've identified general usage patterns, it is recommended to put your steady-state usage on a discount savings program. Only use On-demand to pay per request for data reads and writes your applications perform as the workload ramps up or down. This ensures you only pay for what you use without the risk of overprovisioning capacity.



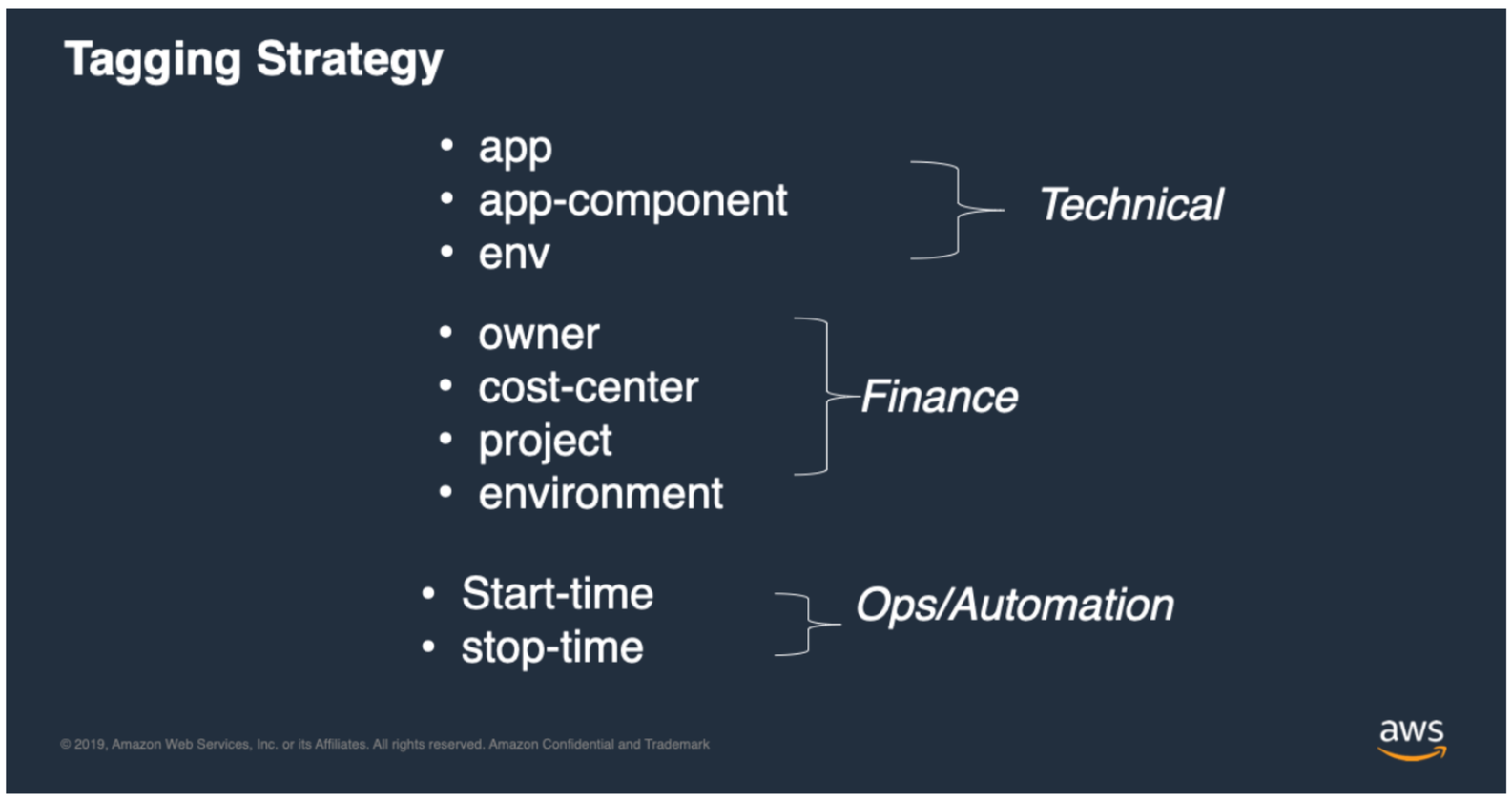
Zesty is breaking new ground with intelligent cloud management technology that auto-scales cloud resources to fit real-time application needs.

As today's cloud environments become increasingly dynamic, Zesty enhances cloud efficiency, improves DevOps productivity, and reduces cloud costs with zero human input. As a result, DevOps engineers can spend less time on manual cloud infrastructure tasks and can enjoy the cloud's flexibility and scalability without worrying about cost or maintenance concerns. Zesty was founded in 2019 in Tel Aviv and is used by leading organizations such as Armis, Gong, GrubHub, Heap, and others. For more information, visit ***Zesty.co.***

Compute

Always, but always, tag

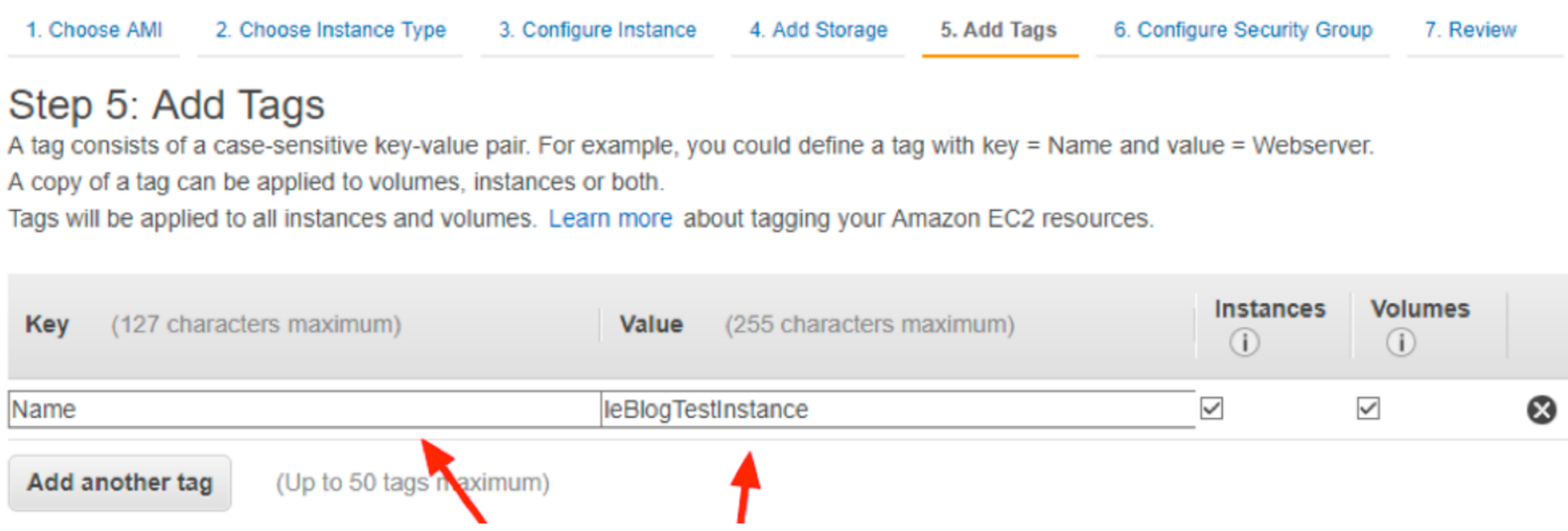
Aside from being a cornerstone tool to support cost optimization, tagging is needed to identify numerous resources you might have, allowing you to collect metrics based on the varied purposes of those applications. Tagging supports the cost allocation of resources, providing insight into the costs generated by each instance, supporting chargeback and payback, and alerting regarding budgets that are about to be exceeded. A good way to start implementing a tagging strategy is to define a tagging dictionary, where you define the "rules of the game" of how every resource should be labeled.



Apply tags per instance in Cost Explorer

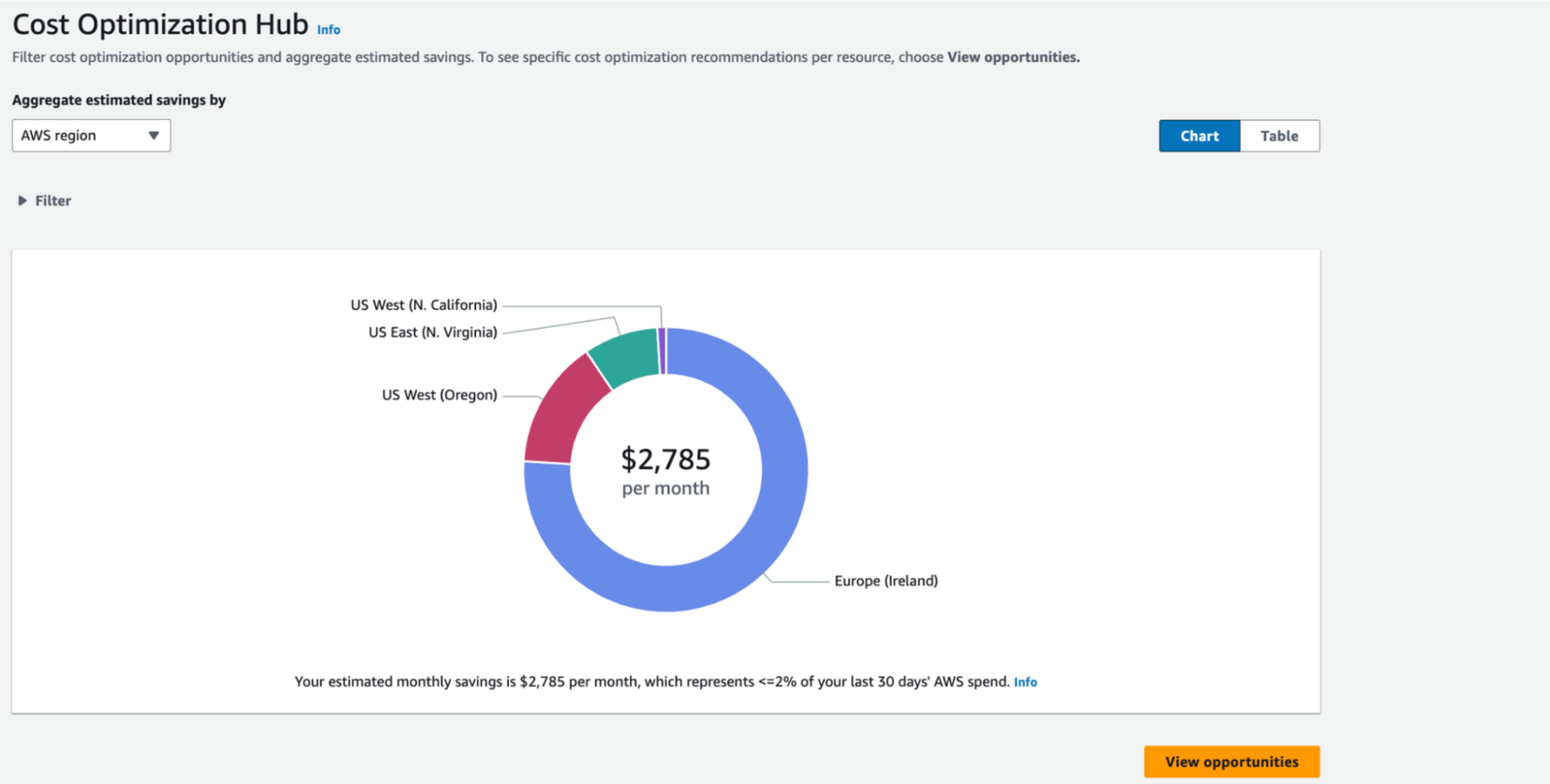
Indicate tags for cost allocation

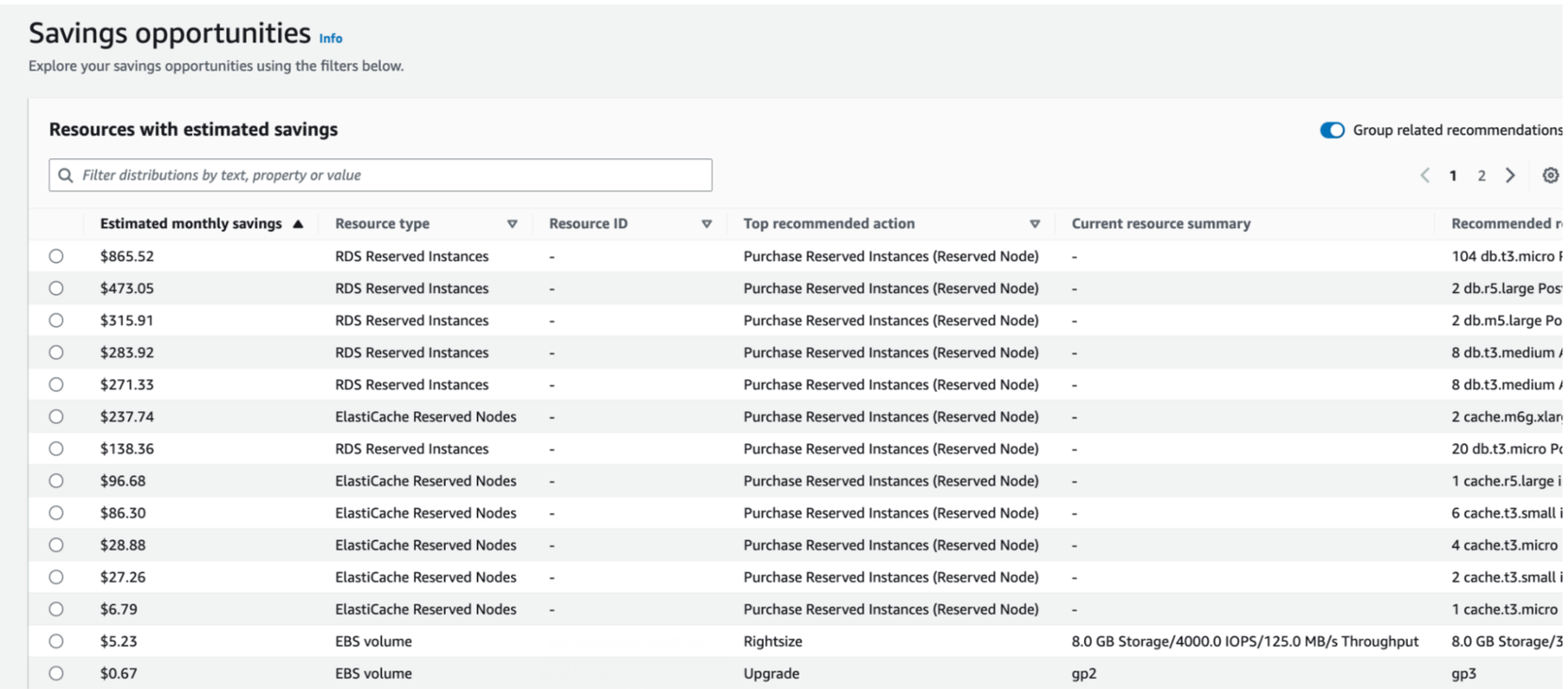
Tags should be activated for cost allocation to indicate to AWS that the associated cost data should be made available throughout the billing pipeline. Once activated, cost-allocated tags can be used to group or filter resources in Cost Explorer and to refine AWS budget criteria.



Check optimization opportunities using the AWS Cost Optimization Hub

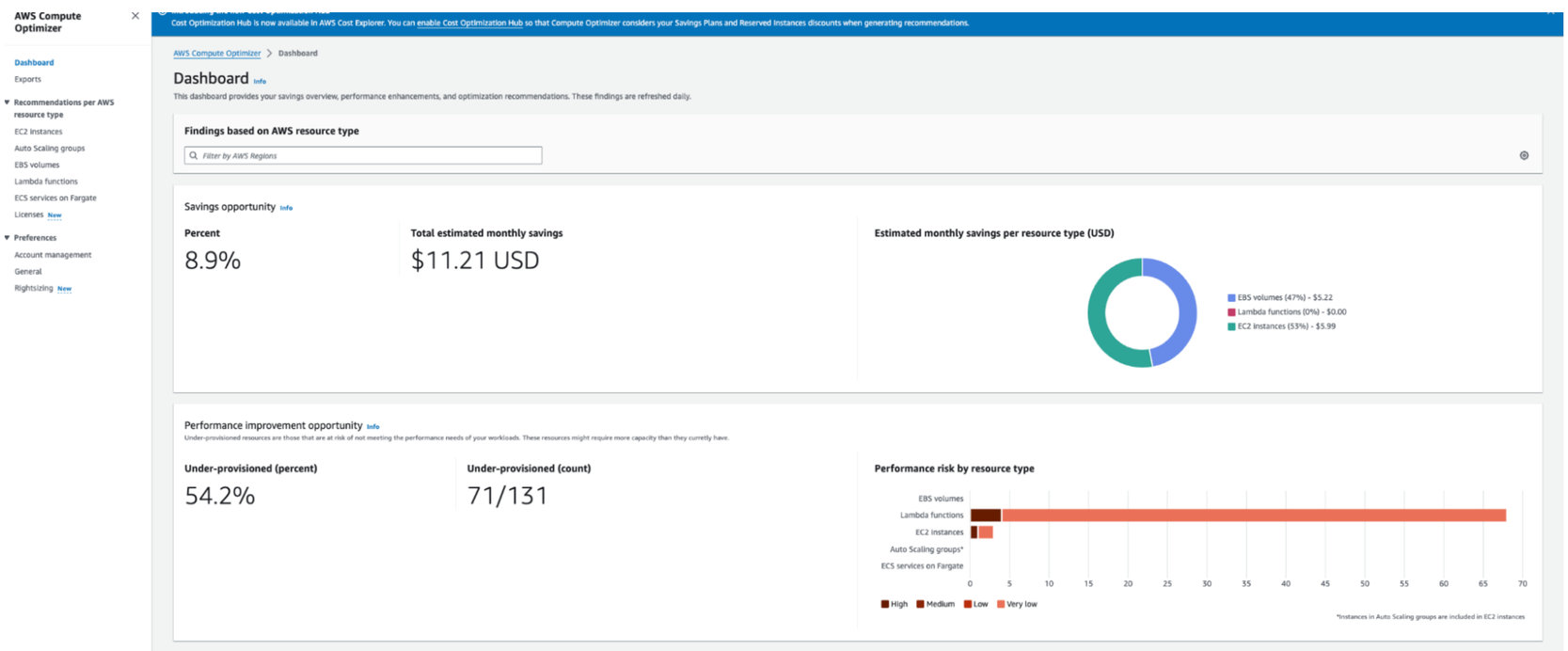
Using the Cost Optimization Hub, you can identify, filter, and consolidate over 15 types of AWS cost optimization recommendations, such as rightsizing your EC2 instances and your Lambda function or upgrading your EBS volume type. You can also find recommendations related to Graviton migration, idle resource detection, or Savings Plans. To help you compare and prioritize the recommendations, the Cost Optimization Hub provides an estimation of the potential savings, taking into account your specific AWS discount plans.





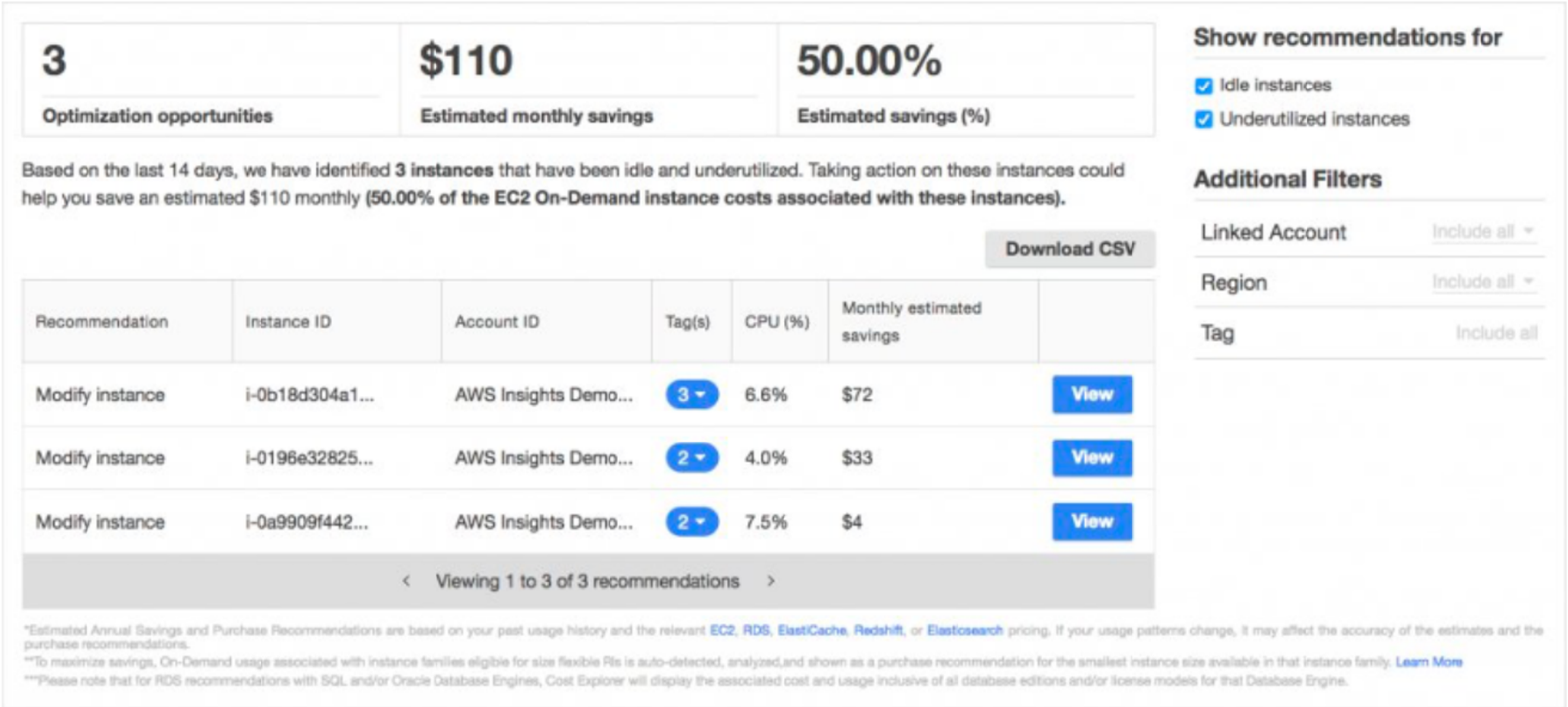
Estimated monthly savings	Resource type	Resource ID	Top recommended action	Current resource summary	Recommended r
$865.52	RDS Reserved Instances	-	Purchase Reserved Instances (Reserved Node)	-	104 db.t3.micro F
$473.05	RDS Reserved Instances	-	Purchase Reserved Instances (Reserved Node)	-	2 db.r5.large Pos
$315.91	RDS Reserved Instances	-	Purchase Reserved Instances (Reserved Node)	-	2 db.m5.large Po
$283.92	RDS Reserved Instances	-	Purchase Reserved Instances (Reserved Node)	-	8 db.t3.medium /
$271.33	RDS Reserved Instances	-	Purchase Reserved Instances (Reserved Node)	-	8 db.t3.medium /
$237.74	ElastiCache Reserved Nodes	-	Purchase Reserved Instances (Reserved Node)	-	2 cache.m6g.xlar
$138.36	RDS Reserved Instances	-	Purchase Reserved Instances (Reserved Node)	-	20 db.t3.micro Pc
$96.68	ElastiCache Reserved Nodes	-	Purchase Reserved Instances (Reserved Node)	-	1 cache.r5.large i
$86.30	ElastiCache Reserved Nodes	-	Purchase Reserved Instances (Reserved Node)	-	6 cache.t3.small i
$28.88	ElastiCache Reserved Nodes	-	Purchase Reserved Instances (Reserved Node)	-	4 cache.t3.micro
$27.26	ElastiCache Reserved Nodes	-	Purchase Reserved Instances (Reserved Node)	-	2 cache.t3.small i
$6.79	ElastiCache Reserved Nodes	-	Purchase Reserved Instances (Reserved Node)	-	1 cache.t3.micro
$5.23	EBS volume		Rightsize	8.0 GB Storage/4000.0 IOPS/125.0 MB/s Throughput	8.0 GB Storage/3
$0.67	EBS volume		Upgrade	gp2	gp3

Later, we recommend you to enter the AWS Compute Optimizer which provides you with a better granularity of EC2, Lambda, ECS, Fargate improvement opportunities:



Shutdown idle EC2 instances

EC2 instances that are idle or have low utilization are still costing you money by the hour. To identify and remove these instances, use the Resource Optimization in Cost Explorer.



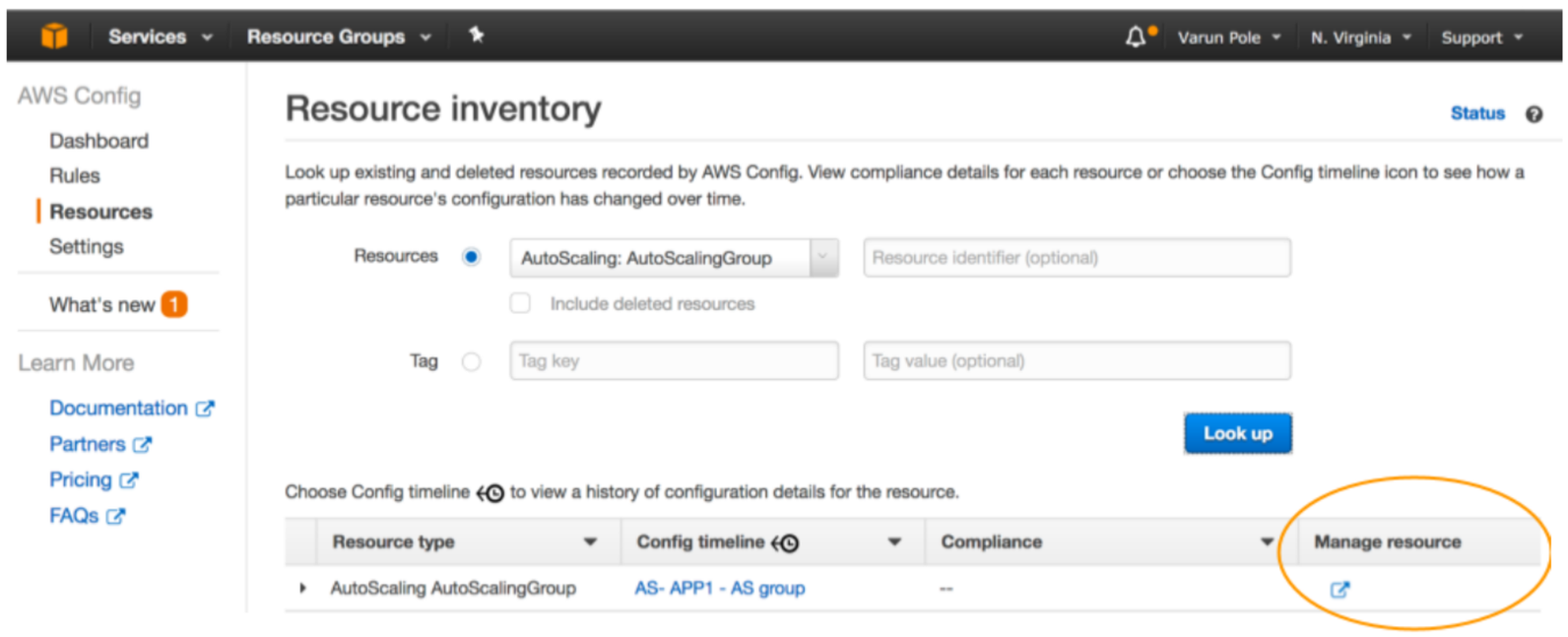
Recommendations for idle and underutilized instances

Pause instances when they're not needed

For instances that you want to keep, but are only used intermittently, you can stop or pause these instances when they're not needed using the AWS instance scheduler.

Tune your EC2 autoscaling groups configuration

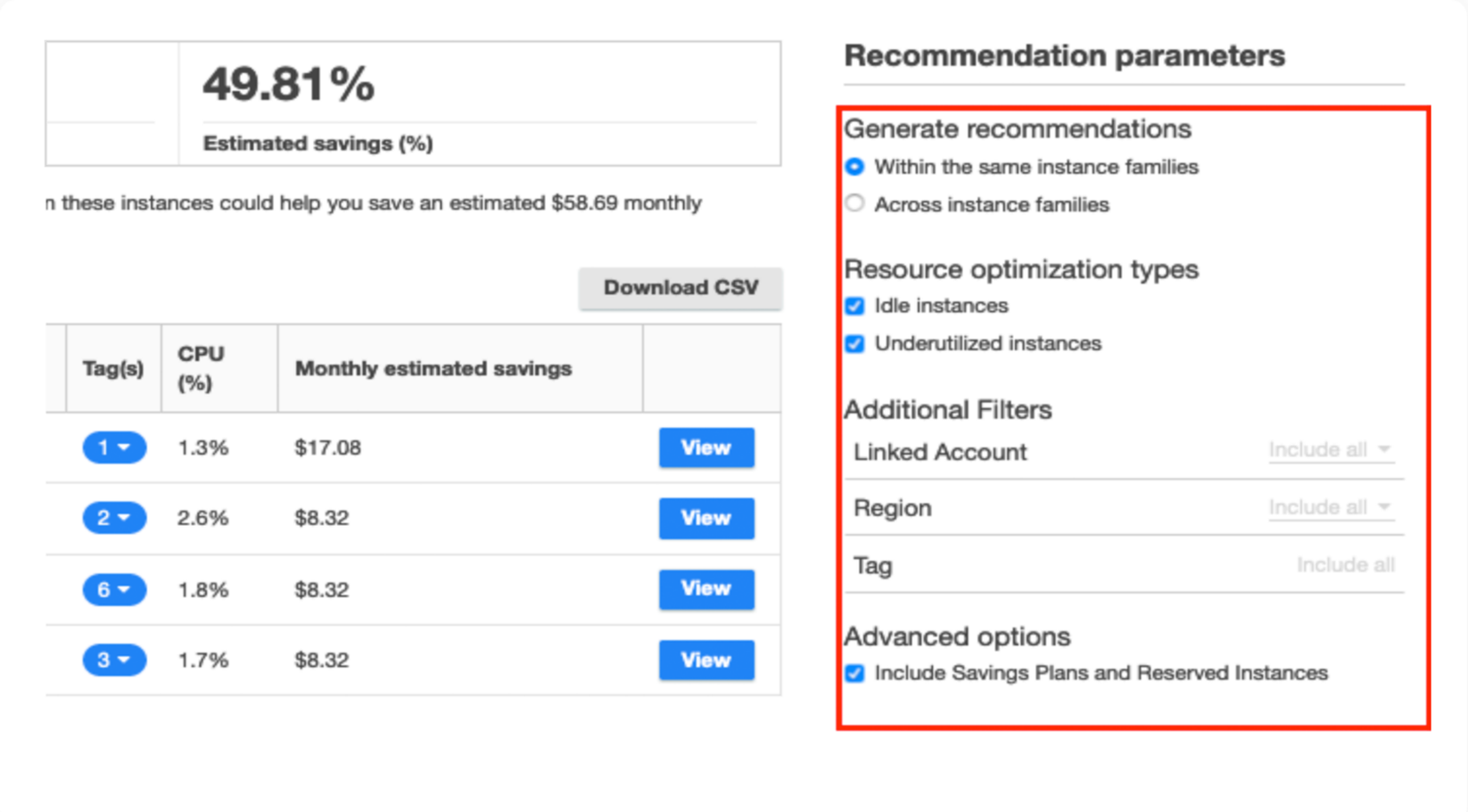
The auto-scaling group enables you to expand or shrink your EC2 fleet based on demand.
To improve cost efficiency, the scaling policy can be tuned to add instances less aggressively.
It can also be tuned to set a lower minimum for the number of instances that are needed
to serve end-user requests.



Change configurations to auto-scaling groups

Rightsize instances

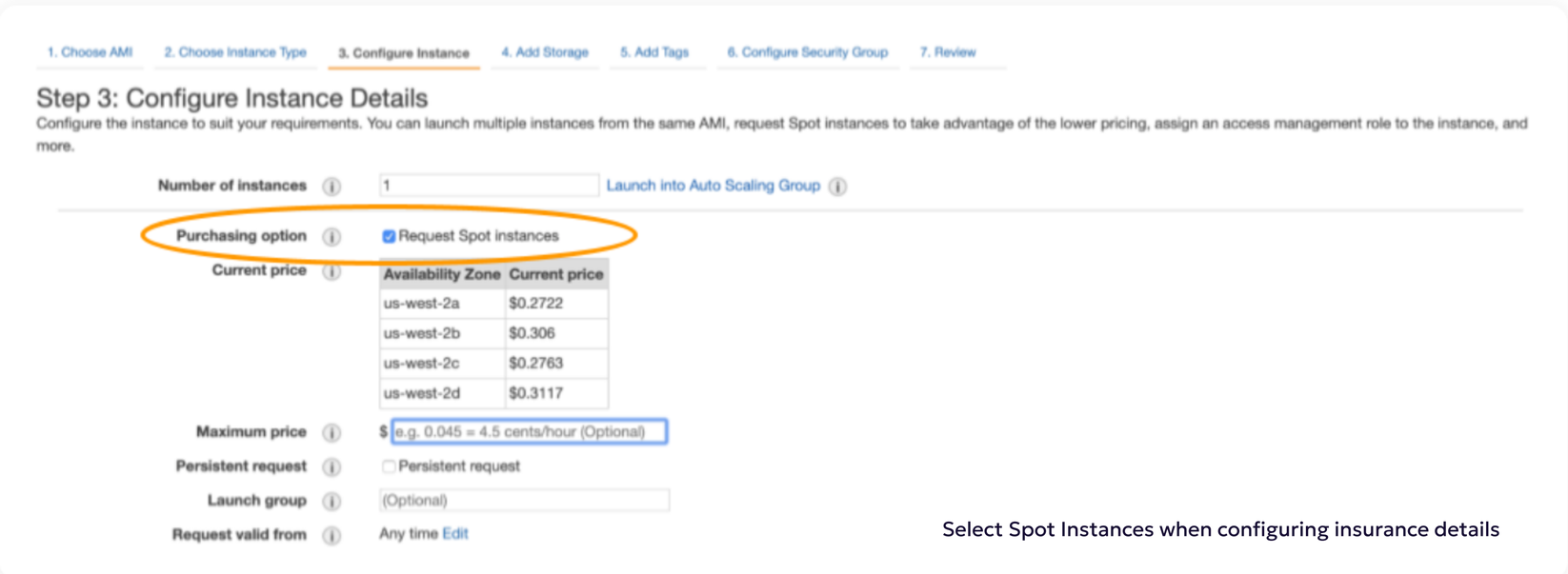
There is a tendency to overprovision instances with more memory and CPU than what's actually needed. To check whether your instances exceed your needs, you can use Cost Explorer which gives recommendations for downsizing within or across instance families, upsizing recommendations to remove performance bottlenecks, and recommendations for EC2 instances that are part of an Auto Scaling group.



Recommendations for idle and underutilized instances

Consider using Spot instances

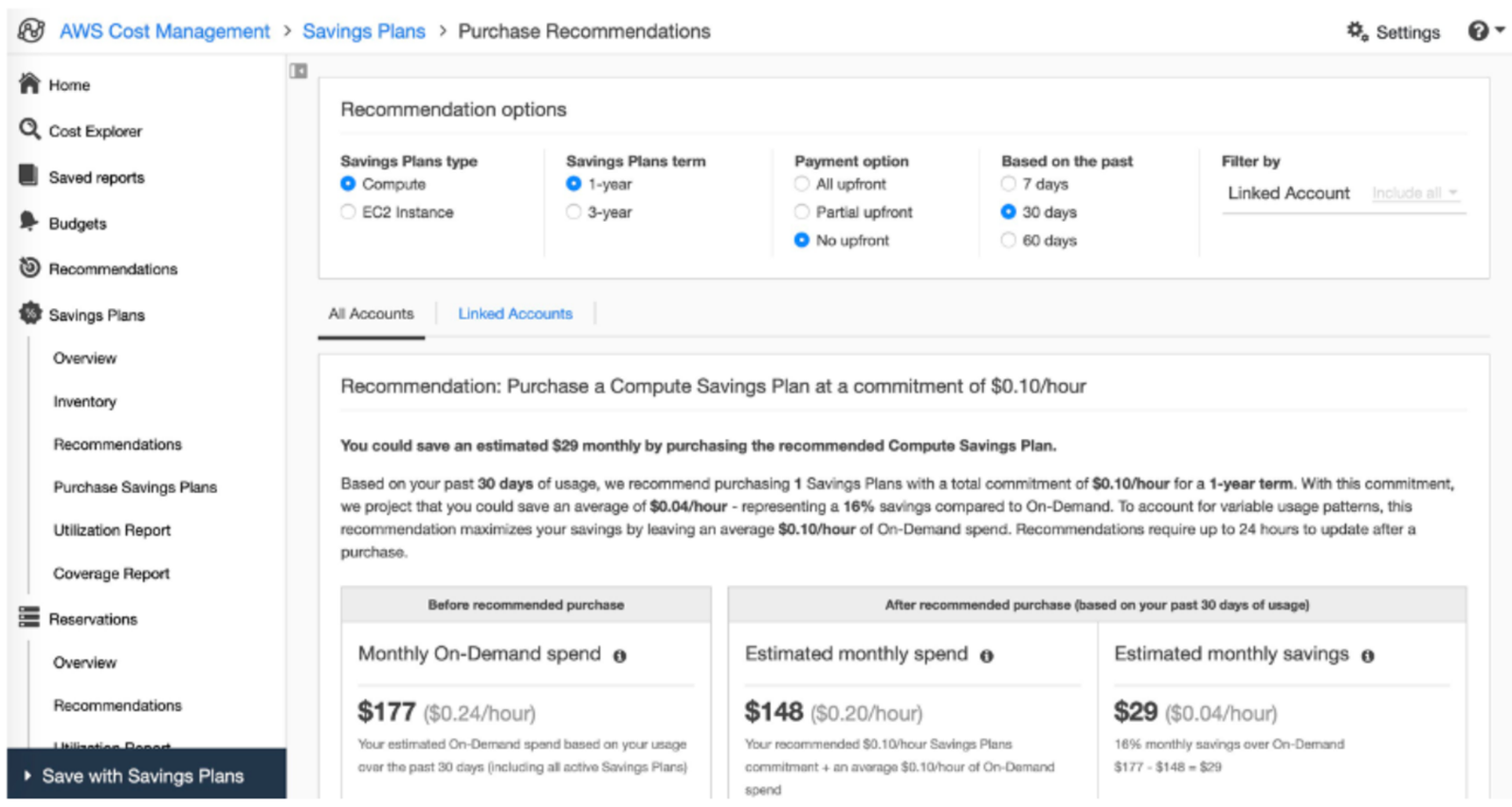
For stateless, fault-tolerant, and loosely coupled workloads, consider using Spot instances which can give discounts of up to 90% off On-Demand costs but can be reclaimed with just a two-minute warning if AWS needs these instances back. Spot instances are ideal for test and development workloads such as CI/CD, and high-performance computing.



Select Spot Instances when configuring insurance details

Apply Savings Plans to Fargate and Lambda costs

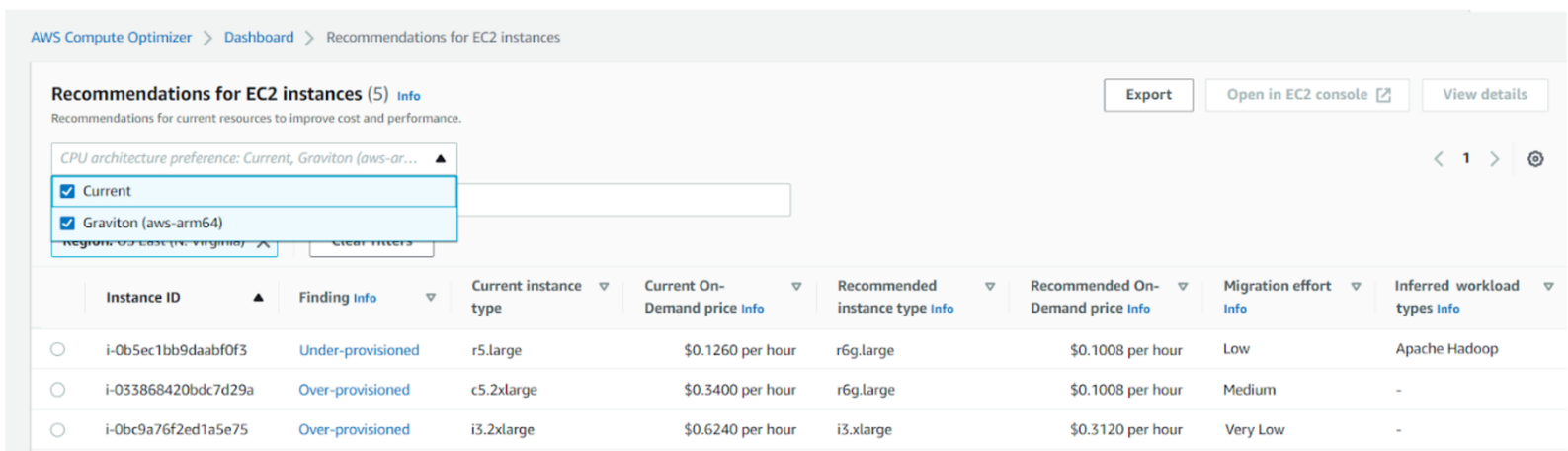
For these compute operations, Savings Plans can be applied to provide a discount of up to 17% from On-Demand costs.



Savings Plans recommended options

Migrate to Graviton Instances (where you can)

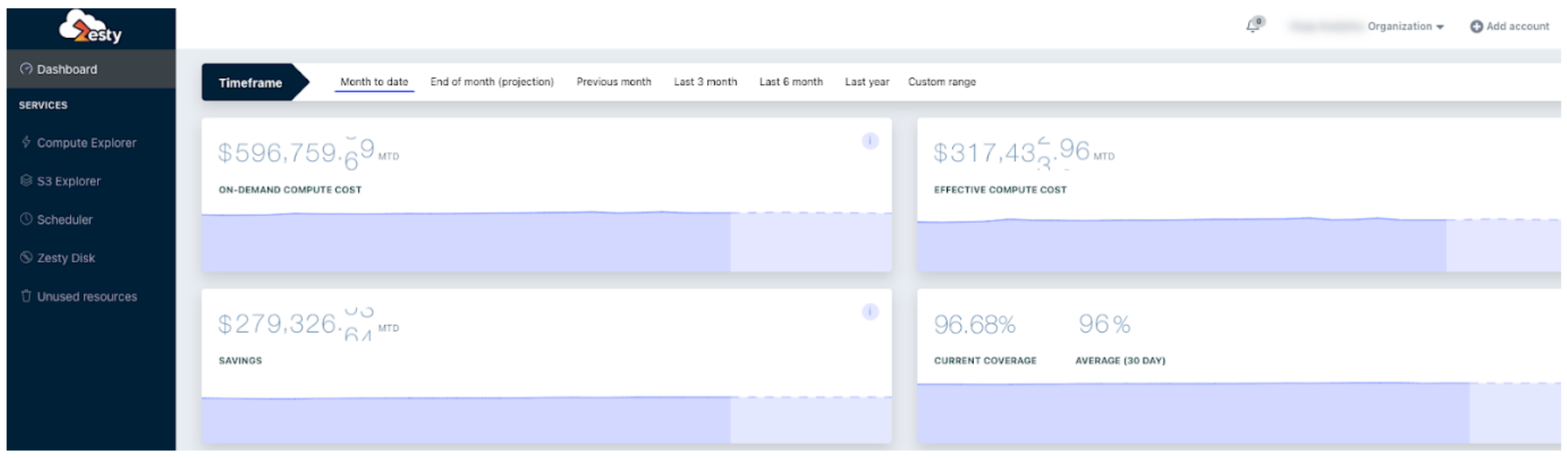
The use of AWS on ARM processing technology is much more efficient and powerful at running servers, requiring just 60% less electricity to run just one server. This entails that for the same cost in power that it takes to run one intel CPU, AWS can run two ARM servers. These ARM servers can be used by selecting Graviton instance types. Graviton processors will reduce your EC2 instance bill by 40% yet still achieve the same level of performance. The challenge is that ARM servers are not similar to Intel processes and the two technologies are not compatible, making it difficult to migrate over. To do so, you need to rewrite and compile your code. While not always a feasible solution, when Graviton can be utilized it can be enormously cost-effective. Graviton can be a good fit for managed services, with users often starting with OpenSearch and RDS. Furthermore, by using Graviton 3 which was introduced at re:invent 2022, you can add another 25% on top of the 40% discount of Graviton 2, compared to the cost of an intel-based M5.



Select Graviton instances in AWS Compute Optimizer

Purchase 1 or 3 Year Discount Plans

If you're in a position where you have a fairly stable compute workload and can forecast the bulk of your usage a year in advance, or even three years in advance, then it is highly worthwhile to purchase Reserved Instances and Savings Plans commitments that will deliver substantial discounts from the cost of your EC2 On-Demand instances. The challenge is all too few of us have a workload that is so stable that you're able to predict what your EC2 usage will be next month, let alone, for the following year or three! In this situation, you may want to consider ***Zesty's Commitment Manager***. The solution automatically manages your discounted commitments for you, making it possible to leverage AWS's deepest discounts, without taking on the risk of over-committing. Commitment Manager saves users up to 50% of their EC2 workload. To find out more, ***read the solution overview***.



Zesty's Commitment Manager dashboard shows compute cost, amount saved, actual spend, and percentage of covered instances

While some of these recommendations will be easy to implement, there will be many others that will require time, effort, and even a holistic cultural change to organizational processes. Towards that end, you may want to set up a CCoE (Cloud Center of Excellence) with representatives from different departments, be they DevOps, Finance, Procurement, and Executives, that will work as a steering committee to implement many of these FinOps best practices.

As a rule of thumb, seek to automate wherever you can, as that will deliver cost savings without adding any manual effort, and will continue to scale as your business grows. For the rest, we wish you the best of luck on your cost optimization journey!

Zesty is breaking new ground with intelligent cloud management technology that auto-scales cloud resources to fit real-time application needs.
As today's cloud environments become increasingly dynamic, Zesty enhances cloud efficiency, improves DevOps productivity, and reduces cloud costs with zero human input. As a result, DevOps engineers can spend less time on manual cloud infrastructure tasks and can enjoy the cloud's flexibility and scalability without worrying about cost or maintenance concerns. Zesty was founded in 2019 in Tel Aviv and is used by leading organizations such as Armis, Gong, GrubHub, Heap, and others. For more information, visit ***Zesty.co.***