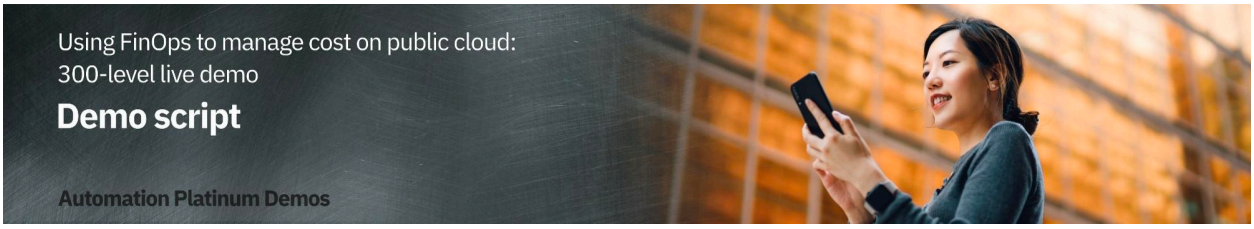
Using FinOps to manage cost on public cloud

300-level live demo script



Introduction

Gen-Z Bank is a financial services platform catering primarily to millennials. The bank has just wrapped up its cloud migration initiative. All mission-critical workloads were successfully lifted-and-shifted to AWS, Azure, or Google Cloud Platform (GCP). However, this transition has resulted in soaring cloud costs, and the bank's monthly cloud bills have been getting regular executive attention from the financial team. The CIO has formed a new FinOps group to address the issue.

In this demo, I'll show you how the IBM Turbonomic platform helps the FinOps team at Gen-Z Bank control their cloud costs.

The demo is divided into four sections:

1) Getting a global view of multi-cloud costs and savings opportunities
2) Rightsizing compute instances
3) Optimizing storage volumes
4) Leveraging cloud provider discounts

Let's start with some basics. In the public cloud pricing model, the total spend is calculated by multiplying the resources you're consuming by the rate you're paying. Therefore, cloud cost savings can be realized by either reducing consumption or lowering your rate.

Let's look at how Turbonomic can help with both.

1 - Getting a global view of multi-cloud costs and savings opportunities

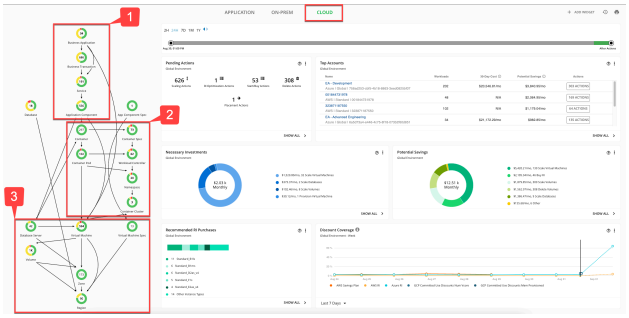
1.1 - Get visibility into the cloud deployment

Narration
Before the bank can do any cost optimization, they must get a global view into their workloads. Turbonomic automatically creates a service graph that models the dynamic relationships of the workloads and their dependent infrastructure layers. This provides a full stack view from the application down to all the dependent cloud resources.

Let's look at the cloud view, which combines all our accounts on public cloud providers (AWS, Azure, and GCP).

Action 1.1.1

- After logging in to the **Turbonomic** instance, select the **CLOUD** view and point out the types of entities, as highlighted in the screenshot and narration below.



Narration
There are three types of entities:

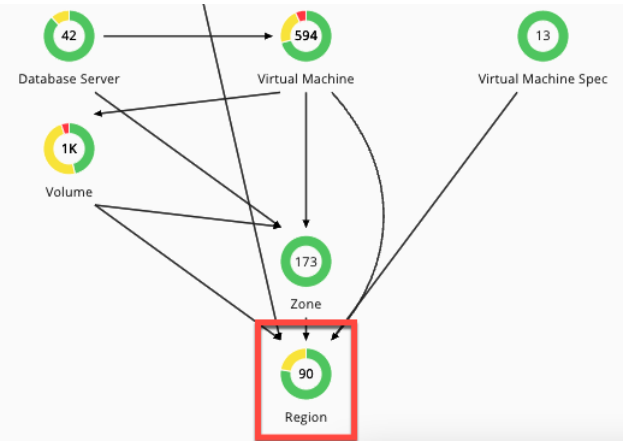
- Entities that represent the *application workload* (1)
- Entities that represent the *container platform* (2)
- Entities related to the *cloud services* (i.e., compute instances, storage volumes, Platform-as-a-Service (PaaS) services) (3)

Since the bank is dealing with the cloud, let's start by examining the 'Region' entity. A Region is a logical grouping of physical data centers (Zones) that are in close geographic proximity. A 'Region' is comprised of one or more of these Zones. A 'Zone' represents an Availability Zone in your public cloud account or subscription. In the language of public clouds, a Zone is a data center.

The current view shows there are 90 Regions and 173 Availability Zones.

Action 1.1.2

- Select the **Region** entity.

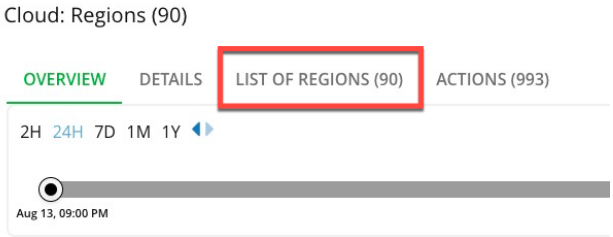


Narration

Let's get a high-level understanding of the geographic spread of the bank's workloads across these regions.

Action 1.1.3

- Click **LIST OF REGIONS**.



Narration

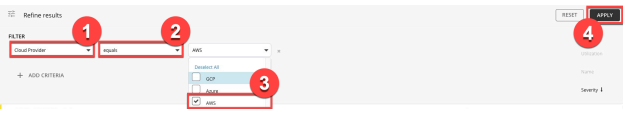
This dashboard lists out the various regions from the cloud providers (AWS, Azure, and GCP), along with the workloads running in a given region. Let's take a look at the AWS workloads.

Action 1.1.4

- Click the **FILTER** icon.

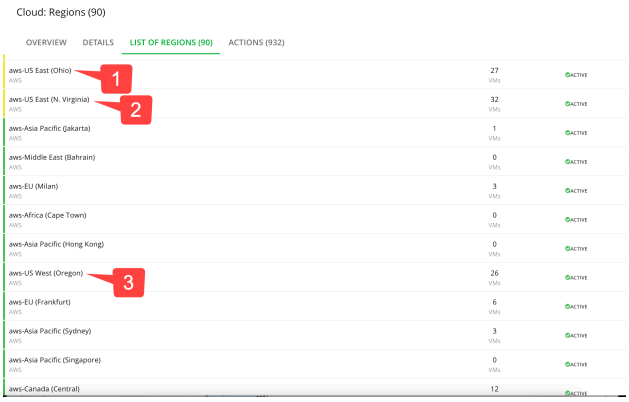
Action 1.1.5

- Set the filter to: **Cloud Provider** (1), **equals** (2), and **AWS** (3). Click **APPLY** (4).



Action 1.1.6

- Point out the spread of the workloads across the various cloud regions, as highlighted in the screenshot and narration below.



Narration

We see that most workloads are concentrated primarily in three regions: US East Ohio (1), US East Northern Virginia (2), and US West Oregon (3). There are also several regions that contain no (or minimal) running workloads. This information is important because it can help the bank make decisions about consolidating workloads in a given region and purchasing pre-paid volume discounts, such as regional or zonal reserved instances (RIs), for maximum financial benefit.

Action 1.1.7

- Click **X** to exit the **Cloud: Regions** view.

Narration

Now that we’re back in the default global cloud view, let’s examine the various summary dashboards.

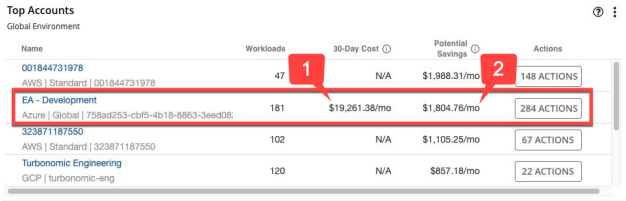
1.2 - Identify the highest spending accounts

Narration
Before the bank can make any cloud cost optimization decisions, they must understand several aspects of the cloud deployment. Turbonomic dynamically synthesizes a set of useful executive dashboards that help the bank better focus their cost optimization efforts. Doing this manually and continuously would be unscalable and inefficient.

Let's take a look at some of these dashboards.

Action 1.2.1

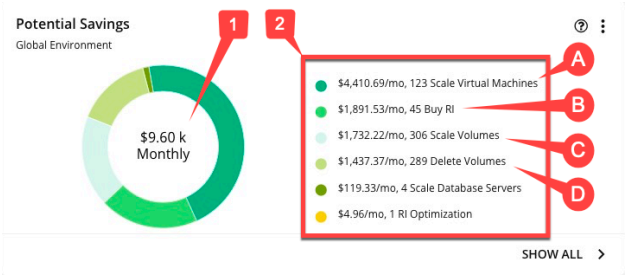
- In the **EA - Development** row of the **Top Accounts** dashboard, point out the values under the **30-Day Cost** (1) and **Potential Savings** (2) columns.
- **NOTE**: Since this is a live application, you may not see the exact same view (this caveat applies to the rest of the demo). In any case, select the row that has both the **30-Day Cost** and **Potential Savings** columns populated.



Narration
The 'Top Accounts' dashboard lists the bank's most utilized public cloud accounts and their monthly costs. Taking the recommended actions can result in a large potential savings (10% in the case of this demo, but it could easily be substantially more). Saving 10-30% in a large-scale cloud environment can yield huge savings.

Action 1.2.2

- In the **Potential Savings** dashboard, point out the **total monthly potential savings** (inside the donut chart) (1) and the **breakdown list** (2).
- **NOTE**: The numbers may be small in this case, as this is a demo environment. The potential savings in a real customer environment are significant.



Narration
The number inside the donut chart provides the bank an estimate of the total potential monthly savings. In the breakdown list on the right, we see that the bulk of these savings can be realized by: A) rightsizing compute instances, B) buying reserved instances, C) rightsizing storage capacity, and D) eliminating waste.

Let’s take a closer look at each of these savings opportunities.

2 - Rightsizing compute instances

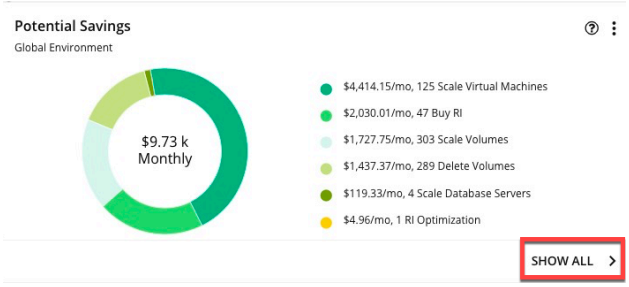
2.1 - Analyze a rightsizing recommendation

Narration
Compute resources are often the biggest line item on a cloud bill – they can total as much as 50% of the total bill. Hence, it makes sense that the bank starts their FinOps initiative by optimizing their compute resources.

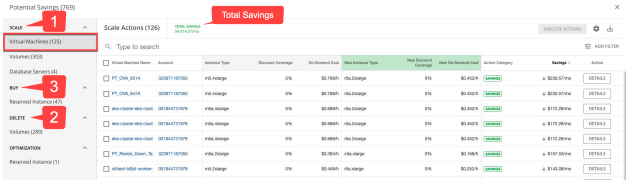
In the language of Turbonomic, optimizing compute resources and scaling virtual machines mean the same thing. Lift-and-shift cloud migration strategies typically employ the legacy on-prem model of grossly oversizing their resources. The elastic scaling of the cloud, however, now enables the ability to continuously balance capacity and demand in real time, yielding higher cost efficiency.

Let's start by exploring the potential savings that can be realized by scaling virtual machines.

Action 2.1.1
- Click **SHOW ALL** on the **Potential Savings** dashboard.



Action 2.1.2
- Select **Virtual Machines** under the **SCALE** category, and point out the details on the **Potential Savings** dashboard, as highlighted in the screenshot and narration below.



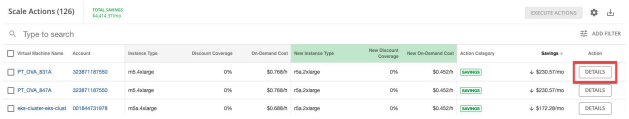
Narration
This dashboard categorizes the specific actions the bank can take to realize their potential savings. Note that the savings actions are classified into three categories:

- *Scale* (1): Identifies cost reductions that can be achieved by addressing overprovisioning of compute instances
- *Delete* (2): Provides recommendations to get rid of unattached storage (we'll talk more about storage in the next section of this demo)
- *Buy* (3): Recommends the purchase of highly discounted reserved instances to obtain significantly discounted pricing (we'll talk more about RIs in the last section of this demo)

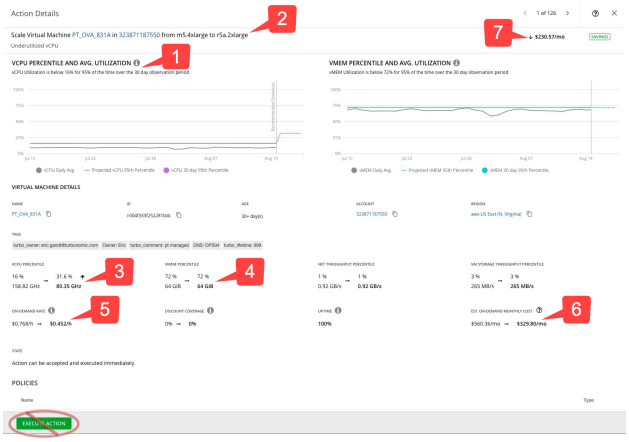
In this view, the 'Total Savings' represents the cost reduction the bank can gain by accepting and executing all the recommended 'Scale Virtual Machines' actions.

Let's examine the details behind the first virtual machine scaling action.

Action 2.1.3
- Click **DETAILS** on the first row.



Action 2.1.4
- Point out the **Action Details**, as highlighted in the screenshot and narration below.
- **NOTE**: Do **NOT** click the **EXECUTE ACTION** button, as this is a shared demo environment.



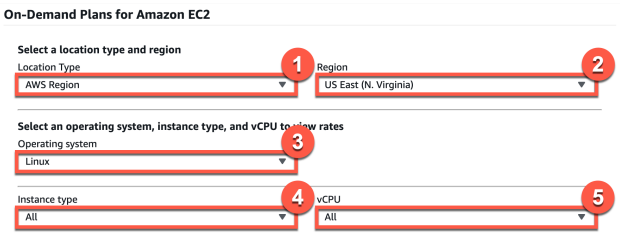
Narration
The current instance [*m5.4xlarge EC2*] is highly underutilized based on the 30-day observation period (1). Turbonomic makes a recommendation to scale down to a different instance [*r5a.2xlarge*] (2). If this action is taken, the instance utilization will increase from 16% to 31.6% (3), and there will be no change in the memory utilization (4). Making the recommended change will decrease the on-demand price (5) and on-demand monthly cost (6). This would result in an approximate monthly savings of $230.57 for this one instance (7). [*Since this is a live application, your savings numbers and percentages may vary slightly.*]

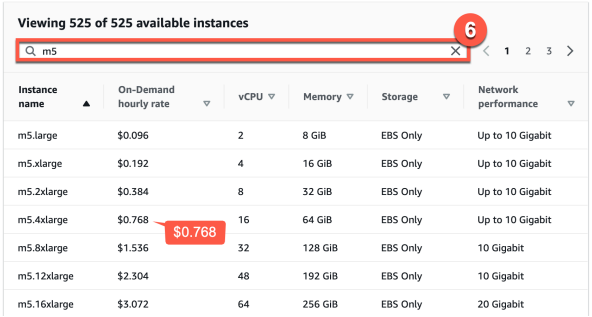
To understand how Turbonomic derived the on-demand price, on-demand monthly cost, and savings, let's use Amazon's AWS EC2 On-Demand Pricing Calculator.

Action 2.1.5

- Click here to get to Amazon's **AWS EC2 On-Demand Pricing Calculator** (this will open in a new tab). Fill out the **On-Demand Plans for Amazon EC2** form with the following data:

1. **Location Type**: AWS Region
2. **Region**: US East (N. Virginia)
3. **Operating system**: Linux
4. **Instance type**: All
5. **vCPU**: All
6. **Search bar**: m5

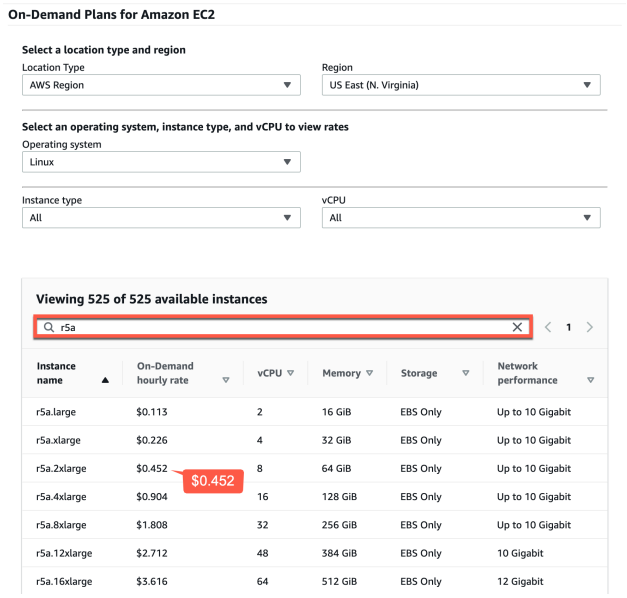




Narration
We can see that the current instance [*m5.4xlarge*] has an on-demand hourly rate of $0.768.

Action 2.1.6

- In the search bar of the **On-Demand Plans for Amazon EC2** form, enter **r5a**.

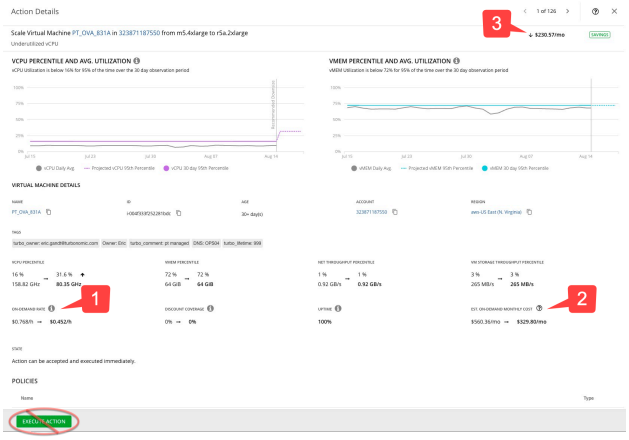


Narration

We can see that the recommended instance [*r5a.2xlarge*] has an on-demand hourly rate of $0.452.

Action 2.1.7

- Return to the **Action Details** panel.
- **NOTE**: Since this is a shared demo environment, the **VMEM PERCENTILE AND AVG. UTILIZATION** graph on the right may not appear when you run the demo. Use the screenshot below to understand the analytics generated by the Turbonomic platform.

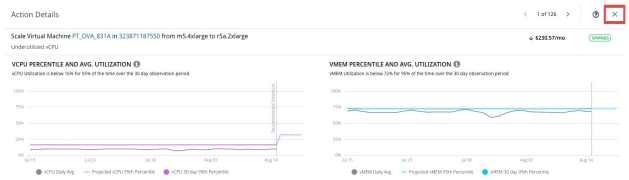


Narration
Turbonomic has a direct integration with AWS to find the current on-demand rates and calculate cost savings. As we return to the Action Details panel, we can see that Turbonomic has already gathered the on-demand hourly rates for the current instance *m5.4xlarge* ($0.768) and the recommended instance *r5a.2xlarge* ($0.452) for our convenience and ease of reference (1). As a result, the estimated on-demand monthly cost for the current instance is $560.36 per month and for the recommended instance is $329.80 per month (2).

Based on this optimization, the bank can save $230.57 per month (3), which is approximately 42% of the cost of the current instance. Note that the potential savings are on the on-demand price. Later in the demo, we will explore how this price can be further optimized by leveraging programmatic discounts like reservations.

The ‘Execute Action’ button demonstrates the ability of the Turbonomic platform to execute the recommendations natively from the platform. This provides the IT Ops teams the convenience to act immediately and realize cost savings instantaneously. Moreover, it enhances operational efficiency by avoiding the need to jump between disparate tools and management consoles to implement a cost savings recommendation.

Action 2.1.8
- Click **X** to close the **Action Details** panel.



Narration
Turbonomic continuously performs these cost optimization calculations in real time across large-scale deployments at CPU speed. Manual operator-driven optimization processes are time-consuming, do not scale, and can’t keep up with the dynamic rate of change.

Moreover, cloud service providers keep introducing new instance families to improve capabilities and price performance. The Turbonomic platform keeps abreast of these innovations and adds them to its analytics engine, providing the bank with updated recommendations based on the latest cloud offerings.

Let’s shift gears and look at storage cost optimization.

3 - Optimizing storage

3.1 - Get a global view of storage consumption

Narration
There are three forms of cloud storage: object storage, file storage, and block storage. The bank's cost optimization effort focuses on block storage, which functions like an external disk to the compute instance and provides low-latency state management for cloud workloads.

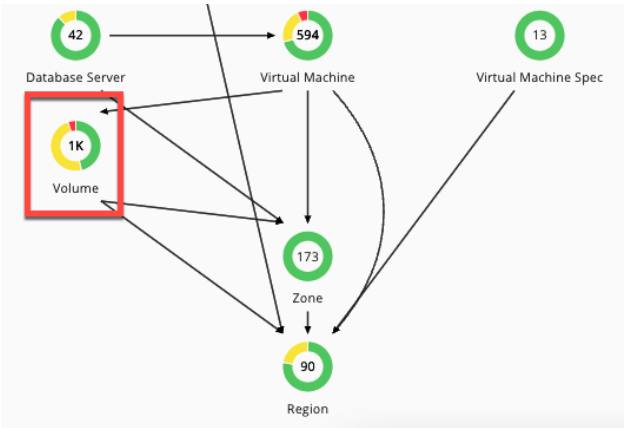
At a general level, there are two types of block storage: solid state drives (SSDs) and hard disk drives (HDDs). SSDs are good for low-latency, heavy IOPS types of workloads, whereas HDDs are good for high throughput sequential workloads.

It is important for the bank to choose a storage service based on the workload characteristics at hand. Some of the most common ways companies overspend on storage are:

1. *Unused volumes*: Compute instances are stopped, but the storage volumes are forgotten. This results in orphaned volumes, which then results in residual and accumulating costs.
2. *Overprovisioning*: Storage services are grossly oversized. For most cloud resources, you pay for what you *provision*, not for what you *use*.
3. *Mismatched storage type*: The storage choices are not appropriate for the requirements of the specific workload. For example, transaction processing is often best done on fast SSDs, whereas sequential workloads — like logging — are more compatible with cost-effective HDDs. This can result in either suboptimal performance or cost overruns.

Let's start exploring the storage optimization opportunities in this environment.

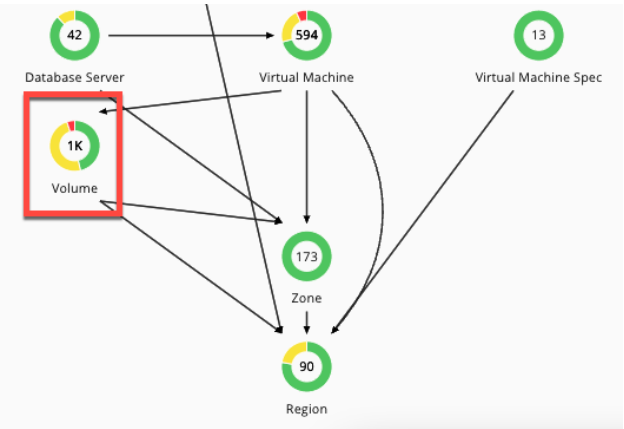
Action 3.1.1
- Point out the **Volume** entity, as highlighted in the screenshot and narration below.



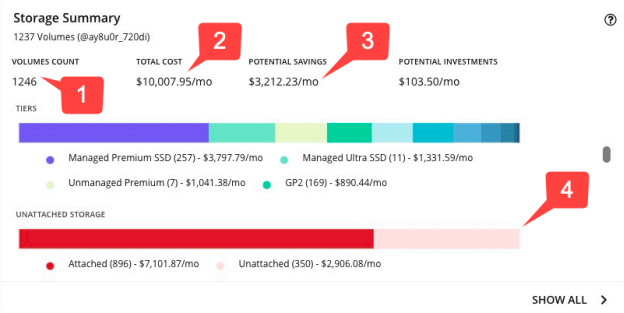
Narration
By looking at the 'Volume' entity in the global supply chain, we see there are about 1,000 storage volumes in this environment. Notice the color coding in the donuts: "green" means healthy, "yellow" represents efficiency recommendations (i.e., cost savings), and "red" represents performance recommendations (i.e., impact to the end user experience). Based on the proportion of green vs. yellow vs. red, the bank can see there are significant opportunities to improve the cost efficiency and performance of storage.

Let's drill down to better understand these cost reduction opportunities.

Action 3.1.2
- Click the **Volume** entity, which will take you to the **Storage Summary** dashboard.



Action 3.1.3
- Point out the details on the **Storage Summary** dashboard, as highlighted in the screenshot and narration below.



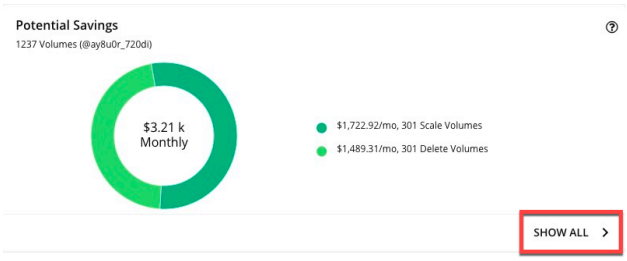
Narration
The 'Storage Summary' dashboard shows the number of volumes (1), total monthly cost of these volumes (2), and potential monthly savings that can be realized (3). The 'Unattached Storage' bar (4) shows that close to 30% of storage is currently unattached.

Let's take a deeper look at the specific volumes that can be reclaimed.

3.2 - Reclaim waste

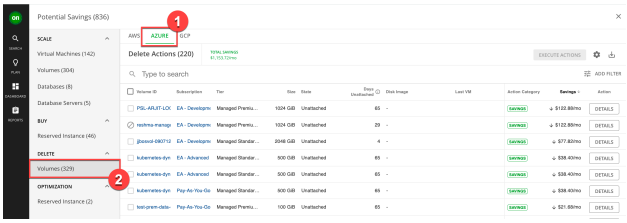
Action 3.2.1

- On the **Potential Savings** dashboard, click **SHOW ALL**.



Action 3.2.2

- Select the **AZURE** tab (1). Under the **DELETE** section of the left navigation bar, click **Volumes** (2).



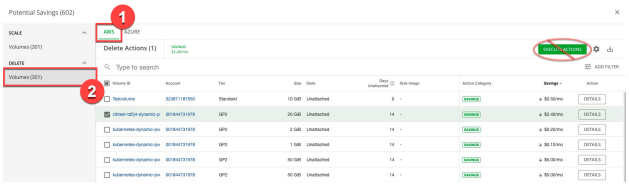
Narration

The dashboard identifies the bank's storage volumes that are unattached, including the number of days they have gone unused. On cloud, the lifecycles of the storage volumes are not intentionally associated with the lifecycles of the compute instances. Volumes that are unattached for many days are good candidates to reclaim and gain some savings. The data for each cloud provider appears on a separate tab. This dashboard displays the bank's potential savings from Azure, assuming they take all the recommended actions.

Turbonomic eliminates the need to hop between various cloud provider consoles. For example, we could also conveniently analyze storage usage in AWS or GCP by simply clicking the appropriate tab.

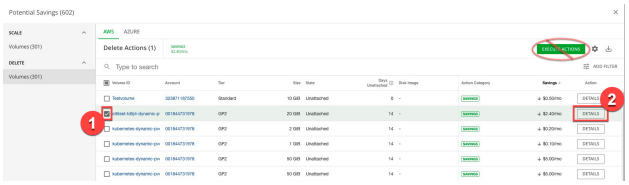
Action 3.2.3

- Select the **AWS** tab (1). Under the **DELETE** section of the left navigation bar, click **Volumes** (2).

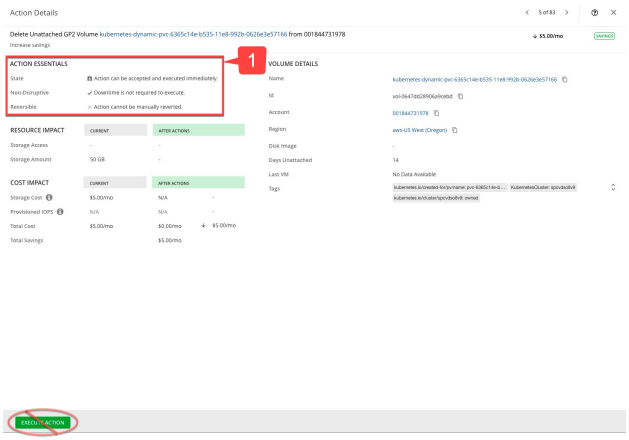


Narration
Just by switching tabs, the bank can now see the potential savings from AWS storage, assuming they take all the recommended actions.

Action 3.2.4
- On the **AWS** tab, select the first row which has a volume unattached for 14 days (1), and click the corresponding **DETAILS** button (2).



Action 3.2.5
- Point out the details under **ACTION ESSENTIALS** on the **Action Details** panel, as highlighted in the screenshot and narration below.



Narration
The Action Details panel provides the bank with supporting details behind the recommendation. Under the ‘Action Essentials’ section (1), it also shows whether the action can be taken immediately, whether downtime is required, and whether the action is manually reversible once taken.

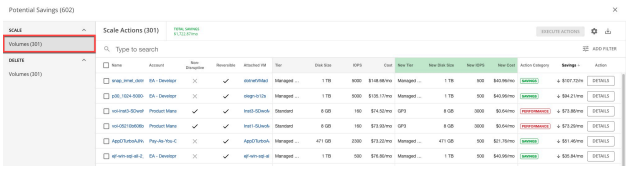
Action 3.2.6
- Click **X** to close the **Action Details** panel.

Narration
Now let’s examine the bank’s cost savings opportunities from moving volumes to alternate storage types that are a better fit for the workloads at hand.

3.3 - Select the right type of storage for the workloads

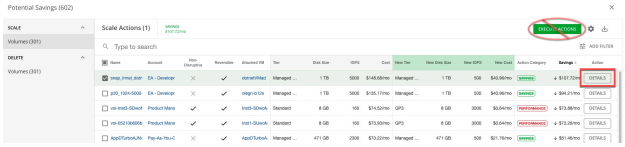
Narration
The bank needs to choose the right storage volume for their workloads. Capacity and performance are two fundamental factors around which these choices should be made. We'll see how even if wrong initial choices are made, the Turbonomic platform will detect these optimization and cost reduction opportunities using its continuous feedback loop.

Action 3.3.1
- Under the **SCALE** category of the left navigation bar, select **Volumes**.

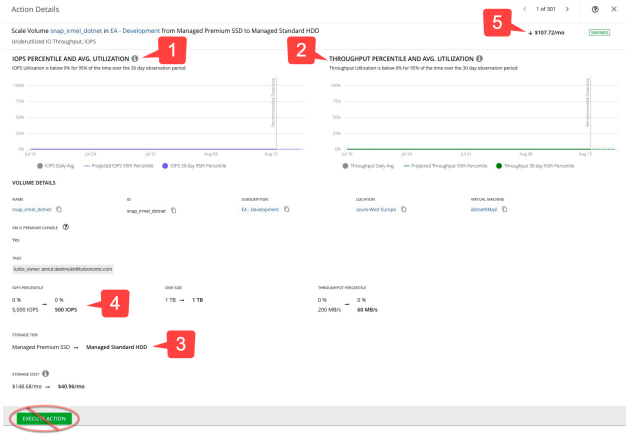


Narration
We can immediately see the various scale options, along with the bank's projected savings.

Action 3.3.2
- Click **DETAILS** on the first row.



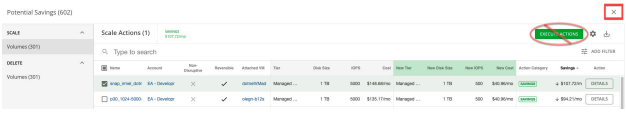
Action 3.3.3
- Point out the details on the **Action Details** panel, as highlighted in the screenshot and narration below.



Narration
We can see that this particular volume is underutilized from both IOPS (1) and throughput (2) perspectives. Based on the actual utilization pattern of this workload, Turbonomic's analytics show that this workload does not need an SSD and that the performance requirements of the workload can be met with a cheaper magnetic disk (3). This workload does not need the provisioned 5,000 IOPS and can get by with a much lower 500 IOPS (4). We therefore see that the storage capacity was overprovisioned by a factor of ten, at a cost that was 10x too high. Turbonomic's analytics engine computes the projected cost savings (5).

Action 3.3.4
- Click **X** to close the **Action Details** panel.

Action 3.3.5
- Click **X** to close the **Potential Savings** panel.



Narration
By analyzing and understanding the data access patterns of their workloads, the bank can make cost-efficient trade-offs about their storage choices, including the storage type, capacity, and IOPS.

4 - Leveraging cloud provider discounts

4.1 - Understand current pre-paid commitments

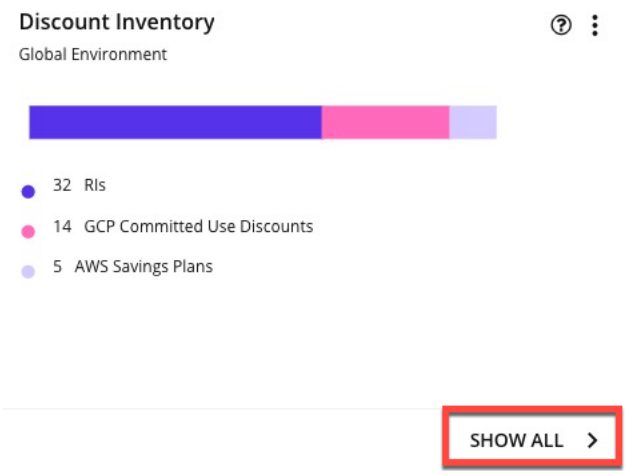
Narration
We've seen two primary ways in which the bank can reduce consumption: rightsizing compute instances and optimizing storage. In a cloud pricing model, this directly impacts the usage levels and therefore has a direct consequence on the bank's cloud bill. The other variable of this cloud spend equation is the unit price the bank pays for these resources.

All cloud providers offer discounts (typically up to 70% of the on-demand prices) in exchange for commitments of usage. These discounts can greatly reduce the rates. It's financially beneficial for the bank to run as many workloads as possible under these discounts.

Keeping track of reservations, monitoring utilization levels, and planning for additional purchases require visibility and continuous management oversight by the FinOps team. Doing all this across a growing multi-cloud environment is almost impossible to do manually.

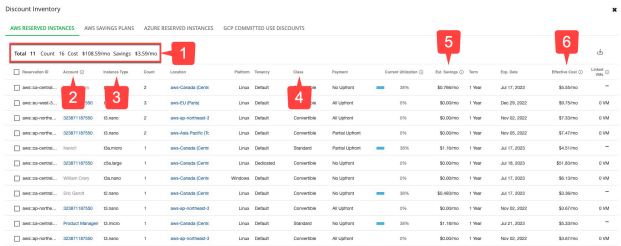
Let's look at how the Turbonomic platform helps the bank manage pre-paid capacity. As always, the first step to rate optimization is to get visibility into existing pre-paid reservations.

Action 4.1.1
- In the **Discount Inventory** panel of the global cloud view, click **SHOW ALL**.



Action 4.1.2

- Point out the details on the **Discount Inventory** panel, as highlighted in the screenshot and narration below.



Narration

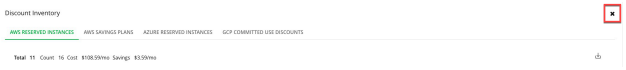
The ‘Discount Inventory’ panel shows the bank’s discounted pricing commitments with their cloud providers. Let’s explore the ‘AWS Reserved Instances’ tab.

Above the table, we see the total number of reserved instances, monthly cost, and monthly savings (1). In the table, each row represents an individual reservation, and the columns show the basic information about the reserved instance. All this information is automatically discovered from the bank’s cloud provider.

At a high level, we can see which accounts own the reservation (2), the instance type covered by this reservation (3), and whether it’s a standard or convertible reserved instance (4). The bank uses the ‘Est. Savings’ (5) and ‘Effective Cost’ (6) columns to quantify the ROI from purchasing these reserved instances.

Action 4.1.3

- Click **X** to close the **Discount Inventory** panel.

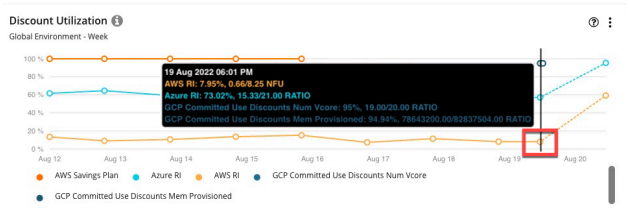


4.2 - Examine utilization levels of existing discounts

Narration
When used properly, reserved instances can deliver up to a 70% discount compared to on-demand prices. Reserved instances that aren't consumed during the commitment period are forfeited. Hence, an awareness of what reservations the bank owns and how well they are utilized is necessary to maximize their ROI. Let's take a look at RI utilization.

Action 4.2.1

- In the **Discount Utilization** dashboard, hover over the **AWS RI** data point on the black vertical line.



Narration
The graph shows the bank's reservation utilization data pulled from their cloud providers. The black line represents the current time. Data points to the left of the black vertical line represent historical data, while data points to the right are projections into the future. AWS uses Normalized Factor Units (NFUs), and Azure uses ratios as a uniform measure of compute capacity to help normalize capacity across different instance types.

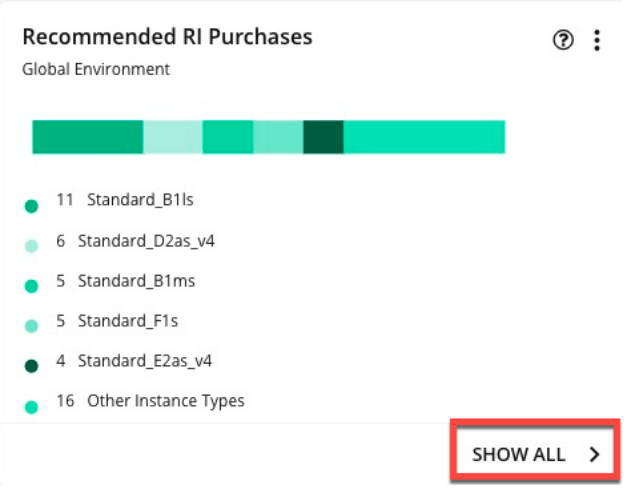
The graph shows current utilization of AWS at around 8% and of Azure at around 73%. As part of the optimization process, Turbonomic predicts that reservation utilization levels will increase to 60% and 90% for AWS and Azure respectively.

The bank's underlying goal here is to maximize the utilization of their inventory and cover as many workloads as possible. In this way, they can take advantage of these highly discounted prices and reduce their total cost of ownership.

4.3 - Explore the additional RI purchase recommendations

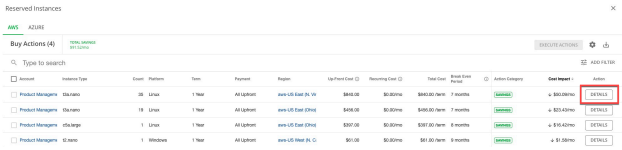
Action 4.3.1

- In the **Recommended RI Purchases** dashboard, click **SHOW ALL**.



Action 4.3.2

- Click **DETAILS** on the first row for additional information.

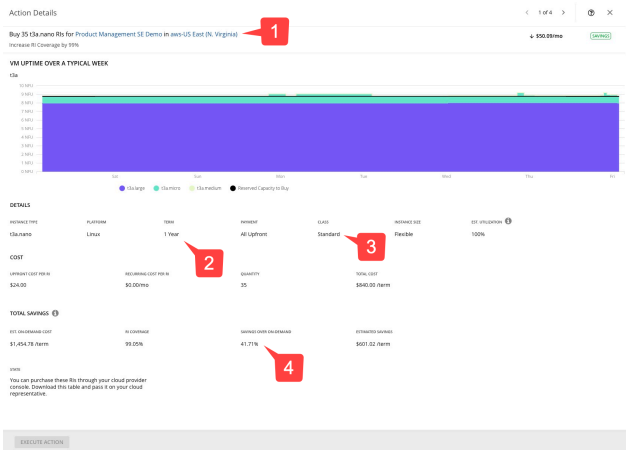


Narration

Reservations require a commitment and cost money. An all-upfront reservation provides the largest return, but it requires the full payment at the time of purchase. This is a bit contrary to the pay-as-you-go cloud model. However, over the committed term of one or three years, Turbonomic computes the break-even point, which is the time when the amortized cost of reserved instances will be lower than paying the on-demand prices.

Action 4.3.3

- Point out the details on the **Action Details** panel, as highlighted in the screenshot and narration below.



Narration

Based on its analysis of the existing workloads, Turbonomic makes a recommendation to purchase additional reserved instances to ensure that more workloads are running at discounted prices (1).

When initially configuring Turbonomic, the bank set up a purchase profile indicating their preferences. Those preferences are then taken into account when making the recommendations on RI terms, such as one or three years (2), or RI class, such as standard or convertible (3).

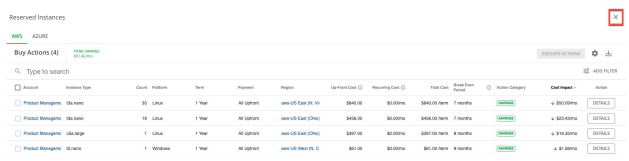
Turbonomic estimates the bank will save an estimated 41.71% from purchasing these discounts (4).

Action 4.3.4

- Click **X** to close the **Action Details** panel.

Action 4.3.5

- Click **X** to close the **Reserved Instances** panel.



Narration
By leveraging cloud provider discounts, the bank is able to significantly reduce their cost per unit. In addition, Turbonomic can automate any of its recommendations — not just for RIs, but also for compute and storage optimization. The bank is able to define specific automation policies that enable them to execute these recommendations in real time, directly from the Turbonomic platform.

Summary

In this demo, we've seen how the Turbonomic platform helps the FinOps team at Gen-Z Bank control their cloud costs. This was not done at the expense of application performance; Turbonomic optimizes both. Turbonomic's cost-cutting recommendations are designed to ensure that performance is not negatively impacted.

Gen-Z Bank used the Turbonomic platform to get a deeper visibility into their cloud deployment. They used the platform-generated analytics to identify and eliminate waste, optimize compute instances, and right-fit storage choices. This ensures their deployments are always available, performant, and cost-optimized, so the bank can achieve the lowest cost without sacrificing their performance needs.

Thank you for attending today's presentation.