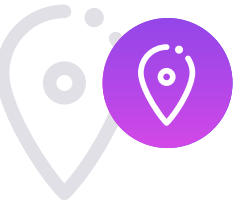


25 Innovative Tactics to Cloud Cost Optimization:

A DevOps Checklist





GeoLocation Configurations

Consider the AWS region that you're working from

There can be a significant difference in pricing between different regions. For example, Northern California is one of the more expensive AWS regions in the USA due to the cost of real estate and the higher personnel cost of managing data centers there. Oregon, therefore, makes a more cost effective alternative.



Network Configurations

Minimize data transfer costs

While it's important to consider the cost of the AWS region, you need to offset this against minimizing the distance of network traffic across AZs and regions. Charges are incurred when data is transferred out from AWS services to the internet, or between AWS regions or AZs. So if you're based in Melbourne, it might not make sense to have your cloud infrastructure in Oregon, even though costs are less there than in Sydney, because of the long network distances this would create.

Get closer to your customers with a CDN

When EC2 instances are behind a load balancer it's difficult to narrow down the cost of the data transfers and the environment they occurred in. Content distribution services and caching services available through CloudFront can reduce this cost by caching any image, video, or static web content at AWS edge locations which can dramatically cut down the distance data has to travel to your users. CloudFront isn't the only option: as a leading CDN innovator with 20 years of experience in content delivery, [GlobalDots offers cost-efficient and flexible CDN](#) solutions from the top providers worldwide.

The screenshot shows the AWS CloudFront console. The left sidebar contains navigation options: Distributions, What's New, Reports & Analytics, Cache Statistics, Monitoring and Alarms, Popular Objects, and Top Referrers. The main content area is titled 'CloudFront Distributions' and includes a 'Create Distribution' button, 'Distribution Settings', 'Delete', 'Enable', and 'Disable' buttons. Below this is a table with columns: Delivery Method, ID, Domain Name, Comment, Origin, CNAMEs, Status, State, and Last Modified. A single distribution is listed with ID 'E3R6PUFQ6PHGH', Domain Name 'dys62g3nrogi.clo', Origin 'qa.conjuror.it', Status 'Deployed', and State 'Enabled'. The last modified date is '2018-02-14 03:02'.

Configuring a CDN on CloudFront

Delete idle Load Balancers

You can identify those load balancers that have had relatively little use by using the Trusted Advisor Idle [Load Balancers check](#). This provides a report of load balancers that have a request count of less than 100 over the past 7 days. To eliminate the costs of infrequently used load balancers you can delete them by following the [instructions here in step 8](#).

Identify and delete unused ELBs

Consolidate Network components where possible

Many customers use a load balancer per environment, in some cases, there is an option to consolidate them and reduce the number that is needed.

Clean up underutilized network resources

Similar to load balancers, there are other resources that include anything from Route 53, to elastic IP resources that were created for development and testing purposes, but are no longer used. These idle resources continue to incur cost irrespective of whether they are used or not. They can be identified by GlobalDots for discovering unused resources or by using AWS Trusted Advisor, which can report on network resources that are underutilized and can be deleted.

| Type | Name | Account | Region | Created at | Unused since | Monthly cost | Permission |
|----------------|------|---------|-----------|--------------------|--------------------|--------------|------------|
| EBS (standard) | ... | ... | us-west-2 | 2013/12/9 22:05:47 | 2022/4/11 20:17:34 | \$5... | ... |
| EBS (standard) | ... | ... | us-west-2 | 2013/12/9 22:05:46 | 2022/4/11 20:17:34 | \$1... | ... |
| EIP | ... | ... | us-west-1 | 2022/2/1 15:26:25 | 2022/4/11 20:17:30 | \$3.65... | ... |
| EIP | ... | ... | us-west-1 | 2022/2/1 15:26:25 | 2022/4/11 20:17:29 | \$3.65... | ... |
| EIP | ... | ... | us-west-1 | 2022/2/1 15:26:25 | 2022/4/11 20:17:30 | \$3.65... | ... |
| EIP | ... | ... | us-west-1 | 2022/2/1 15:26:25 | 2022/4/11 20:17:30 | \$3.65... | ... |

GlobalDots' platform to discover unused resources

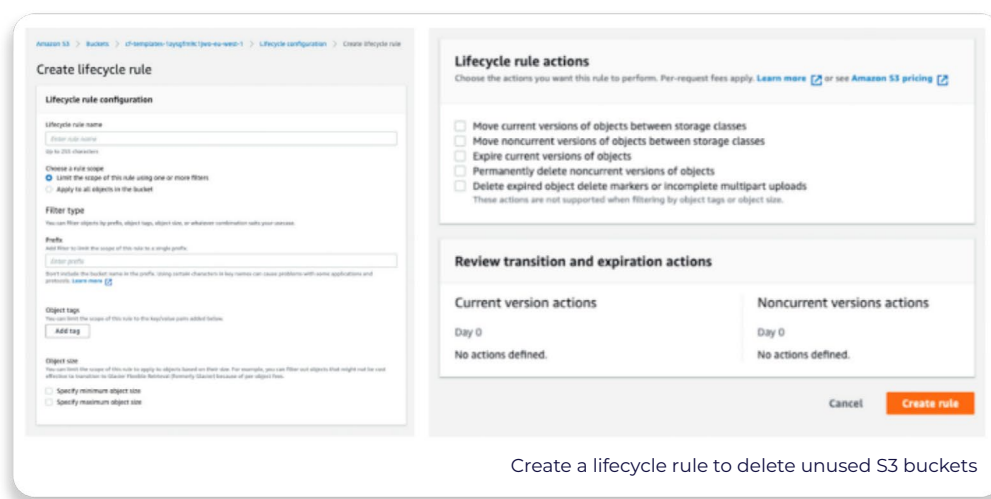


Storage

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](#).



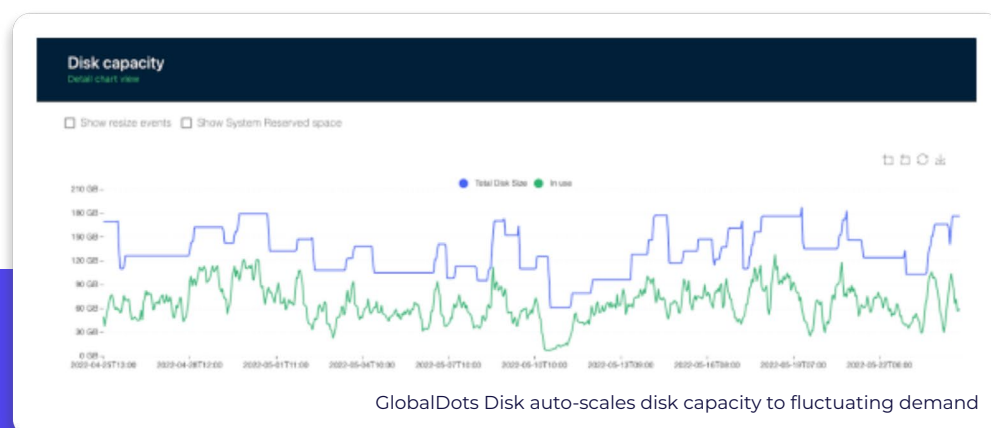
Create a lifecycle rule to delete unused S3 buckets

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 overprovision 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 utilizing the solutions available on GlobalDots, 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.

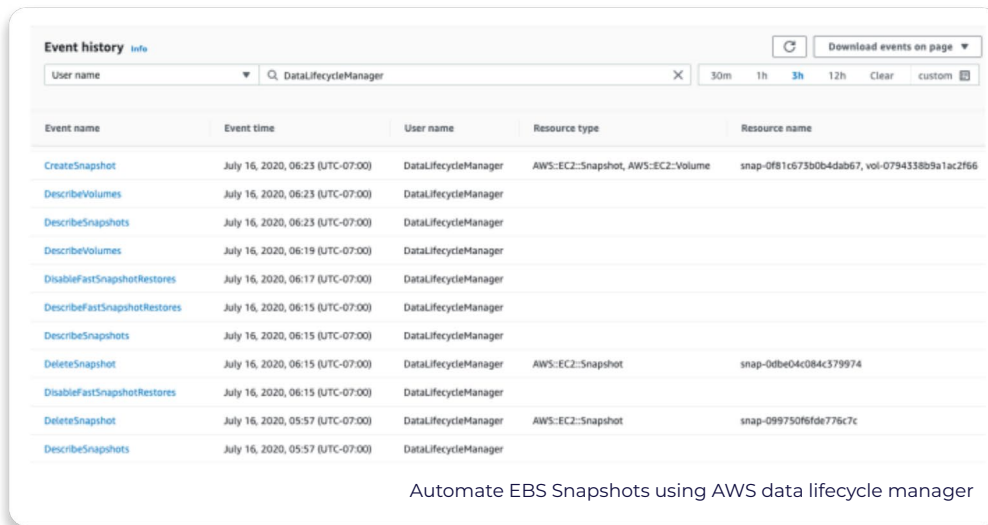


Automate the management of Snapshots

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.



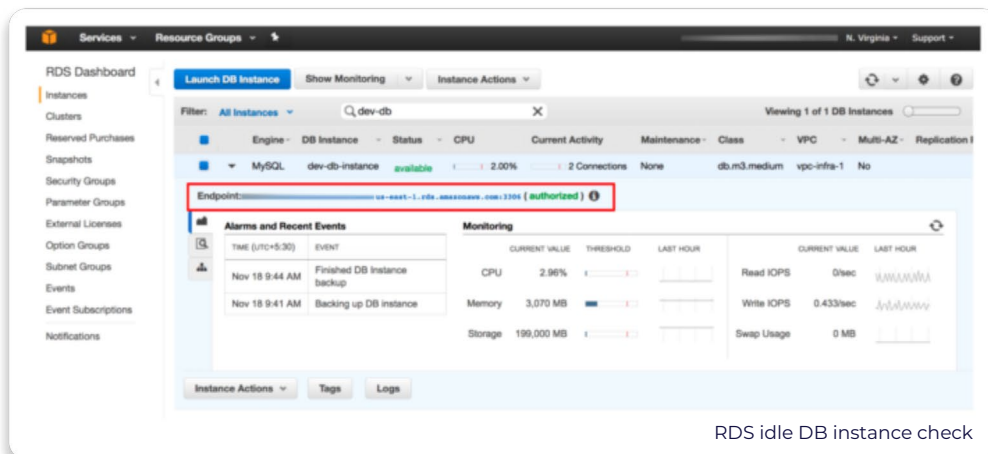
Automate EBS Snapshots using AWS data lifecycle manager



Databases

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.



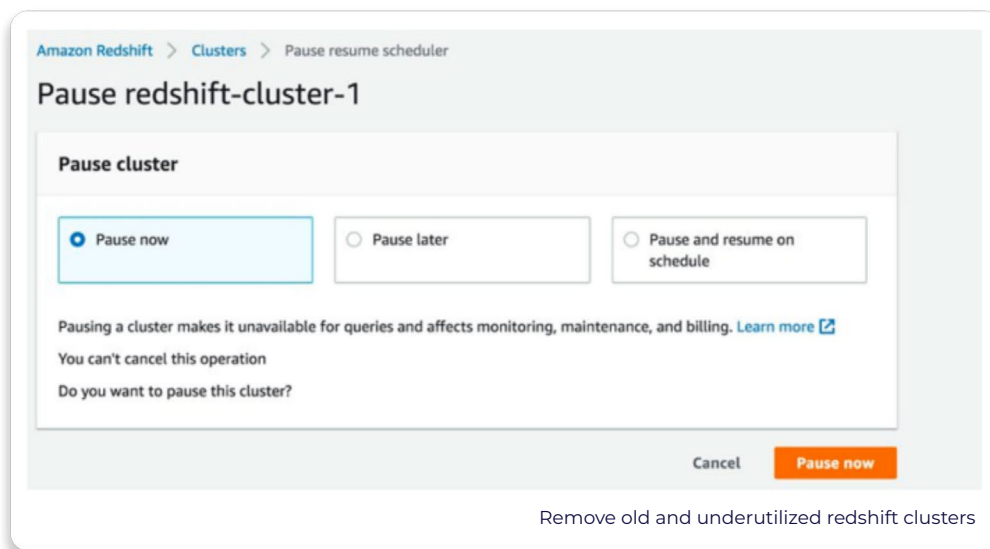
RDS idle DB instance check

Stop paying for idle Redshift clusters

It's easy to overprovision Redshift clusters or leave them running during the evening, 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.

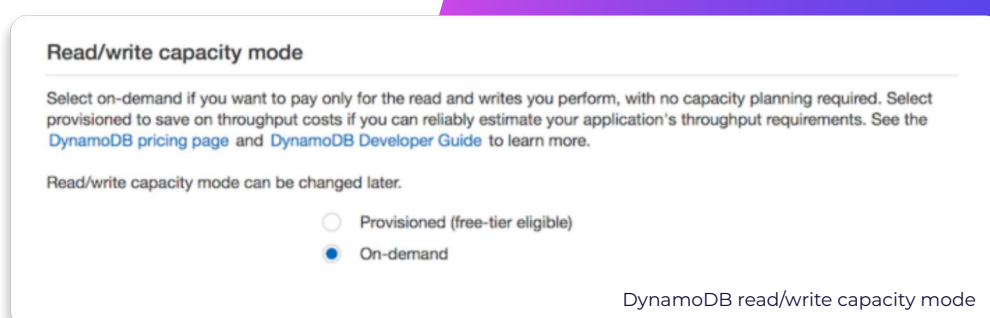
Remove old Redshift clusters

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.



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.

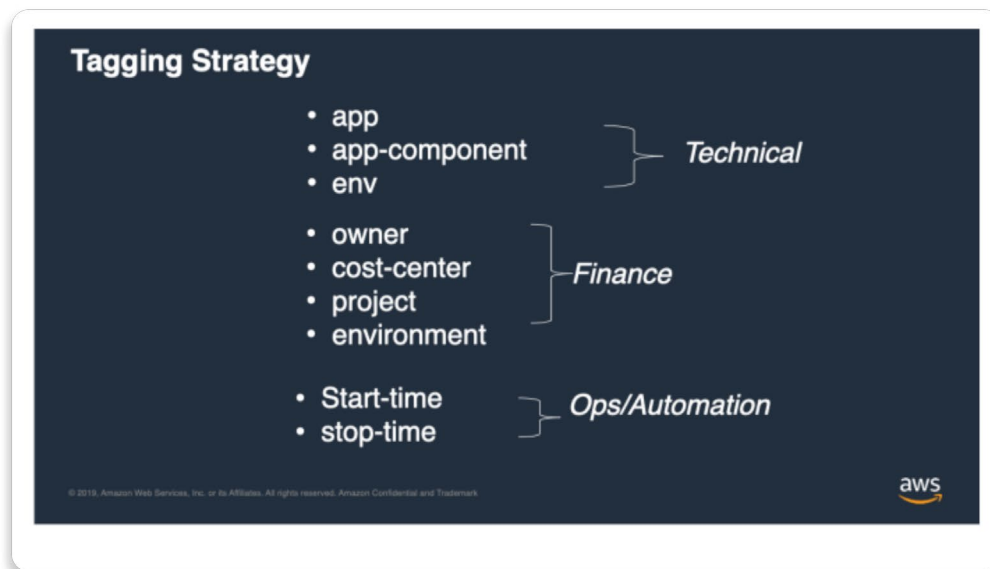




Compute

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 to 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.



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 used to refine AWS budget criteria.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

| Key (127 characters maximum) | Value (255 characters maximum) | Instances | Volumes |
|------------------------------|--------------------------------|-------------------------------------|-------------------------------------|
| Name | leBlogTestInstance | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

[Add another tag](#) (Up to 50 tags (maximum))

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.

The screenshot displays the 'Resource Optimization' section in AWS Cost Explorer. It shows 3 optimization opportunities with an estimated monthly savings of \$110 and 50.00% savings. Below this, a table lists three recommendations for modifying instances to save costs.

| Recommendation | Instance ID | Account ID | Tag(s) | CPU (%) | Monthly estimated savings | |
|-----------------|-----------------|----------------------|--------|---------|---------------------------|----------------------|
| Modify instance | i-0b18d304a1... | AWS Insights Demo... | 3 + | 6.6% | \$72 | View |
| Modify instance | i-0196e32825... | AWS Insights Demo... | 2 + | 4.0% | \$33 | View |
| Modify instance | i-0a909f442... | AWS Insights Demo... | 2 + | 7.5% | \$4 | View |

Additional filters on the right include 'Show recommendations for' (Idle instances, Underutilized instances), 'Additional Filters' (Linked Account, Region, Tag), and a 'Download CSV' button.

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.

The screenshot shows the 'Resource inventory' page in AWS Config. The 'Resources' dropdown is set to 'AutoScaling: AutoScalingGroup'. The 'Look up' button is highlighted. Below the search filters, a table lists the resource configuration, with the 'Manage resource' link circled in orange.

| Resource type | Config timeline | Compliance | Manage resource |
|------------------------------|---------------------|------------|---------------------------------|
| AutoScaling AutoScalingGroup | AS- APP1 - AS group | -- | Manage resource |

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. Based on this information [Operations Conductor](#) can be used to rightsize instances or change instance types.

49.81%
Estimated savings (%)

On these instances could help you save an estimated \$58.69 monthly

[Download CSV](#)

| Tag(s) | CPU (%) | Monthly estimated savings | |
|--------|---------|---------------------------|----------------------|
| 1 | 1.3% | \$17.08 | View |
| 2 | 2.6% | \$8.32 | View |
| 6 | 1.8% | \$8.32 | View |
| 3 | 1.7% | \$8.32 | View |

Recommendation parameters

Generate recommendations

- Within the same instance families
- Across instance families

Resource optimization types

- Idle instances
- Underutilized instances

Additional Filters

Linked Account [Include all](#)

Region [Include all](#)

Tag [Include all](#)

Advanced options

- Include Savings Plans and Reserved Instances

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.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option Request Spot instances

Current price

| Availability Zone | Current price |
|-------------------|---------------|
| us-west-2a | \$0.2722 |
| us-west-2b | \$0.306 |
| us-west-2c | \$0.2763 |
| us-west-2d | \$0.3117 |

Maximum price

Persistent request Persistent request

Launch group

Request valid from [Edit](#)

Select Spot Instances when configuring instance 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.

The screenshot shows the AWS Cost Management console's Savings Plans Purchase Recommendations page. The interface includes a navigation sidebar on the left with options like Home, Cost Explorer, Saved reports, Budgets, Recommendations, Savings Plans, Reservations, and Utilization Report. The main content area displays 'Recommendation options' with filters for Savings Plans type (Compute selected), term (1-year selected), payment option (No upfront selected), and based-on period (30 days selected). Below this, a recommendation is shown for a Compute Savings Plan at a commitment of \$0.10/hour. A summary states that purchasing this plan could save an estimated \$29 monthly. A comparison table shows the 'Before recommended purchase' state with a monthly on-demand spend of \$177 (\$0.24/hour) and the 'After recommended purchase' state with an estimated monthly spend of \$148 (\$0.20/hour) and an estimated monthly savings of \$29 (\$0.04/hour).

| Before recommended purchase | After recommended purchase (based on your past 30 days of usage) | |
|---|---|--|
| Monthly On-Demand spend | Estimated monthly spend | Estimated monthly savings |
| \$177 (\$0.24/hour) | \$148 (\$0.20/hour) | \$29 (\$0.04/hour) |
| Your estimated On-Demand spend based on your usage over the past 30 days (including all active Savings Plans) | Your recommended \$0.10/hour Savings Plans commitment + an average \$0.10/hour of On-Demand spend | 16% monthly savings over On-Demand \$177 - \$148 = \$29 |

Savings Plans recommended options

Migrate to Graviton Instances (where you can)

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.

The screenshot shows the AWS Compute Optimizer 'Recommendations for EC2 instances' page. It features a 'CPU architecture preference' dropdown set to 'Current, Graviton (aws- or...)'. Below this, a table lists recommendations for three instances. The first instance is 'Under-provisioned' and the other two are 'Over-provisioned'. The table columns include Instance ID, Finding info, Current instance type, Current On-Demand price, Recommended instance type, Recommended On-Demand price, Migration effort, and Inferred workload types.

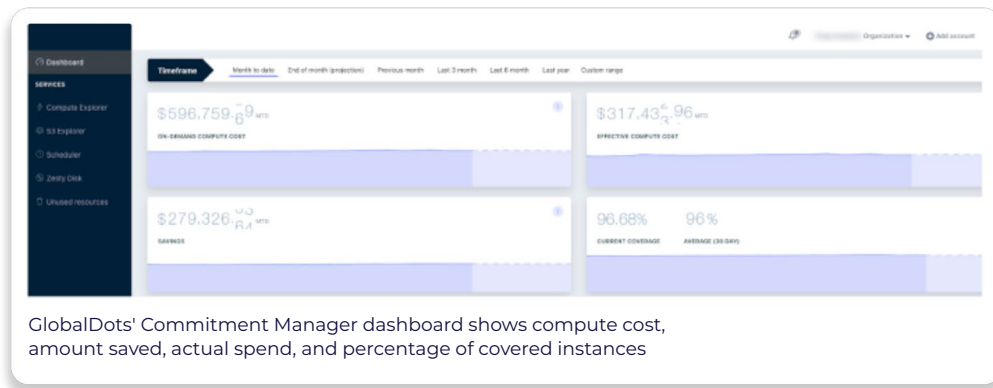
| Instance ID | Finding info | Current instance type | Current On-Demand price | Recommended instance type | Recommended On-Demand price | Migration effort | Inferred workload types |
|---------------------|-------------------|-----------------------|-------------------------|---------------------------|-----------------------------|------------------|-------------------------|
| i-0b5ec1bb9daabf0f5 | Under-provisioned | i5.large | \$0.1260 per hour | r6g.large | \$0.1008 per hour | Low | Apache Hadoop |
| i-033868420bdc7d29a | Over-provisioned | c5.2xlarge | \$0.3400 per hour | r6g.large | \$0.1008 per hour | Medium | - |
| i-0bc9a76f2ed1a5e75 | Over-provisioned | i5.2xlarge | \$0.6240 per hour | i5.large | \$0.3120 per hour | Very Low | - |

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 [GlobalDots' 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 on average 50% off their EC2 workload.



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!

About GlobalDots

GlobalDots is a 20-year global leader in cloud innovation, connecting over 1,000 global businesses with the latest cloud and web technologies, such as Security, Web Performance, DevOps & Cloud Management, Corporate IT, and advanced AI/ML models. Led by a team of seasoned engineers and architects, GlobalDots offers easy end-to-end innovation adoption, from consulting to ongoing professional services, proactively introducing newer and better solutions to support businesses in maintaining a scalable, up-to-date technology posture in a quickly-changing world.

Trusted by

