

How to Optimize Your Public Cloud Workload Cost with Capgemini and CloudWisdom



Imagine if your organization suddenly had a quarter-million dollars freed up in your engineering budget without having to sacrifice on performance. What would you do with it? For a vacation-experience customer at Capgemini, the decision was easy: reinvest in the roll-out of new applications.

[Capgemini](#) is a leader in digital transformation that delivers what's next for customers through data science, strategy, and innovation. That's why they decided to use Virtana's CloudWisdom product to discover major cost savings for their project. CloudWisdom empowers IT management to optimize Amazon Web Services (AWS) workload resource utilization and reinvest savings without sacrificing application performance.

It took just 10 days for their customer to realize 30 percent savings on overlooked AWS assets, resulting in an overall 10 percent optimization in AWS cost. This level of savings is a boon for any business.

Keep reading to learn how you can optimize your public cloud workloads using Virtana's CloudWisdom platform.

Basic Steps for Cloud Cost Optimization

Here are the three steps we use at Virtana to optimize cost for customers:

1. Remove idle resources, which poses zero risk on application performance and gets you instant savings.
2. Right size over-allocated resources. This should be done after proper data analysis.
3. Plan [Amazon EC2 Reserved Instances](#) (RI) purchasing for resized EC2 instances to capture long-term savings.

Addressing idle resource clutter looks easy, but costs often get lost in environments lacking a proper tagging strategy. Tags created to address wasted resources also help you to properly size resources by improving capacity and usage analysis. After right sizing, committing to reserved instances gets a lot easier—most of the work required to scope out

ideal contracts has been done. You then have everything needed to realize significant savings.

By following these steps, we estimate the total savings for Capgemini's vacation-experience customer project to approach \$2 million dollars annually. In fact, more than \$250,000 was saved in the first month alone.

Phase 1: Creating Asset Awareness

Capgemini's project had a mandate to architect, implement, and operationalize a new application platform. We scoped and coordinated the effort involved in setup that would allow for a deep analysis of the customer's development, quality assurance (QA), and production environments.

In total, these environments were comprised of hundreds of virtual servers or [Amazon Elastic Compute Cloud](#) (Amazon EC2) instances spread across multiple AWS accounts. Once they understood the scope, it was time to lay the foundation of good governance.

Start with Tagging All Assets

We used tags to overlay business and organizational information onto the customer's billing and usage data. When you apply tags to your AWS resources and activate the tags, AWS adds this information to the Cost and Usage Reports. These tags become available when accessing your billing data through the API.

After all tagging is complete, we recommend using the Consolidated Billing feature found in your [AWS Organizations](#) console. This makes tracking easy and opens up savings opportunities via volume discounts.

Integrate with CloudWisdom

Capgemini needed a cost analysis solution that would pair well with the AWS API to optimize the cost of this project with minimal impact on application performance. They chose Virtana's CloudWisdom platform because it delivers an ever-growing list of reporting and recommendation tools to keep your AWS portfolio optimal and lean.

CloudWisdom's tools helped with:

- Bill analysis and cost alerts to detect unusual changes in billing.
- Detecting idle and unused resources.
- Identifying over-allocated capacity and right sizing the computing resources.
- Recommending RI purchases based on granular usage analysis.
- Detecting performance bottlenecks to ensure safe capacity adjustments.

Track Your Inventory

CloudWisdom kept track of each asset's relationships in its element (resource) inventory and monitored their performance.

Update Internal Processes

All assets being tagged, tracked, and integrated with Virtana's CloudWisdom platform meant that Capgemini was ready to create a new change management process. They started by capturing a set of application benchmarks in CloudWisdom using their customer's QA environment. These benchmarks consisted of baseline utilization measurements of all standard components.

They then set up alerting policies in CloudWisdom that would notify us of any upward deviation in memory, IOPS, or CPU utilization for a new release. A new release would not leave QA without additional justification and approval if these deviations surpassed specific thresholds. This enabled our customer to understand the operational cost impact of change.

Next, Capgemini created a technical review process to validate any resizing recommendations given by CloudWisdom. This included subject matter expert adjustment and sign-off before execution since sizing recommendations don't have full context for resources in question.

Finally, they created a plan for removing the idle resource volumes found by CloudWisdom. This involved archival of idle [Amazon Elastic Block Store](#) (EBS) volumes into [Amazon Simple Storage Service](#) (Amazon S3) or [Amazon Glacier](#) until they could be fully reviewed for removal.

Capgemini also improved the automation of de-provisioning of Amazon EC2 instances. While primary EBS volumes can be automatically terminated with the Amazon EC2 instances, secondary EBS volumes can't. Enhanced automation ensured that secondary EBS volumes wouldn't get forgotten and orphaned as unattached volumes.

With those governance tasks complete, Capgemini was ready to tackle right sizing the hosted cloud platform and ultimately optimize its resource usage.

Phase 2: Turning Data into Strategy

When most people think of optimizing for cost, their initial thoughts are to size instances down—but that's not always the case. CloudWisdom's EC2 Recommendation Report discovered areas where it was more cost-effective for our customer to size up their instances.

DETERMINE YOUR RIGHT SIZING REQUIREMENTS

Start with your historic workload patterns. We analyzed utilization measurements in CPU, memory, I/O, and network for this project. The screenshot in Figure 1 shows five live

instances that needed to be resized, using CloudWisdom's [EC2 Recommendation Report](#).

This report analyzes data ingested from:

- Customer's detailed billing CSV files to ascertain current cost and element metadata.
- AWS Price List Service Endpoint to compare current spend with all of the available SKUs in the AWS product line.

Current Type	Proposed Type	Current Cost	Projected Cost	Projected Savings	Projected Savings %
m4.16xlarge	x1e.2xlarge	27955.20	14571.65	13383.55	48%
m4.16xlarge	x1e.2xlarge	27955.20	14571.65	13383.55	48%
m4.16xlarge	x1e.2xlarge	27955.20	14571.65	13383.55	48%
m4.16xlarge	x1e.2xlarge	27955.20	14571.65	13383.55	48%
m4.10xlarge	x1e.xlarge	17472.00	7285.82	10186.18	58%

Figure 1 – CloudWisdom's EC2 Recommendation Report table.

If the instances in Figure 1 were in your portfolio, you'd net over \$63,000 in yearly savings—on just those five instances. That's not including additional savings to be gained by converting those instances from on-demand to reserved.

CloudWisdom calculated the projected savings by cross-referencing the customer's given instance types with all possible compatible SKU permutations (over 400,000). Those permutations were boiled down to a list that would meet capacity requirements. Finally, that list was refined to the single, most-optimal instance type by applying any additional constraints set by the report user.

Note that the level of insight achieved at this stage is directly dependent upon having your billing accounts consolidated and all assets consistently tagged. By having everything prepared for analysis, you can:

- Determine requirements for each and every Amazon EC2 instance.
- Select configurations best suited for each specific workload.

Check for Idle Resources

Consider how often your environment changes. It's difficult to keep track of whether every resource purchased is being used. A couple of unattached [Elastic Load Balancers](#), some old EBS snapshots, and a few idle Amazon EC2 instances may not be a huge issue, but scale this problem up. Larger projects with multiple environments and ephemeral instances often harbor significant wasteful spending.

For this project, CloudWisdom's suite of [Idle Resource Reports](#) exposed 1,152 unused resources, which created \$250,068 in instant savings for our customer.

In Figure 2, you can see an example screenshot of the Unattached EBS Volume Report in CloudWisdom; this report uses detailed billing to obtain cost data and the AWS Cost Explorer Service Endpoint to obtain element metadata.

Name	Location	Size	VolumeId	IOPS	Type	Monthly Cost
Volume-Name-01	us-east-1a	2,048	vol-09a876b54c356	6,144	gp2	\$204.80
Volume-Name-02	us-east-1b	1,000	vol-34a876b54c225	3,000	gp2	\$100.00
Volume-Name-03	us-east-1b	1,000	vol-56b76b54c69	3,000	gp2	\$100.00
Volume-Name-04	us-east-1b	1,000	vol-78j876b54c09	3,000	gp2	\$100.00
Volume-Name-05	us-east-1b	1,000	vol-91h876b54c21	3,000	gp2	\$100.00

Figure 2 – Results from CloudWisdom's Unattached EBS Volume Report.

There are currently three Idle Resource Reports available in CloudWisdom that look for:

- Unattached EBS volumes
- Unattached Elastic Load Balancers
- EBS volumes on stopped Amazon EC2 instances

Examine Results

The total savings results are shown in Figure 3 and were made possible by utilizing Capgemini's implemented governance processes, AWS services, and Virtana's CloudWisdom platform.

Type of Savings	Identified Savings
Idle resources	\$250,068
Right sizing	\$1,164,532
Reservations	\$580,656

Figure 3 – Total savings results calculated using tools from Capgemini, Virtana, and AWS.

Phase 3: Maintaining the Success

Building a culture of proactive cost-awareness is not easy when the emphasis in our industry is often on project delivery speed, application performance, and platform stability. Once you've finally got the buy-in and managed to complete an initial savings project, how do you prevent your applications from gradually returning to an expensive, less-optimal state?

Use Gates to Implement Change

Suppose you have a new release being deployed. During UAT testing, your capacity measurements reveal the new software version consumes computing resources at a much higher rate. This would be the right time to decide between rolling back to a previous version or adjusting your budget and upgrading.

Use this opportunity to create a gate that adds governance for capacity (and cost) management.

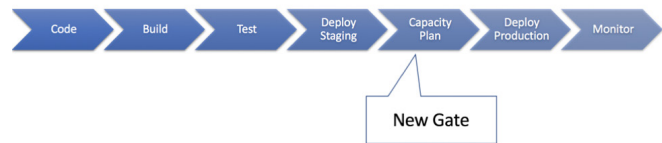


Figure 4 – Create a gate to add governance for capacity and cost management.

Capgemini measured resource consumption for this project using CloudWisdom's Utilization Boxplot Report. This report collects utilization data through the AWS Cost Explorer Service Endpoint, aggregating it over time. Figure 5 shows a list of resources on the left, with quartile boxes for each resource which gauge the min-max utilization (as well as the median).

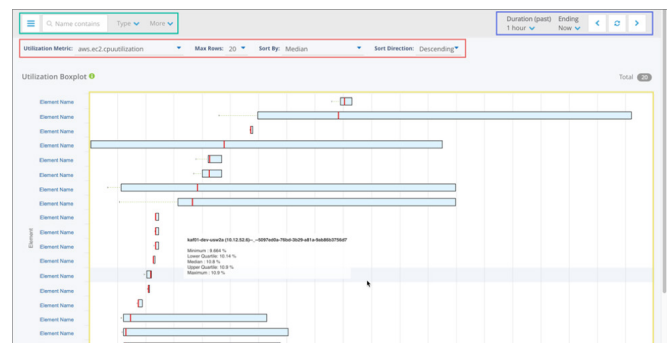


Figure 5 – CloudWisdom's Utilization Boxplot report shows resource utilization over time.

Measure Performance Before and After Changes

CloudWisdom isn't just a cost-optimization solution. CloudWisdom also detected a major performance bottleneck issue for Capgemini's customer—before their pre-existing monitoring solution—thanks to CloudWisdom's set of

default alerting policies that look for common infrastructure performance issues.

These policies are backed by machine learning anomaly detection algorithms that operate in real-time to learn the expected behavior of each metric. This is achieved within the context of other metrics for a given resource, for each individual resource.

The expected behavior is called a contextual band, and when using Virtana's monitoring dashboards it can be seen as the thick blue line on a metric widget.

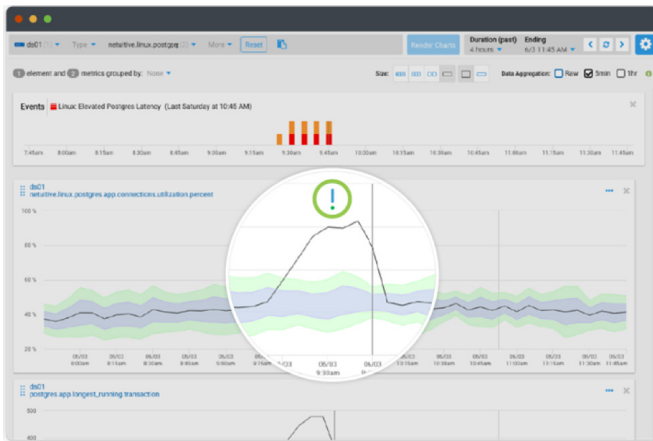


Figure 6 – A dashboard metric widget using CloudWisdom's anomaly detection capability.

Because CloudWisdom detected the issue so fast, Capgemini asked for backup from Virtana's support service team to troubleshoot.

This incident ended up becoming a case study which was presented during a CTO council meeting to highlight the benefits of managing infrastructure cost, capacity, and performance together.

Conclusion

At Virtana, we recommend laying the proper foundation at the onset of any cloud migration project, regardless of your application architecture, so you can avoid wasting money and maintain a cost-conscious operational culture over time.

To recap, here are the most important steps to remember:

- Eliminate waste from your current platform by removing idle and over-allocated capacity.
- Purchase Reservation Instances (RIs) after you have stabilized your requirements.
- Monitor for performance bottlenecks that may be caused by capacity change.
- Implement a new gate in your CI/CD process to detect and plan for capacity changes.

It's never too early to implement these tools and processes to avoid waste. After all, your budget is best spent on reinvesting in innovation projects.



[Learn](#) more about VirtualWisdom



[Contact](#) a Virtana Expert



[View](#) more Resources

Virtana
2331 Zanker Road
San Jose, CA 95131
Phone: +1.408.579.4000