VEEAM

16 powerful tips to lower your AWS spending

Andrey Zhelezko
Technical Product Analyst

Contents

The shift to public cloud providers like AWS offers many advantages for companies. According to AWS CEO Andy Jassy, the conversation starter for cloud adoption is almost always cost savings. For many companies, this means trading the old model of using heavy capex for capital to invest in data centers and servers upfront to a variable expense model that is pay-as-you-go. However, organizations often forget to adapt the strategy of using the cloud. Rather, they continue to operate new services with the old mindset of traditional data centers, and because of that, they lose money.

To maximize the cost optimization model of AWS, companies need to plan accordingly and leverage the many tools that AWS provides for monitoring resource usage. In this article, we're going to discuss the most popular, yet often overlooked, tools. This is not intended to be an ultimate guide, but it is a great place to start planning AWS usage or evaluating existing practices. I'll spend most of the time on EC2, EBS and S3 services, but will leave hints for some others too.

1. Talk to your AWS representative

This might sound counterintuitive if we don't see the cloud model on a bigger scale. **AWS always prefers businesses to use something** rather than nothing. That's why they will be eager to talk, share best practices for cost optimization and provide their own instructions. Get in touch with your AWS account representative and show them what you're up to.

In addition, small startup businesses may be entitled to AWS credits through business incubators or AWS itself. Again, talk to people, **explain the action plan** and ask for the credits to run testing and proof of concept (POC) implementations. On the other hand, big companies can get a discount by reaching a certain level of resource consumption. The key here, again, is to know an account manager and to be transparent.

2. Start by using the free tier

Going back to the startups out there, why not start from the free tier? AWS gives a small number of resources for a limited or even unlimited time. However, there are some performance, time and volume constraints. For example, EC2 instance types are just limited to t2.micro or t3.micro and DynamoDB database only comes with 25 GB of space, but it's still better than nothing. The free tier is currently available for over 60 products. There are also three different types of offers depending on the product used:

- Always free: Doesn't expire at all
- 12 months free: Expires within 12 months after the initial signing up
- Trials: Short-term free trials

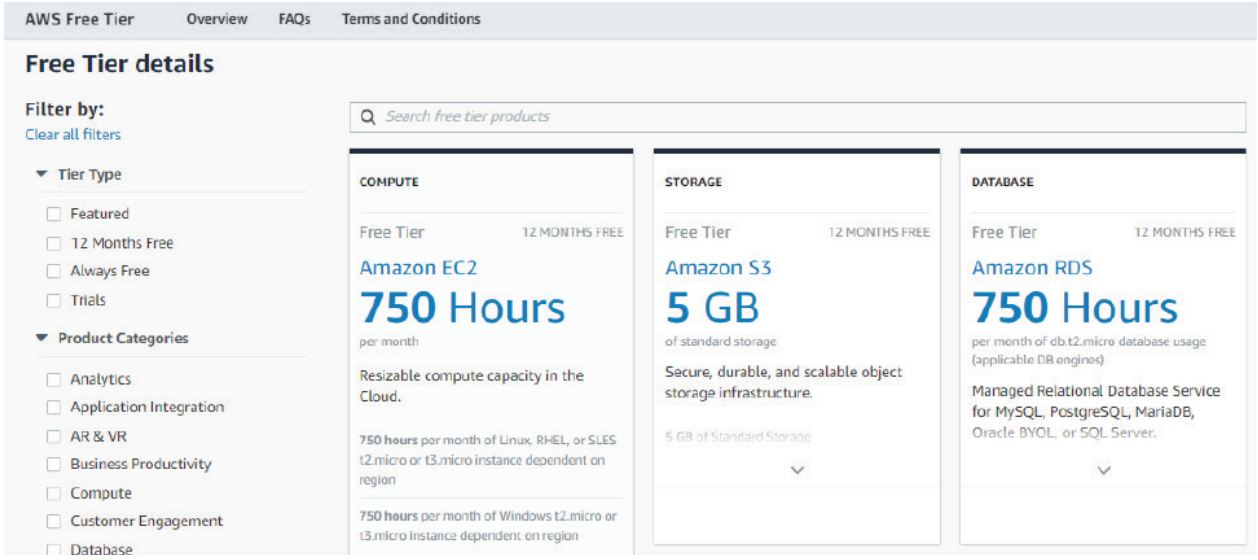

Figure 1. AWS free tier page

3. Consider regions accordingly

It's easy to notice that the price for AWS services depends on the physical location where data centers are placed. This may sound obvious but migrate resources to a region with lower prices **when it makes sense.**

Let me share a small example by calculating the current prices. Imagine a need to run 10 t3.2xlarge instances in Europe with 150 GB SSD gp2 disk capacity each. If you select a data center in Frankfurt, Germany, then using compute AWS will charge $0.5312/hour, whereas a data center in Stockholm, Sweden would cost $0.4928/hour. Add in the storage cost of $0.119 vs $0.1045 per GB-month and run them for a year. The difference will equal about 24*365*10*$0.0384+ $0.0145*1,500 GB*12m = **$3,364 + $261 = $3,625** of annual cost reduction with the same service, simply because the cheaper region was chosen.

Does this mean we have to abandon all highly priced regions and at once? Not necessarily. Like I mentioned before, do it when it makes sense. When the company is operating in Asia and trying to provide users with the lowest latency, there is no reason to be moving applications to the United States. Although, feel free to do that for less demanding services or static content (more about that later). Another possible caveat to watch for here is the usage of personal data. Regulations around the globe like General Data Protection Regulation (GDPR) or CCPA might be a roadblock, as some data must stay inside of a certain geographic area where users come from.

EC2 instances

4. Select appropriate EC2 instance types

An EC2 service would probably be one of the first picks for a cloud journey with AWS. With over 60 instance types available, picking the most suitable is often an overwhelming choice to make. Take a deep breath and think of the actual purpose of the instance. Based on this, you can narrow down the types that are the best fit by looking at the table below.

Use cases	**Example**	**Preferred instance type**
General purpose machines, which should be balanced on compute, storage and network.	Apache, NGINX, Kubernetes, Docker, VDI and development environments	T3
Compute bound applications that benefit from high performance CPU.	High performance web servers, highly scalable multiplayer gaming and video encoding	C4 and C5
Applications that process large data sets in memory.	High performance databases (e.g. SAP HANA), big data processing engines (e.g. Apache Spark or Presto) and high performance computing (HPC)	X1 and R5
Floating point number calculations, intensive graphics processing or data pattern matching.	Machine/deep learning applications, computational finance, speech recognition, autonomous vehicles or drug discovery	G4, F1 and P3
Intensive sequential read/write operations or handling large data sets.	NoSQL databases (e.g. Cassandra, MongoDB, Redis), scale-out transactional databases and data warehousing.	D2 and i3

Figure 2. EC2 instance type selection table

Remember, **right-sizing** is choosing the cheapest option that still meets performance requirements. A rule of thumb is to achieve 80% instance resource utilization over a long period of time.

5. Always prefer the most modern CPUs

Be sure to come back to the instance type table periodically, as technology doesn't freeze and CPU manufacturers are coming up with more performant and less energy-consuming CPUs almost every year and AWS implements them. For example, a simple switch from c4.xlarge to c5.xlarge for 10 instances would provide approximately **$2,500** in annual savings, while delivering more RAM (16 -> 20 GB) for each instance and around 5% better performance at the same time. What are you waiting for?

Region: US East (Ohio)

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
Compute Optimized - Current Generation					
c5.large	2	10	4 GiB	EBS Only	$0.085 per Hour
c5.xlarge	4	20	8 GiB	EBS Only	$0.17 per Hour
c5.2xlarge	8	39	16 GiB	EBS Only	$0.34 per Hour
c5.4xlarge	16	73	32 GiB	EBS Only	$0.68 per Hour
c5.9xlarge	36	139	72 GiB	EBS Only	$1.53 per Hour

Region: US East (Ohio)

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
Compute Optimized - Previous Generation					
c4.large	2	8	3.75 GiB	EBS Only	$0.10 per Hour
c4.xlarge	4	16	7.5 GiB	EBS Only	$0.199 per Hour
c4.2xlarge	8	31	15 GiB	EBS Only	$0.398 per Hour
c4.4xlarge	16	62	30 GiB	EBS Only	$0.796 per Hour
c4.8xlarge	36	132	60 GiB	EBS Only	$1.591 per Hour

Figure 3. C4 and C5 families comparison

The hint here is to check the AWS Compute Optimizer as this tool can advise this type of change.

6. Consider EC2 Instance Store for temporary data

Typically, AWS users would look at EBS storage (i.e., disks) when setting up new EC2 instances. Those disks can be attached to instances, detached from them and snapshotted for data protection cases. Whenever the instance is stopped, the data remains on the disk and doesn't go anywhere.

There is another option that is worth considering: Using local disks via the EC2 Instance Store. The major difference between these disks is that they are cleaned once the corresponding instance is stopped. They are "free" as users are paying for instance usage only.

While it's understandable that you wouldn't want to use local disks for valuable data, there are other cases when they fit perfectly, such as with temporary data like cache or logs. And don't worry, there is just enough performance, as underlaying storage is as fast SSD or even NVMe. Using the EC2 Instance Store will result in less EBS consumption as well as smaller monthly bills.

7. Make use of spot instances

The concept of spot instances is better understood when compared to the public market. The price fluctuates based on the supply and demand of available unused AWS EC2 capacity. AWS users may go there and shout (i.e., bid) their desired price. If the demand isn't that great, the market would agree to temporarily sell at the lower price, resulting in a price reduction that's three to six times less than the regular price.

So, what's the catch? In a moment of increased demand, you might not receive the required resources. As the price goes above that preferred limit, provisioned instances will be automatically terminated with short notice. Obviously, no one wants to have critical data exposed to such fluctuations. However, there are many cases where this model will work just fine (winking to all **container** fans here). Media rendering, big data, analytics and web services behind a load balancer should be among the first candidates for this feature too.

For example, here's a Spot Instance pricing history for the North Virginia region, showing the demand and pricing of t2.large instances over the last three months.

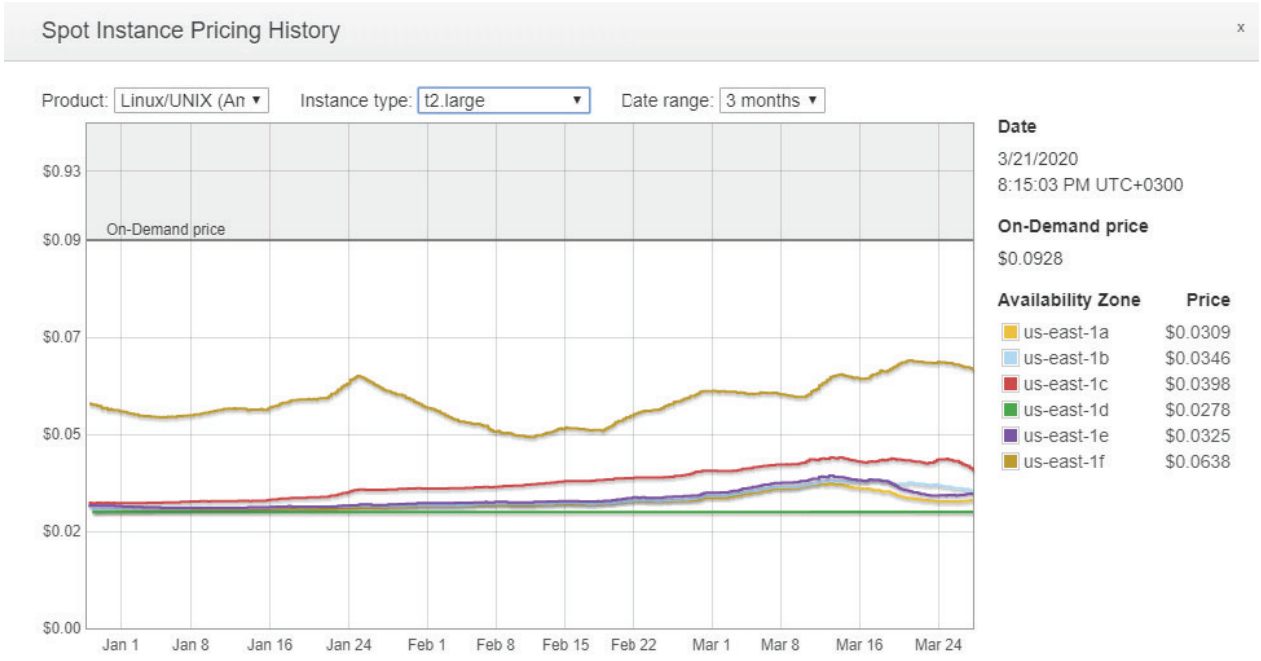

Figure 4. AWS Spot Instance pricing history

What we can see from the table above is:

- The real price could be as low as three times less than on-demand
- All six AZs have spared capacity, although they aren't equal
- The price is not stable, so it would be logical not to bid the lowest rate possible

Complement this technology with active CloudWatch and Auto Scaling and as soon as a two-minute termination alert is received, the system could rebalance the load and switch to on-demand instances. Voila!

8. Aim to predict the use of services and commit upfront

Although we talk about public cloud flexibility and the shift from CAPEX to OPEX, some of the AWS features that reduce invoice, namely reserved instances (RIs) and saving plans, would ironically look like going back to CAPEX. However, with an actual discount of over 50%, they should be on everyone's "best practices for implementation" list.

Start with looking at RIs, which provide a discounted hourly rate and an optional capacity reservation for EC2 instances. In exchange for a commitment of one or three years, you get a discount on instance costs. **On-demand instances** are a good option for someone who prefers to provision workloads with unlimited flexibility. However, for constantly running workloads with a predictable load (i.e., web services), RIs are much better.

Savings summary:

- The more reservation time taken, the better the discount
- The more paid upfront, the better the discount (but a zero upfront payment is also available)
- Standard class is cheaper than convertible (but the instance type can't be changed after creation)

Watch out for **convertible instances,** as they can't be downsized or sold on the AWS Marketplace. Start with the smallest instance and upgrade when needed to avoid being left with a commitment to a monthly payment for up to 36 months.

The Saving Plans feature extends this same concept while allowing for more flexibility. It also seems that they are going to gradually supersede RIs. One may want to consider RIs in early 2020 only if they are using RDS databases, as Saving Plans currently don't support that. For the rest of the cases, check out the AWS comparison here:

Comparing Savings Plans and RIs

	Compute Savings Plans	EC2 Instance Savings Plans	Convertible RIs*	Standard RIs
Savings over On-Demand	Up to 66 percent	Up to 72 percent	Up to 66 percent	Up to 72 percent
Lower price in exchange for monetary commitment	✓	✓	—	—
Automatically applies pricing to any instance family	✓	—	—	—
Automatically applies pricing to any instance size	✓	✓	—**	—**
Automatically applies pricing to any Tenancy or OS	✓	✓	—	—
Automatically applies to Amazon ECS using Fargate	✓	—	—	—
Automatically applies to Lambda	✓	—	—	—
Automatically applies pricing across AWS Regions	✓	—	—	—
Term length options of 1 or 3 years	✓	✓	✓	✓

Figure 5. RIs and Saving Plans comparison

Note: Be sure to check out the AWS Pricing Calculator tool (see below) when considering RIs and Saving Plans. This instrument will be very helpful, as it shows all the potential savings and it allows for much better planning.

Object storage

9. Hail to object storage

AWS provides multiple storage tiers at different prices that are designed to meet requirements for performance, availability and durability. There are three broad categories of storage services offered: Object, block and file storage. Amazon's object storage offering, Simple Storage Service (S3), is the most cost effective of the three storage categories. Within Amazon S3, you can easily move data between the storage classes further down the road to balance the frequency of access with price to optimize storage costs. All storage types have different usage scenarios and their pricing varies.

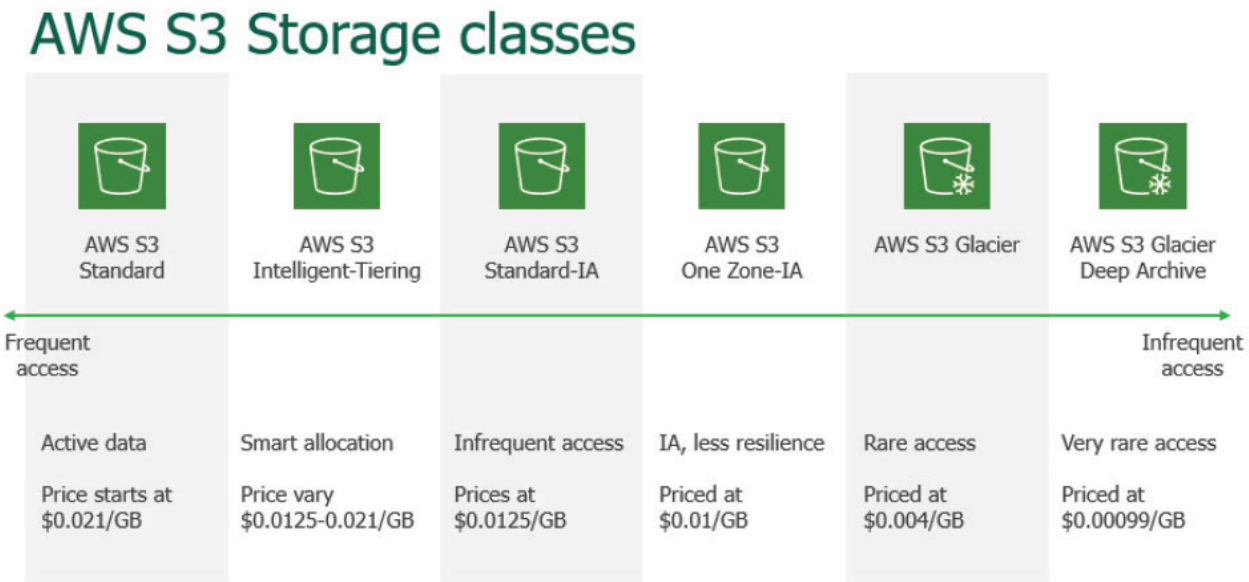

Figure 6. AWS S3 storage classes partial comparison

The smart approach here is to combine them, depending on the type of task, object nature and access frequency. Click on desired S3 bucket, select "management", then "analytics" and then add "storage class analysis," which will be helpful to see access patterns. While it doesn't give recommendations for transitions to the One Zone-IA or S3 Glacier, it provides deeper visibility into the data.

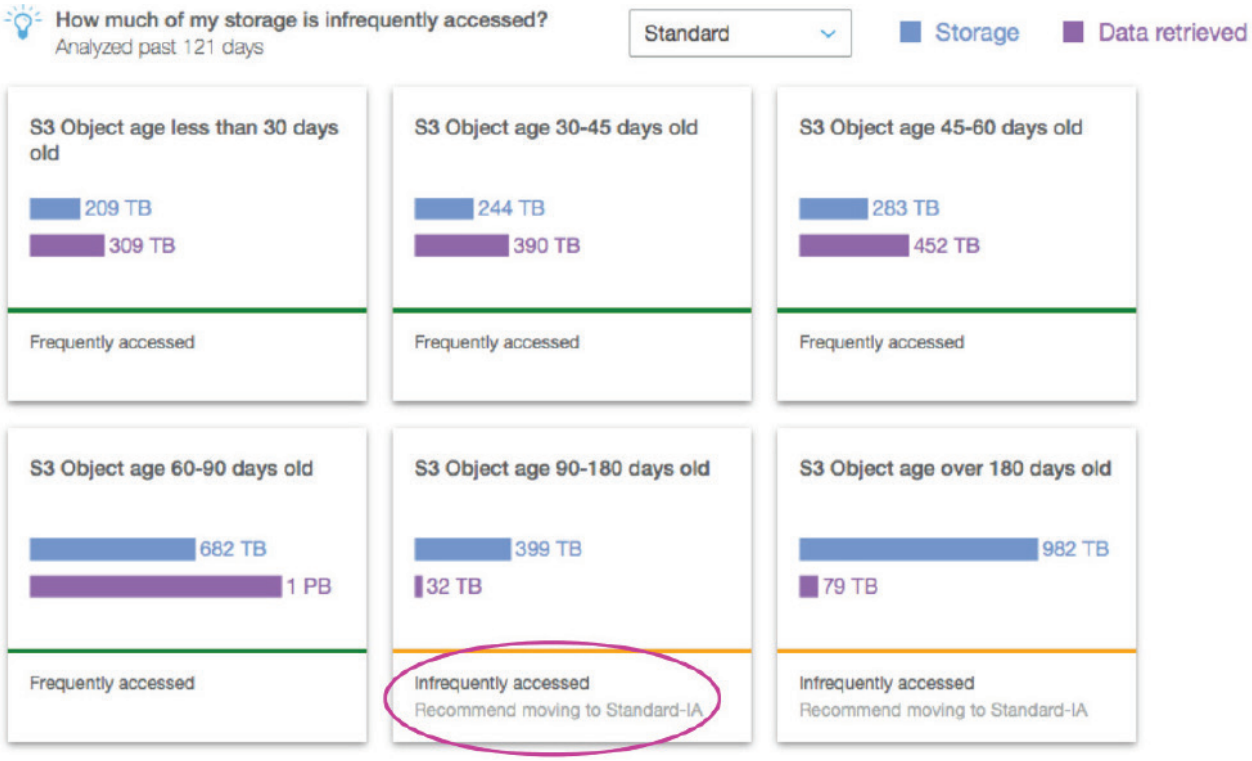

Figure 7. S3 bucket storage class analysis

Make sure to consider the new kid on the block: **S3 Intelligent Tiering.** For a small additional fee, Amazon will automatically detect access data patterns and, depending on the popularity of the object, will move them between two tiers, standard and infrequent access, all without performance impact or operational overhead. Less popular objects (i.e., ones that haven't been accessed for 30 consecutive days) will be moved to the infrequent access tier and brought back later should they be requested. Since there are no retrieval fees within this mechanism, objects can go back and forth forever. Real-world scenarios show up as having 20% savings, since the theory is partially lowered by the payment for such monitoring, yet is still too attractive to miss. Enable this technology by stating storage class "INTELLIGENT_TIERING" using S3 API or CLI or configure a lifecycle rule.

However, this doesn't work for everything. Objects that are less than 128 KB will never be transitioned to the infrequent access tier and thus will be billed at the usual rate for the frequent access tier. Another thing it won't work for is objects that live less than 30 days, as they will be billed for a minimum of 30 days regardless.

10. Befriend a lifecycle policy

Speaking of a lifecycle policy, it's a no-brainer whenever you operate with AWS S3. That said, implementing a lifecycle policy requires some learning and reading the documentation before enabling it on an infrastructure. The policy is a combination of rules for objects that have a well-defined use pattern, so you can:

- Use lifecycle rules to manage objects throughout their lifetime
- Automate the transition to tiered storage to lower costs
- Expire objects based on retention needs or service-level agreements (SLAs)

This is a very powerful tool when configured properly. Learn the difference between the S3 storage classes and start playing with lifecycle rules to better fit organization needs.

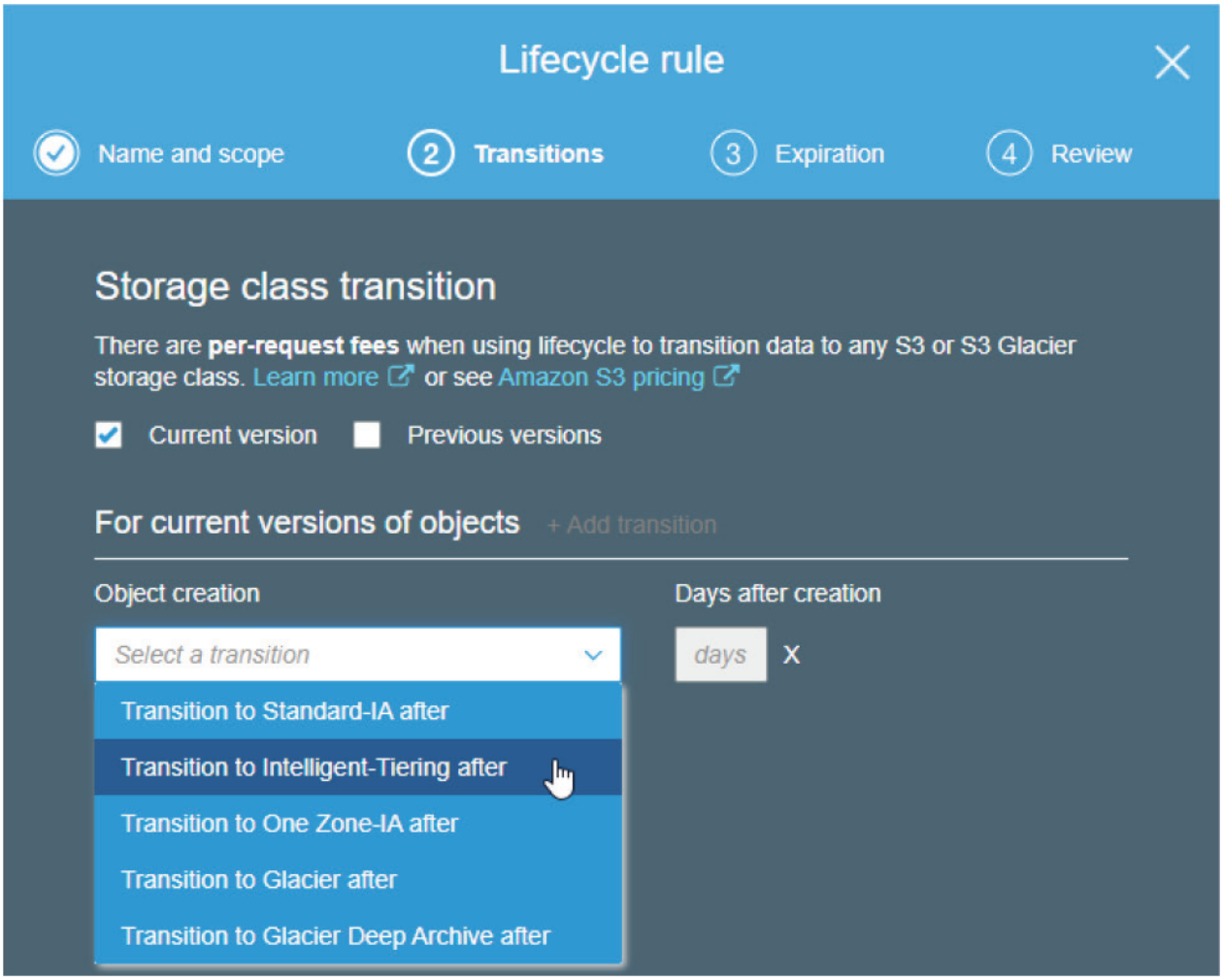

Figure 8. Lifecycle rule creation

11. Clean up multipart incomplete uploads

S3's **multipart upload feature,** which is enabled by default, accelerates the upload of large objects by splitting them up into logical parts that can be uploaded in parallel. The issue comes in when those uploads never finish for some reason. The actual incomplete data won't be visible in the bucket, nor will it be automatically deleted, so you won't notice a thing, except when the cost incurs in larger monthly bills. To prevent this, go to the "bucket management" settings, create a new lifecycle rule and enable the "clean up incomplete multipart uploads" option.

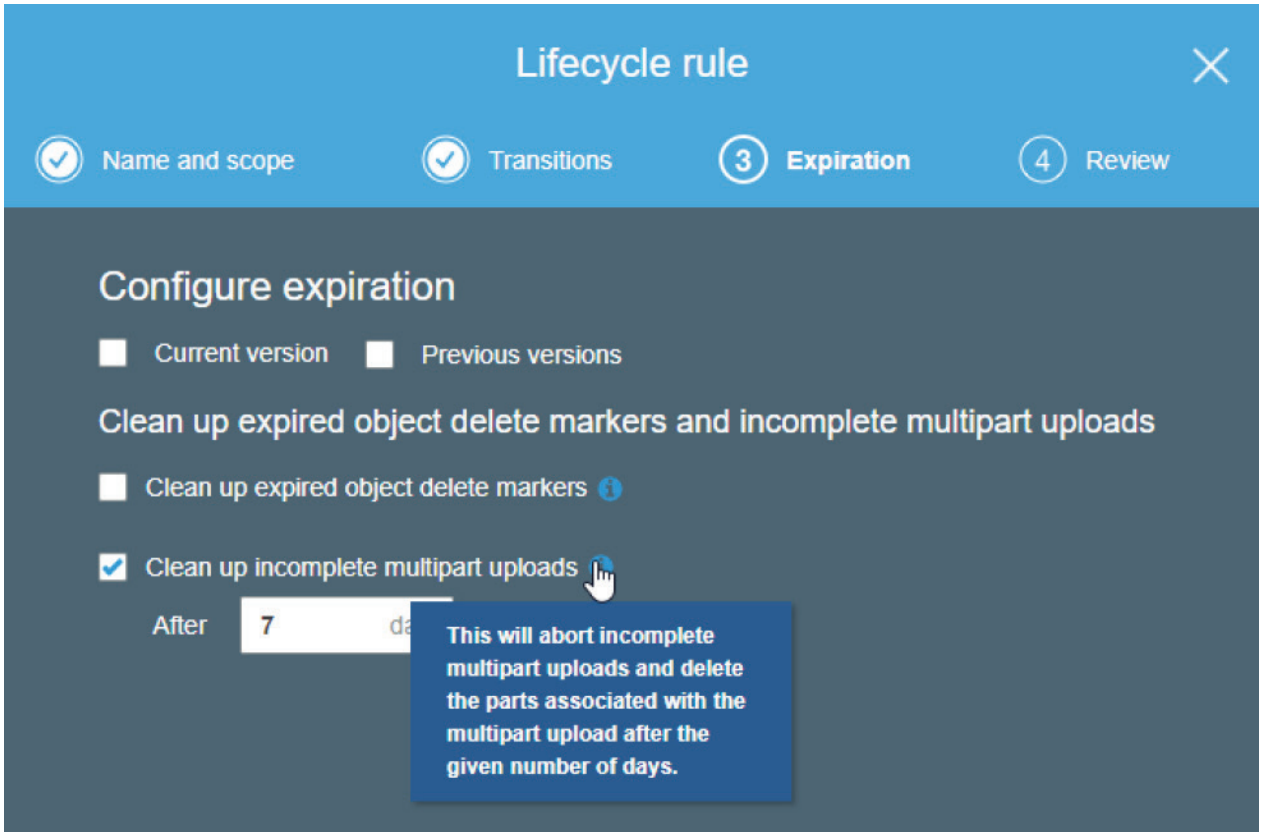

Figure 9. S3 bucket. Expiration configuration

Monitoring and management

Again, AWS tries hard to help companies on their cloud spend optimization journey. So, it's developed some tools, each targeting a certain aspect of price reduction. I've decided to highlight my personal preferences in hopes that they will be useful.

12. Automatically turn off unused instances and databases

Idle resources can be a major cost contributor to your AWS bill. Letting unused instances and databases sit idle means accruing charges for something that isn't used. For example, if you have a development environment that's only accessed during the day on weekdays, the best option would be not to run it 24/7. By programmatically stopping it at night and starting it again the following morning (hint: Have a look at the **Auto Instance Scheduler**) and at the end of the week, the costs could be cut almost in half. It's also a good strategy to enable **Amazon CloudWatch** alarms to automatically stop or terminate instances that have been idle for longer than a specified period.

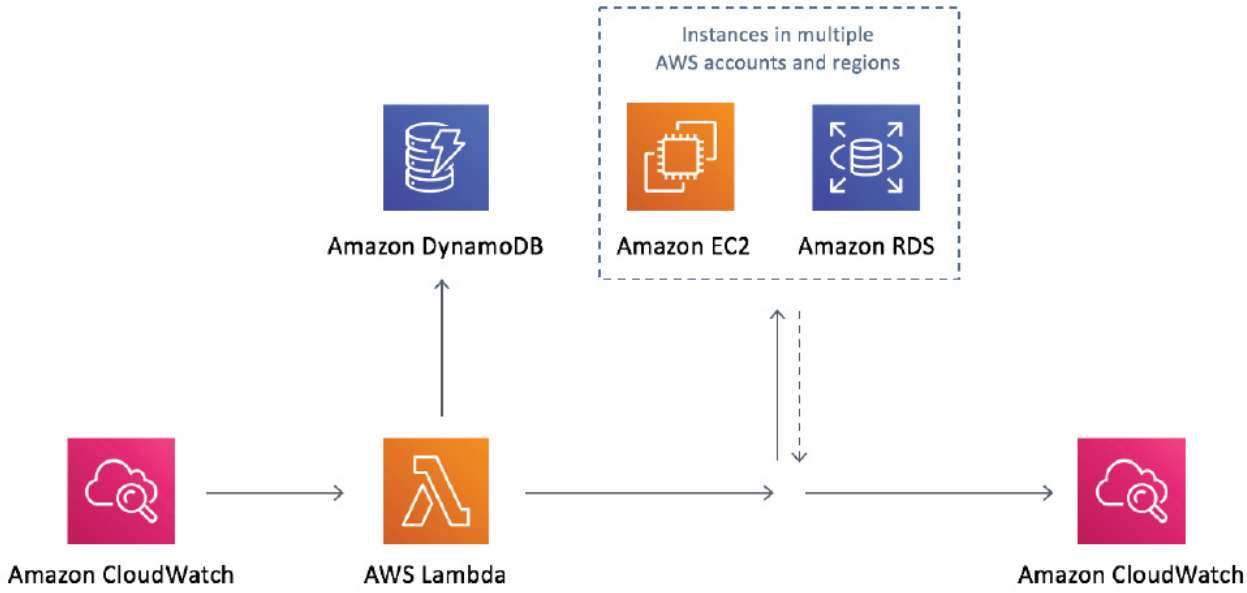

Figure 10. Amazon CloudWatch diagram

13. Bring the AWS Pricing Calculator into action

I've previously mentioned this tool in this white paper, as it allows to estimate account spending for over 40 AWS services. That said, sometimes it can be very helpful to decide between other competitive strategies too. For example, when switching from on-demand EC2 instances to RIs and Savings Plan offerings, you can check the discount each feature provides. With every value added to the table, it will show a resulted calculation and help to select the actual service.

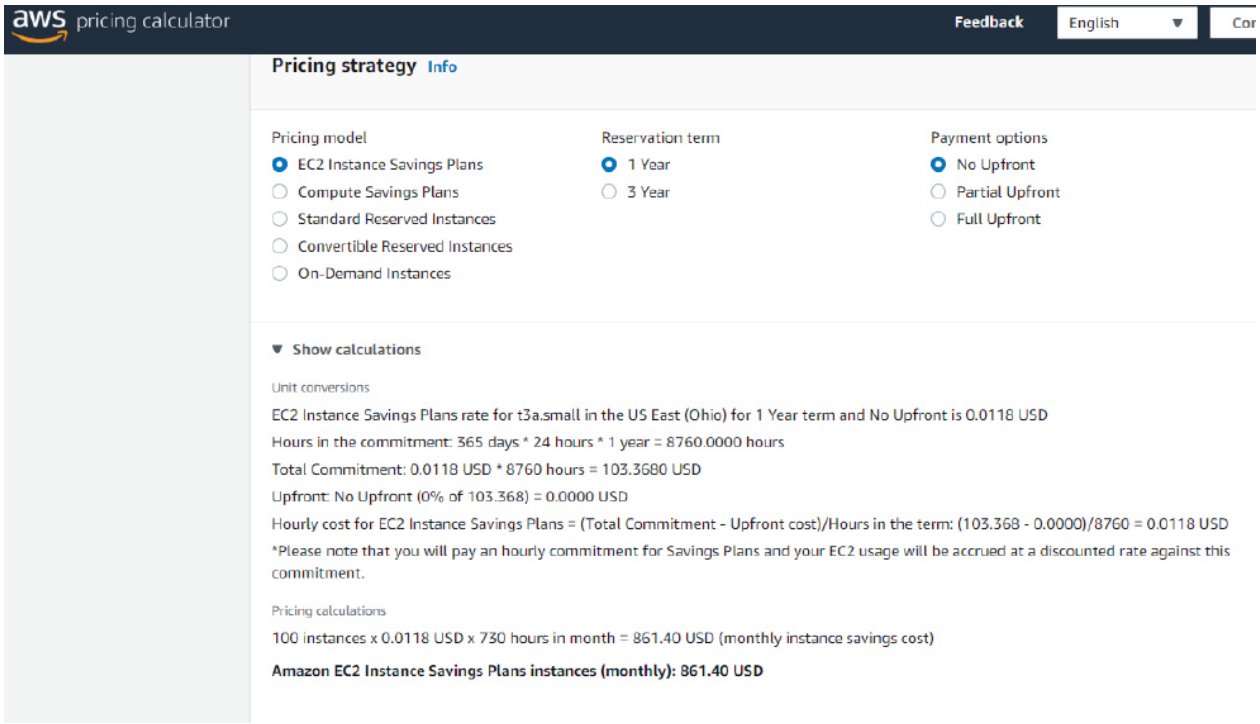

Figure 11. AWS pricing calculator cost prediction

14. Monitor consumption with standard tools

AWS Cost Explorer should be your go-to, since this instrument makes it very easy to visualize where all the money goes and address the services that consume the most. This can be found based on the analysis conducted, identifying interesting patterns and drilling down to the root cause.

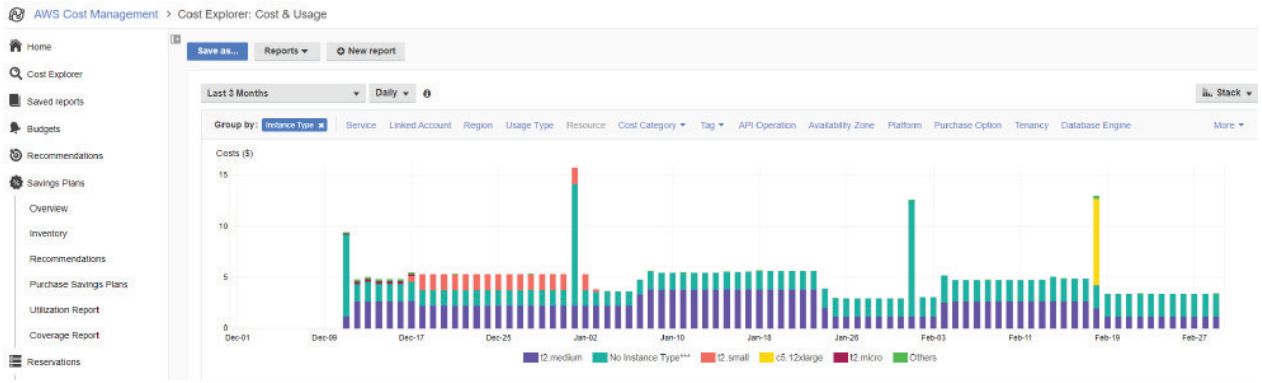

Figure 12. Cost Explorer. View instance types consumption.

AWS Compute Optimizer provides an overview of optimization opportunities for AWS resources based on the data that's been collected and analyzed for the account (or all the accounts under the master one).

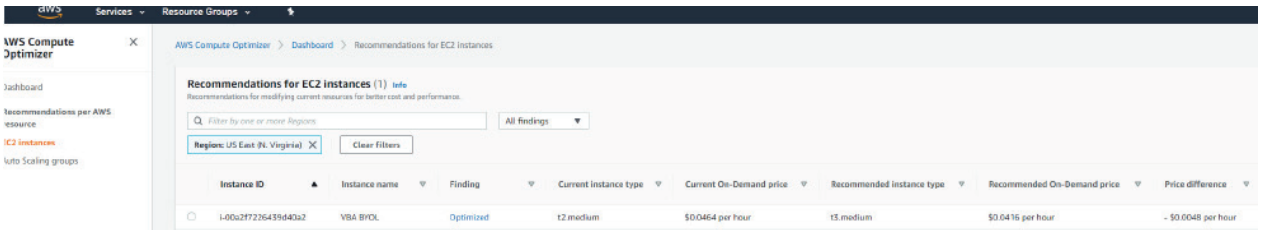

Figure 13. AWS Compute Optimizer. Recommendations for EC2 instances.

15. Consolidating users under a master account

If multiple people in organization need to operate AWS, consolidated billing will not only make payments easy but can also reach a certain threshold that once reached, an organization get a discount on consumed resources like S3. Some AWS tiers also reward higher usage with lower prices and discounted rates for purchasing instances in advance (like the RIs and Saving Plans from above). In addition, unused resources can be redistributed from one child account to another. Apply the previously mentioned cost optimization tools here and prevent organization chaos when operating numerous accounts.

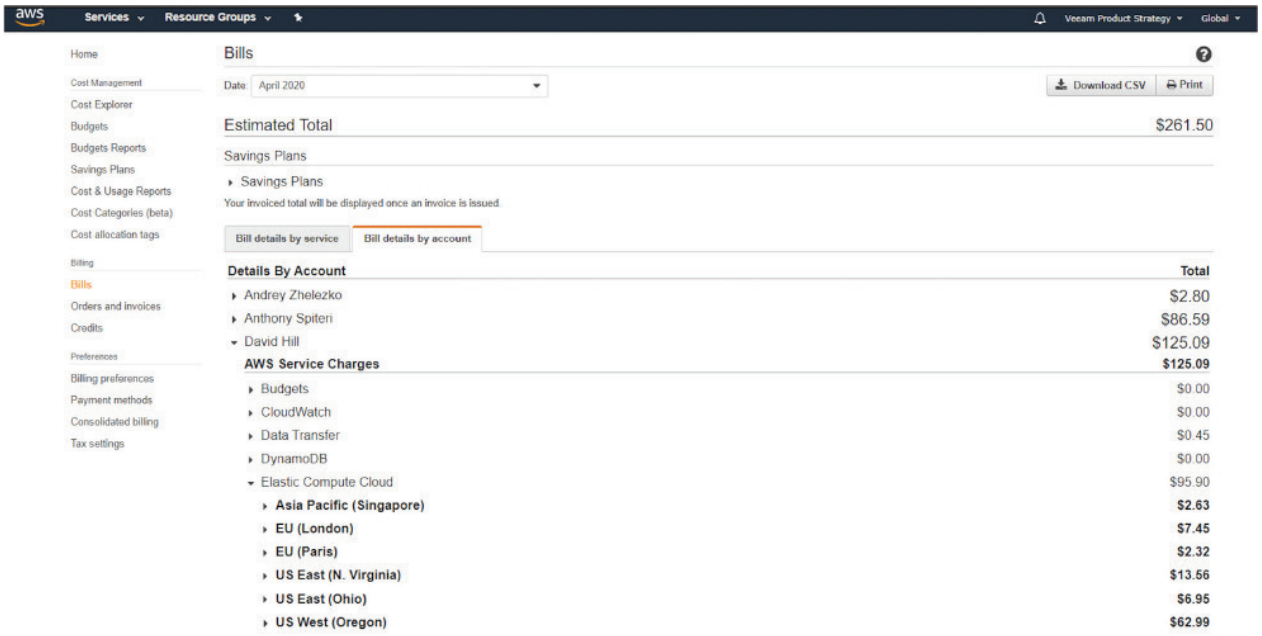

Figure 14. Master account console

16. Don't forget about networking costs

AWS networking deserves its own paper, but I feel like I should mention a few important tips here anyway without going too deep to keep things simple.

1. In case of massive traffic exchange between on-premises sites and AWS, the **Direct Connection** feature will help you get more consistent network experience, increase bandwidth throughput and secure connectivity.
2. Static content (i.e., images, videos, music) is better and more cheaply distributed through S3 and the CloudFront combination. With a great set of edge servers in different areas of the world, end users will get the data coming from the service cache that's located closer to them.
3. Analyze the traffic flow between different AWS services. Configure VPC endpoints so the traffic from VPC to other services like S3 would go through the endpoint, which bypasses the internet and public networks and results in a more secure and cheaper connection.
4. Don't forget about **inter-availability zone traffic,** as that will be another expense. Reconsider fault-tolerance architecture, it might not be needed for all services, or can be achieved through other technologies.

Bonus tip

Now that you're armed with this information on cost management tactics, Veeam is here to help. Veeam Backup *for AWS* can take backups of EC2 instances to ensure its protection and perform additional operations, such as restoring systems back to AWS or even on-premises, to move data around the infrastructure as required. Moreover, it will always show a predicted AWS cost amount caused by these operations, so any protection strategy can be adjusted accordingly (check figure 15).

Other free offerings from Veeam include Cloud Mobility, which allows to restore systems from on-premises to EC2, and Cloud Tier, which provides efficient backups in S3, as well as optionally making them immutable, so the data is safe from ransomware. Both Cloud Mobility and Cloud Tier are essential parts of Veeam Backup & Replication. Feel free to check them out:

Veeam Backup *for AWS:* https://www.veeam.com/aws-backup-recovery.html

Veeam Backup & Replication *Community Edition:* https://www.veeam.com/virtual-machine-backup-solution-free.html

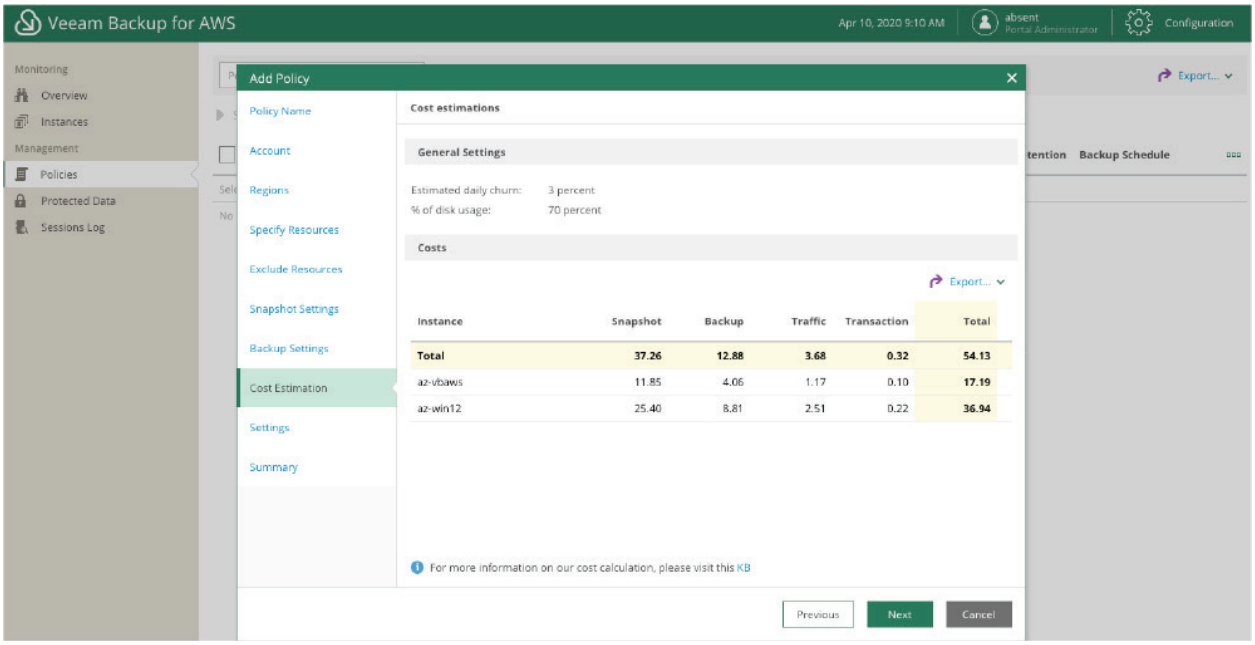

Figure 15. Veeam Backup for AWS. Cost Estimation.

Conclusion

While there is no magic button that can make all the optimization and cloud spending "just all right," there are many things administrators can do about it in reality! With the number of tools and resources available, make AWS cost optimization another part of your organization financial routine. Invest in learning settings, various tactics and native cost optimization services now, so it will pay itself later.

About the Author

Andrew Zhelezko, currently working as a technical product analyst in Veeam Product Strategy team, he is a certified IT professional with over a decade industry experience. Initially doing technical support for various solutions, he has got practical expertise, which helps him to speak the same language as Veeam community members. You can always find him presenting at different offline/online events, where he loves to solve the challenges associated with data protection. His motto is to help others realize the beauty and power of modren virtualization and cloud technologies.

About Veeam Software

Veeam® is the leader in Backup solutions that deliver Cloud Data Management™. Veeam provides a single platform for modernizing backup, accelerating hybrid cloud and securing your data. With 365,000+ customers worldwide, including 81% of the Fortune 500 and 66% of the Global 2,000, Veeam customer-satisfaction scores are the highest in the industry at 3.5x the average. Veeam's global ecosystem includes 70,000+ partners, including HPE, NetApp, Cisco and Lenovo as exclusive resellers. Headquartered in Baar, Switzerland, Veeam has offices in more than 30 countries. To learn more, visit https://www.veeam.com or follow Veeam on Twitter @veeam.

VEEAM

Veeam Backup *for AWS*

Easily recover from ANY cloud data loss scenario in minutes

- ✓ AWS-native
- ✓ Cost-effective
- ✓ Secure

Get started for FREE here